**A Deep Transfer Learning-based Edge Computing Method for Home Health Monitoring**

**1.INTRODUCTION**

In India, only 1.9 millions hospital beds in all kind hospitals are currently available for population around 1.35 billion that is, only 1.4 beds per 1000 peoples. This situation is also not far better in other countries. In addition, those countries that are comparably on top of that list also may not be able to cope with the challenges arising from a pandemic. Therefore, home health services need to be improved to cope with a pandemic or epidemic such as COVID-19. Moreover, as the parentage of aged people(elders) is increasing steadily , so home health services, are also very useful health practice for elders who live at home. As Artificial Intelligence (AI) is augmenting human capabilities for many human-centred tasks. Therefore, AI could also assist home health services in many ways . Automated patient or elders monitoring (in short we are calling it ‘Home Health Monitoring’) one such non-intrusive and economical sub-area of these services; these sub-area may include activity monitoring, sleep monitoring, respiration monitoring, fall detection, facial expression understanding, speech recognition, hand hygienic practice monitoring, etc.. For these kind of tasks, deep learning (DL) and computer vision (CV) are very effective as studied. But DL especially for tasks of CV, required GPU-enabled computing devices , which may not be available for every household. To address this issue, one approach is to leverage cloud computing technique, where data needs to be sent to a remote cloud server for processing outside from home. But in this case, privacy, security and bandwidth scarcity are big issues and real-time computing may not be possible. These disincentives motivate to use the new technology of Edge Computing (EC). EC could be used to compute data of home health monitoring inside the home or house. However, some challenges also exists as edge devices(ED) are generally small and have low computing capabilities. In addition, the DL-based model usually takes a large amount of data which is also a big challenge for health sectors. In this article, we propose a deep transfer learning-based edge computing method for home health monitoring (TL-ECHM). Here, we consider a transfer learning approach, where a pre-trained Convolutional Neural Network -based model which is trained with its available dataset, may use in ED with fine-tuning using a small amount of ground labeled dataset. In this way, it would take much less computing resources and the required on-site visual computing shall be possible at an ED. Therefore, mitigation of the abovementioned challenges shall be possible. A possible working scenario of TL-EC-HM is depicted in Fig. 1, where how a caregiver center, cloud server, ED, and IoT device(sensor) are connected to each other to form a system is shown. The highlights of this article are listed as below:

* We provide a study on health and activity monitoring for patient as well as elders at their home for mitigating health crisis.
* We propose a method (TL-EC-HM) based on DTL and EC for home health monitoring.
* We analyze the proposed privacy-preserving TL-EC-HM for on-site visual computing.
* We provide some future research directions.

* 1. **Objective of the Project**

The health-care gets huge stress in a pandemic or epidemic situation. Some diseases such as COVID-19 that causes a pandemic is highly spreadable from an infected person to others. Therefore, providing health services at home for noncritical infected patients with isolation shall assist to mitigate this kind of stress. In addition, this practice is also very useful for monitoring the health-related activities of elders who live at home. The home health monitoring, a continuous monitoring of a patient or elder at home using visual sensors is one such nonintrusive sub-area of health services at home. In this article, we propose a transfer learning-based edge computing method for home health monitoring. Specifically, a pre-trained convolutional neural network-based model can leverage edge devices with a small amount of ground-labeled data and fine-tuning method to train the model. Therefore, on-site computing of visual data captured by RGB, depth, or thermal sensor could be possible in an affordable way. As a result, raw data captured by these types of sensors is not required to be sent outside from home. Therefore, privacy, security, and bandwidth scarcity shall not be issues. Moreover, real-time computing for the above-mentioned purposes shall be possible in an economical way. Keywords—AI-enabled Health Monitoring, Ambient Intelligence, Computer Vision, COVID-19 Pandemic, Deep Learning, Edge Computing, Transfer Learning, Visual Sensors

**2. LITERATURE SURVEY**

**“Covid19 in india : State-wise estimates of current hospital beds, intensive care unit (icu) beds and ventilators,”**

**Back ground** The rapid spread of COVID-19 globally has prompted policymakers to evaluate the capacity of health care infrastructure in their communities. Many hard-hit localities have witnessed a large influx of severe cases that strained existing hospitals. As COVID-19 spreads in India, it is essential to evaluate the country’s capacity to treat severe cases.

**“Hospital readiness for covid-19,”** Hospitals play a critical role within the health system in providing essential medical care to the community, particularly in a crisis. Prolonged and combined outbreaks can lead to the progressive spread of disease with rapidly increasingservice demands that can potentially overwhelm the capacity of hospitals and the health system at large. To enhance the readiness of the health facilities to cope with the challenges of the outbreak, a pandemic, or any other emergency or disaster, hospital managers need to ensure the initiation of relevant generic priority action. This document aims to provide a checklist of the key action to take in the context of a continuous hospital emergency preparedness process.  
This checklist has been prepared with the aim of supporting hospital managers and emergency planners in achieving the above by defining and initiating actions needed to ensure a rapid response to the COVID-19 outbreak. The checklist is structured on eleven key components; under each component, there is a list of questions regarding the status of implementation of the recommended action specific to that component. Hospitals at risk of increased health service demand should be prepared to initiate the implementation of each action promptly. The section on “Recommended reading” lists selected tools, guidelines and strategies relevant to each component, as well as other supporting documentation

, **“The practical implementation of artificial intelligence technologies in medicine,”**The development of artificial intelligence (AI)-based technologies in medicine is advancing rapidly, but real-world clinical implementation has not yet become a reality. Here we review some of the key practical issues surrounding the implementation of AI into existing clinical workflows, including data sharing and privacy, transparency of algorithms, data standardization, and interoperability across multiple platforms, and concern for patient safety. We summarize the current regulatory environment in the United States and highlight comparisons with other regions in the world, notably Europe and China.**.**

**“Sceh: smart customized e-health framework for countryside using edge ai and body sensor networks,”** Due to the shortage and unbalance of medical resources, it is difficult for patients in the countryside to get high-quality and timely medical services from the central medical facility. Existing researches of fog e-health has the potential of providing real-time medical services for the countryside with body sensor networks (BSN), but there are two limitations. On one hand, because of the medical services requiring not only low-latency but also high-quality, constructing an AI e-health service on resource-constrained fog with edge AI is necessary but unsolved. On the other hand, because of the regional differences in disease risk, there is a lack of an effective mechanism to provide a customized fog AI e-health service for patients in different regions. To address these issues, a smart customized e-health (SCEH) framework is proposed in this paper to provide edge-intelligent and customized medical services for the countryside. Firstly, semantics-based lightweight and meticulous load management mechanism is designed to reduce data load and involve medical semantic. Secondly, model-ensemble based fog AI collaborative analysis mechanism is proposed for load balance and knowledge integration. Thirdly, an attention-weight based customized fog AI e-health generation mechanism is devised for regional medical model reconstruction. The simulation results demonstrate the effectiveness of SCEH which ensures both the accuracy and low latency of fog e-health with limited resource.

**“A nurse-driven method for developing artificial intelligence in “smart” homes for aging-in-place,”** To offer practical guidance to nurse investigators interested in multidisciplinary research that includes assisting in the development of artificial intelligence (AI) algorithms for “smart” health management and aging-in-place.

**“Homecare robotic systems for healthcare 4.0: Visions and enabling technologies,”** Powered by the technologies that have originated from manufacturing, the fourth revolution of healthcare technologies is happening (Healthcare 4.0). As an example of such revolution, new generation homecare robotic systems (HRS) based on the cyber-physical systems (CPS) with higher speed and more intelligent execution are emerging. In this article, the new visions and features of the CPS-based HRS are proposed. The latest progress in related enabling technologies is reviewed, including artificial intelligence, sensing fundamentals, materials and machines, cloud computing and communication, as well as motion capture and mapping. Finally, the future perspectives of the CPS-based HRS and the technical challenges faced in each technical area are discussed.

**“Remote patient monitoring: a comprehensive study,”**

Lifestyle influences morbidity and mortality rates in the world. Physical activity, a healthy weight, and a healthy diet are key preventative health behaviours that help reduce the risk of developing type 2 diabetes and its complications, such as cardiovascular disease. A healthy lifestyle has been shown to prevent or delay chronic diseases and their complications, but few people follow all recommended self-management behaviours. This work seeks to improve knowledge of factors affecting type 2 diabetes self-management and prevention through lifestyle changes. This paper describes the design, development, and testing of a diabetes self-management mobile app. The app tracked dietary consumption and health data. Bluetooth movement data from a pair of wearable insole devices are used to track carbohydrate intake, blood glucose, medication adherence, and physical activity. Two machine learning models were constructed to recognise sitting and standing. The SVM and decision tree models were 86% accurate for these tasks. The decision tree model is used in a real-time activity classification app. It is exciting to see more and more mobile health self-management apps being used to treat chronic diseases.

**“Vision-based patient monitoring: a comprehensive review of algorithms and technologies.”**

Vision-based monitoring for assisted living is gaining increasing attention, especially in multi-modal monitoring systems owing to the several advantages of vision-based sensors. In this paper, a detailed survey of some of the important vision-based patient monitoring applications is presented, namely (a) fall detection (b) action and activity monitoring (c) sleep monitoring (d) respiration and apnea monitoring (e) epilepsy monitoring (f) vital signs monitoring and (g) facial expression monitoring. The challenges and state-of-art technologies in each of these applications is presented. This is the first work to present such a comprehensive survey with the focus on a set of seven most common applications pertaining to patient monitoring. Potential future directions are presented while also considering practical large scale deployment of vision-based systems in patient monitoring. One of the important conclusions drawn is that rather than applying generic algorithms, use of the application context of patient monitoring can be a useful way to develop novel techniques that are robust and yet cost-effective.

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

Health monitoring is the monitoring of a worker by doctors to**identify changes in their health status because of exposure to certain substances.** If you are a person conducting a business or undertaking (PCBU), you must provide health monitoring to workers if there is a significant risk to workers health because of exposure to a hazardous chemical.

**Disadvantage**

1.time taken process

**3.2 PROPOSED SYSTEM**

A pandemic or epidemic crisis puts a lot of strain on the medical system. Some illnesses, like COVID-19, which creates a pandemic, are very contagious when someone is infected with them. others. Therefore, giving medical care at home for patients who are not in severe condition and who are placed in isolation will help to reduce particular form of pressure. Furthermore, this technique is really beneficial. for keeping an eye on the health-related activities of elderly residents at home. The home health monitoring is a persistent watch of One such nonintrusive division of at-home medical care involves a patient or elderly person employing visual sensors. This article explores suggest an edge computing approach based on transfer learning for health monitoring at home. An especially trained convolutional Using edge devices with a neural network-based model tiny quantity

**Advantage**

1.More Accuracy.

2.quick response.

**MODULES**

To implement this project we have designed following modules

To implement this project we have designed following modules

1. Edge Cloud Server: this is a cloud application which received images from client and display to medical peoples for monitoring
2. AI Sensor Client Application: this module we will upload dataset to train AI model and then load the model and whenever user upload any images then it will predict condition and report to cloud server.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artefacts will be produced. Software artefacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artefacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artefact is linked to a specific design element, and that each developed artefact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behaviour of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**HARDWARE REQUIREMENTS:**

# Processor - Pentium –IV

* Speed - 1.1 GHz
* RAM - 4GB(min)
* Hard Disk - 500GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows 10/above
* Programming Language - Python 3.7 /above

**4. SYSTEM DESIGN**

**UML Diagram:**

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* + **User Model View**
    1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective.
  + **Structural Model view**
    1. In this model the data and functionality are arrived from inside the system.
    2. This model view models the static structures.
* **Behavioural Model View**

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* **Implementation Model View**

In this the structural and behavioural as parts of the system are represented as they are to be built.

* **Environmental Model View**

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake



**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

****

**Sequence diagram:**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.



**Component Diagram:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

****

**Deployment Diagram:**

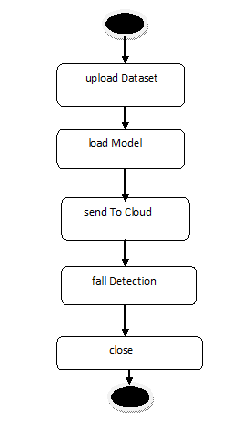
A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

****

**Activity Diagram:**

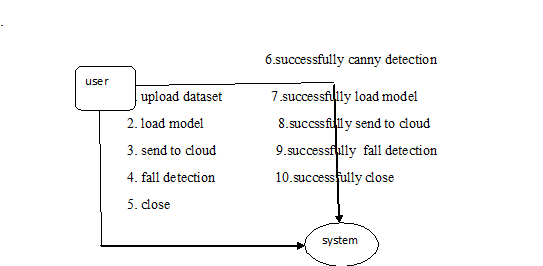
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

****

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results



**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

**SENSOR.PY**

import socket

from threading import Thread

from socketserver import ThreadingMixIn

import os

import pickle

import cv2

running = True

SERVER.PY

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

from tkinter.filedialog import askopenfilename

import socket

from keras.preprocessing.image import load\_img

from keras.preprocessing.image import img\_to\_array

import pickle

from keras.models import load\_model

import cv2

import numpy as np

main = tkinter.Tk()

main.title("A Deep Transfer Learning-based Edge Computing Method for Home Health Monitoring") #designing main screen

main.geometry("1300x1200")

global filename

global model

global X,Y

class\_labels = ['Fall', 'No Fall']

def uploadDataset():

global filename

filename = filedialog.askdirectory(initialdir=".")

text.delete('1.0', END)

text.insert(END,filename+" loaded\n");

def loadModel():

text.delete('1.0', END)

global model

model = load\_model('model/model.h5')

text.delete('1.0', END)

text.insert(END,"VGG16 Deep Transfer Learning Health Monitoring Model Loaded\n\n");

def sendToCloud(img, result):

img = cv2.resize(img,(100,100))

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client.connect(('localhost', 2222))

features = []

features.append("sensordata")

features.append(result)

features.append(img)

features = pickle.dumps(features)

client.send(features)

def fallDetection():

global model

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="testImages")

temps = cv2.imread(filename)

img\_h, img\_w, c = temps.shape

image = load\_img(filename, target\_size=(80, 80))

image = img\_to\_array(image) / 255.0

image = np.expand\_dims(image, axis=0)

(boxPreds, labelPreds) = model.predict(image)

boxPreds = boxPreds[0]

x = boxPreds[0]

y = boxPreds[0]

w = boxPreds[0]

h = boxPreds[0]

x\_min = int((x-w/2)\*img\_w)

y\_min = int((y-h/2)\*img\_h)

x\_max = int((x+w/2)\*img\_w)

y\_max = int((y+h/2)\*img\_h)

cv2.rectangle(temps, (x\_min,y\_min), (x\_max, y\_max), (0, 255, 0), 2)

predict= np.argmax(labelPreds, axis=1)

predict = predict[0]

accuracy = np.amax(labelPreds, axis=1)

accuracy = accuracy[0]

print(str(class\_labels[predict])+" "+str(accuracy))

temps = cv2.resize(temps, (600,500))

cv2.putText(temps, "Health Condition Predicted As "+str(class\_labels[predict]), (10,50), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (255, 0, 0), 2)

cv2.putText(temps, "Prediction Accuracy "+str(accuracy), (10,110), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (255, 0, 0), 2)

sendToCloud(temps, class\_labels[predict])

cv2.imshow("Health Condition Predicted As "+str(class\_labels[predict]), temps)

cv2.waitKey(0)

def close():

main.destroy()

font = ('times', 16, 'bold')

title = Label(main, text='A Deep Transfer Learning-based Edge Computing Method for Home Health Monitoring')

title.config(bg='deep sky blue', fg='white')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=50,y=120)

text.config(font=font1)

font1 = ('times', 13, 'bold')

uploadButton = Button(main, text="Upload Fall Detection Dataset", command=uploadDataset)

uploadButton.place(x=50,y=550)

uploadButton.config(font=font1)

loadButton = Button(main, text="Generate & Load Health Monitoring Model", command=loadModel)

loadButton.place(x=350,y=550)

loadButton.config(font=font1)

detectionButton = Button(main, text="Detect Fall Detection from RGB Image", command=fallDetection)

detectionButton.place(x=740,y=550)

detectionButton.config(font=font1)

exitButton = Button(main, text="Exit", command=close)

exitButton.place(x=50,y=600)

exitButton.config(font=font1)

main.config(bg='LightSteelBlue3')

main.mainloop()

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Implementation

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

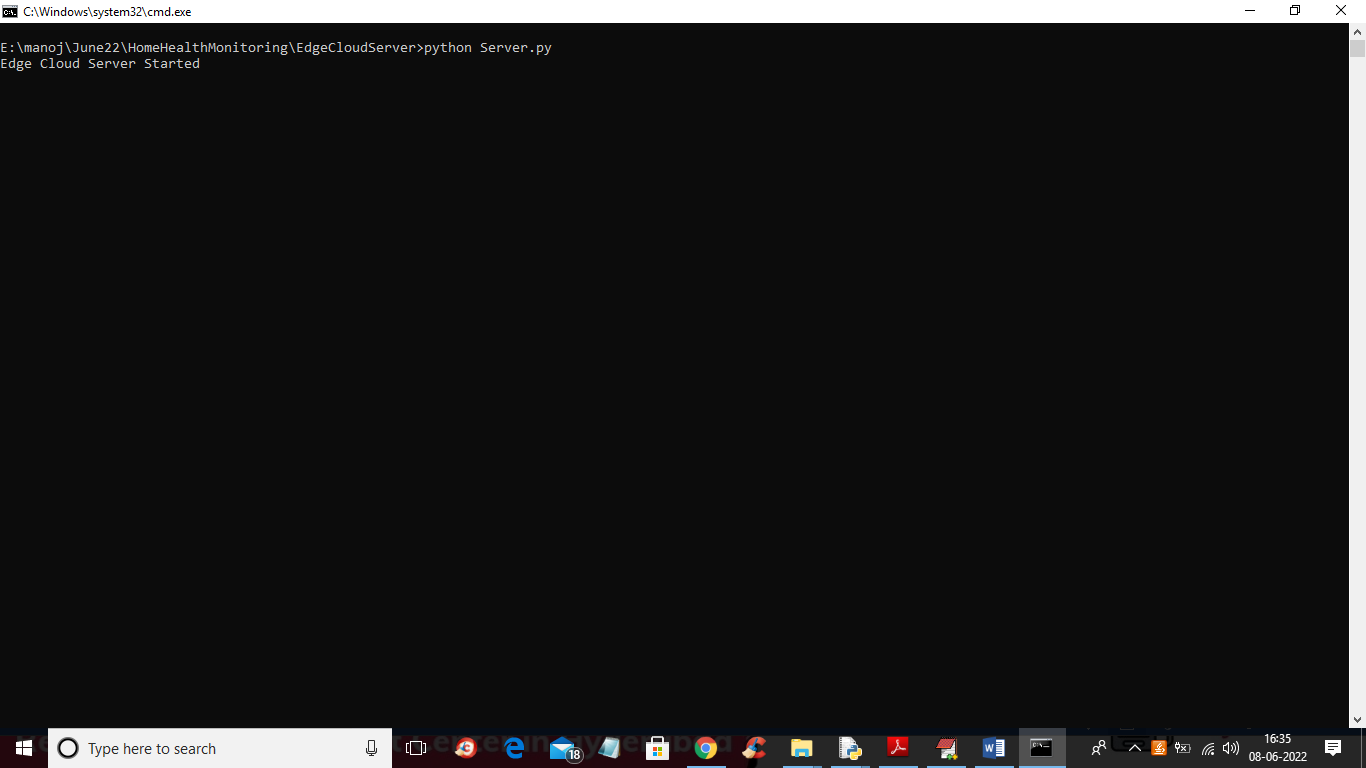
**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

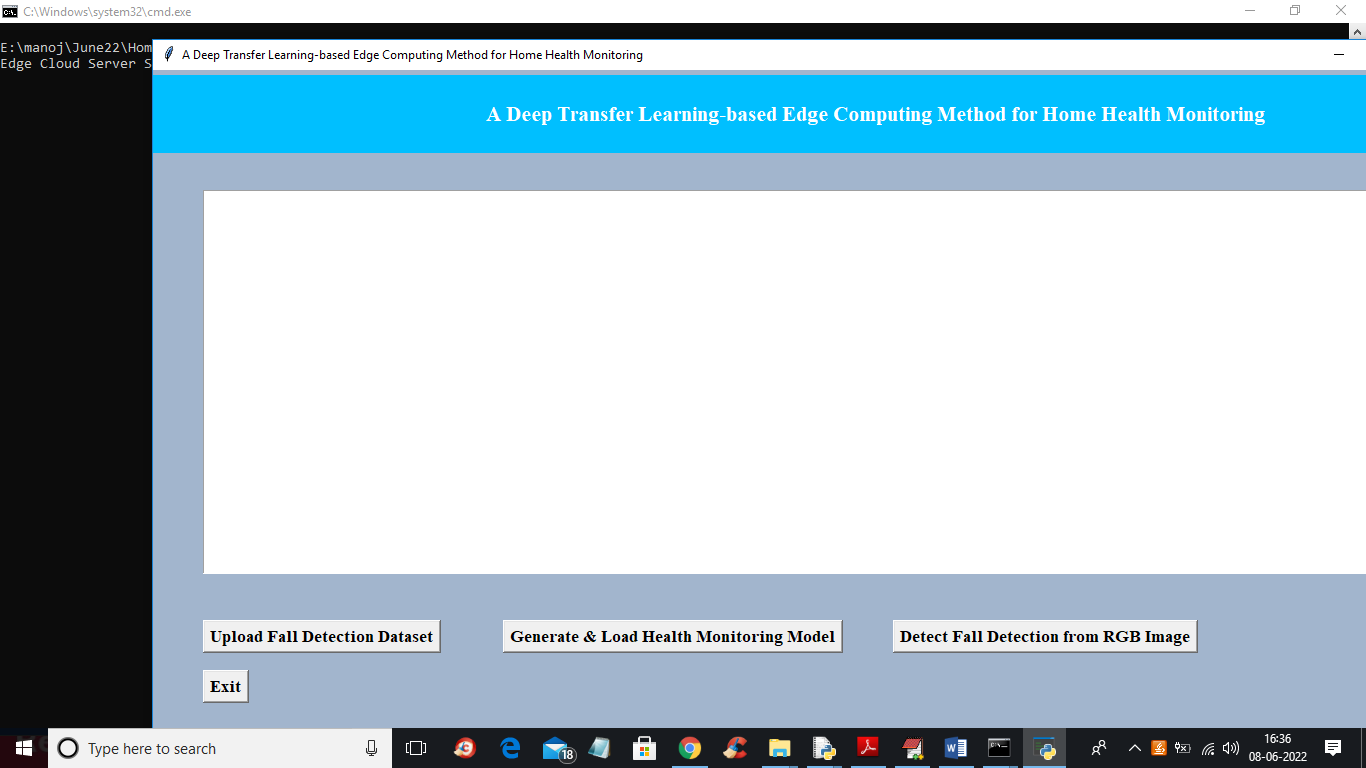
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | upload dataset | Test whether the Personality Dataset is uploaded or not into the system | If the Personality Dataset may not uploaded | We cannot do further operations | Personality Dataset uploaded we will do further operations | High | High |
| 02 | load model | Test whether the model is uploaded or not into the system | If the Model may not uploaded | we cannot do  loaded Model | we will do further  operations | High | High |
| 03 | send to cloud | Test whether send to cloud or not | If the cloud is  Not sent | We cannot do send to cloud operations | Sent the cloud operations we will do further | High | High |
| 04 | fall detection | Verify the fall detection or not | Without  Fall detected | we cannot  detected | we can fall detected | High | High |
| 05 | close | Verify the application is closed or not | Without closed | we cannot closed | we can closed | High | High |

**SCREEN SHOTS**:

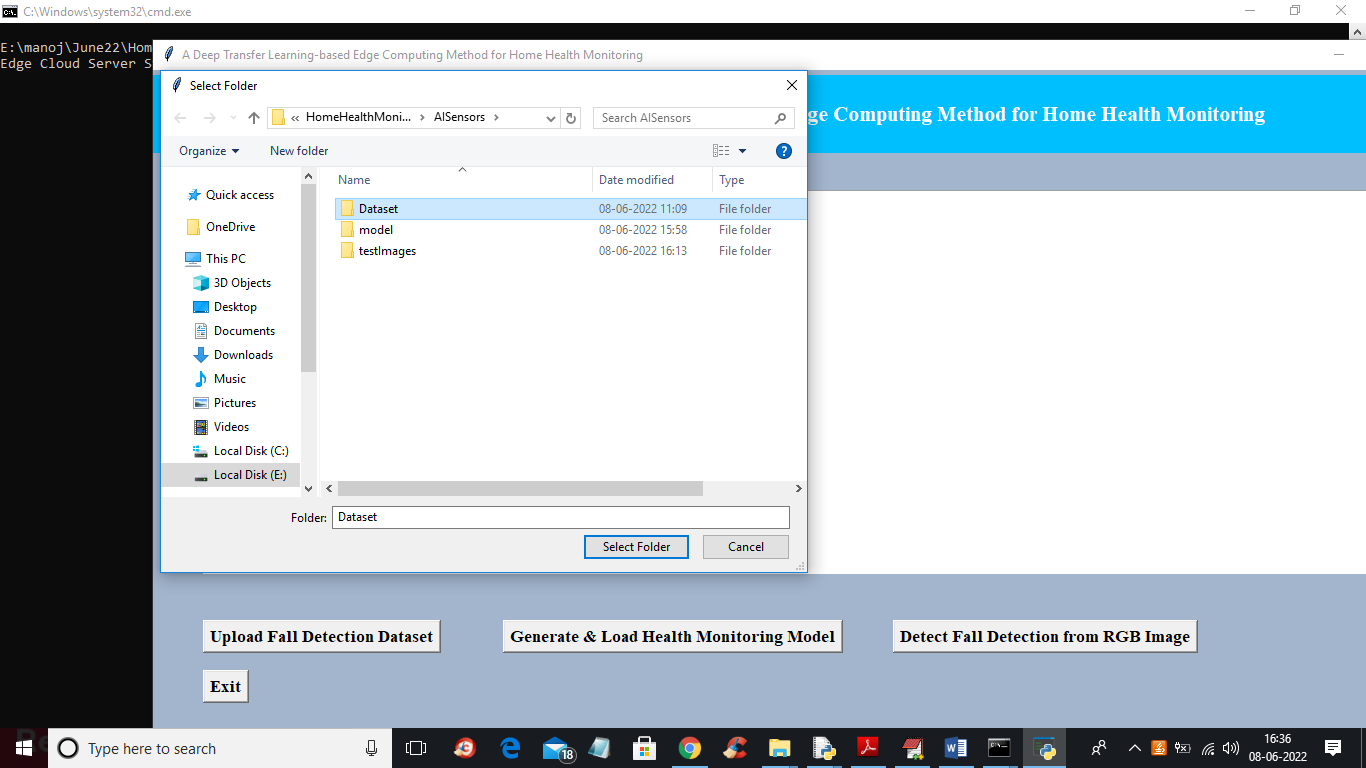
To run project first double click on ‘run.bat’ from ‘EdgeCloudServer’ folder to start cloud server and get below screen



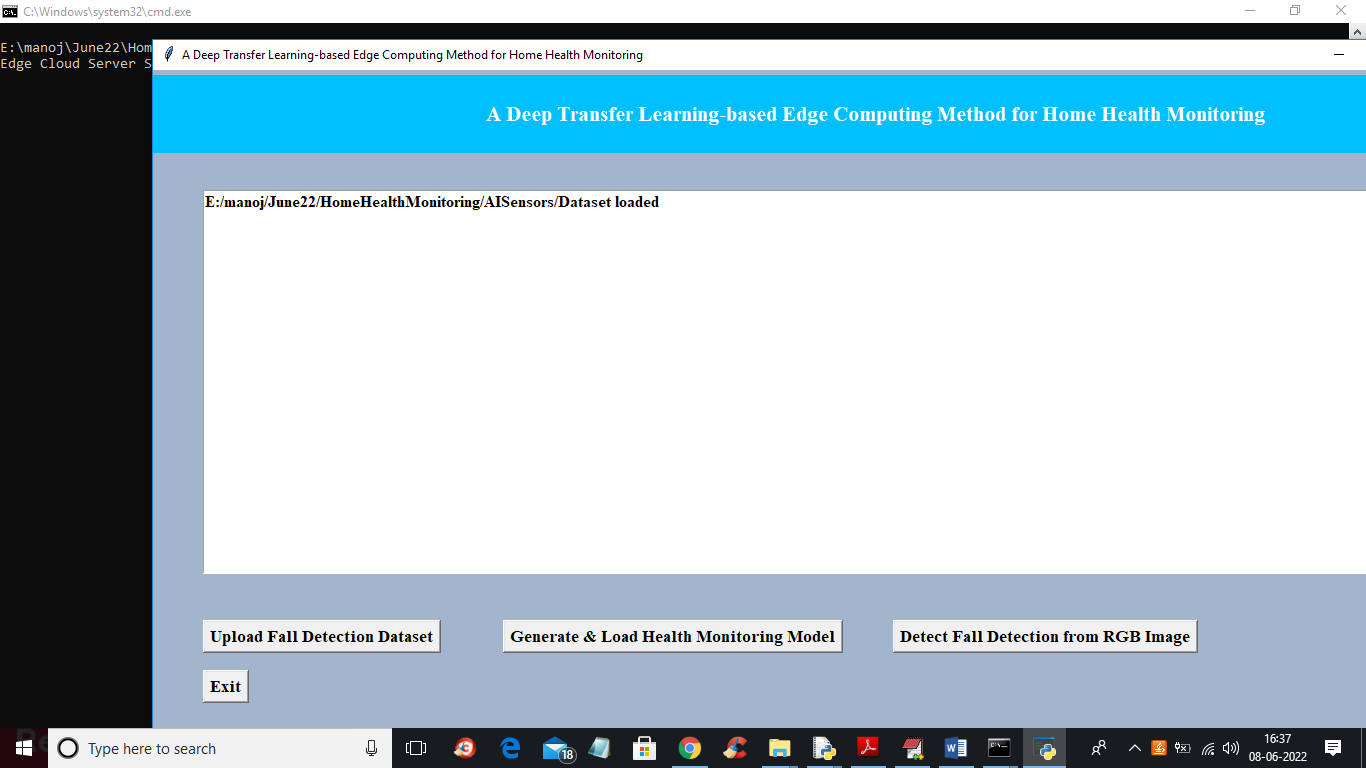
In above screen we can see edge cloud server started and now double click on ‘run.bat’ file from ‘AISensors’ folder to start client sensor application and get below output



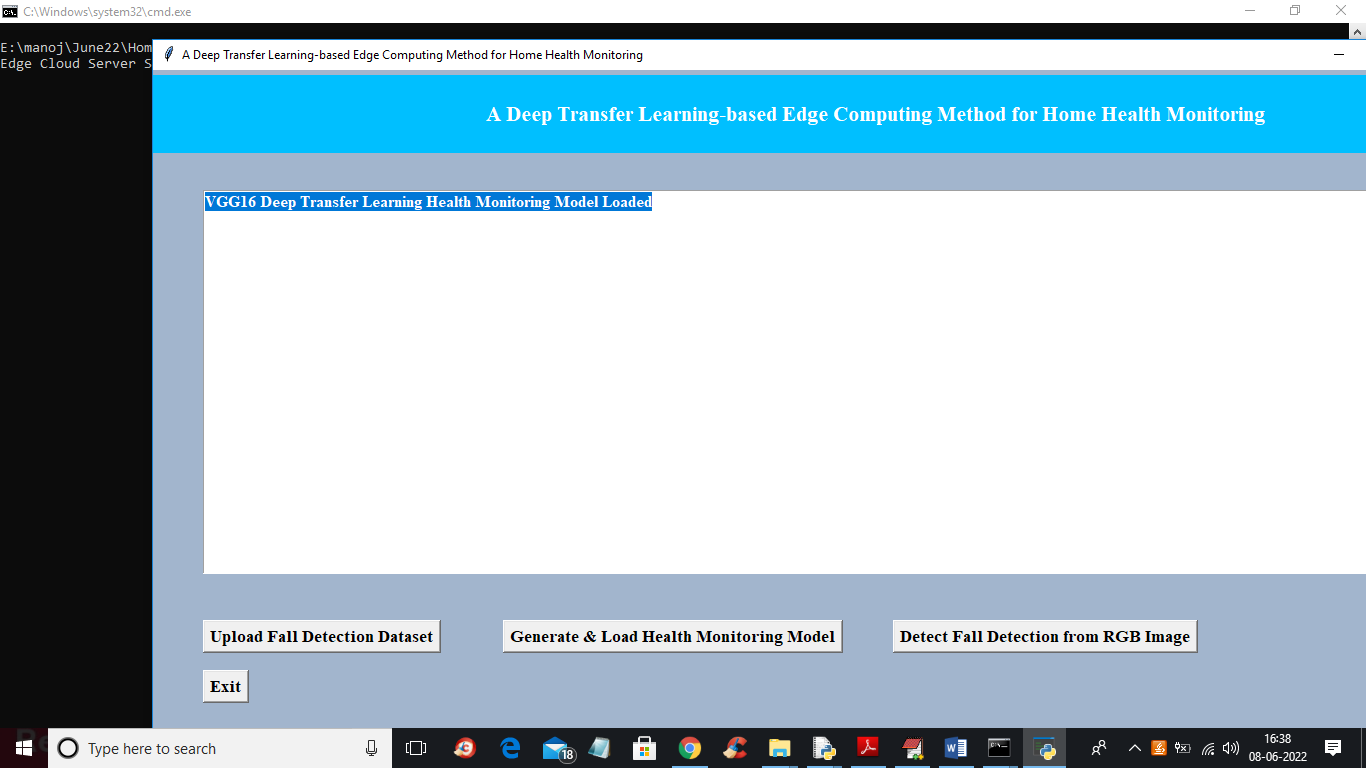
In above client application user can click on ‘Upload Fall Detection Dataset’ button to upload dataset and get below output



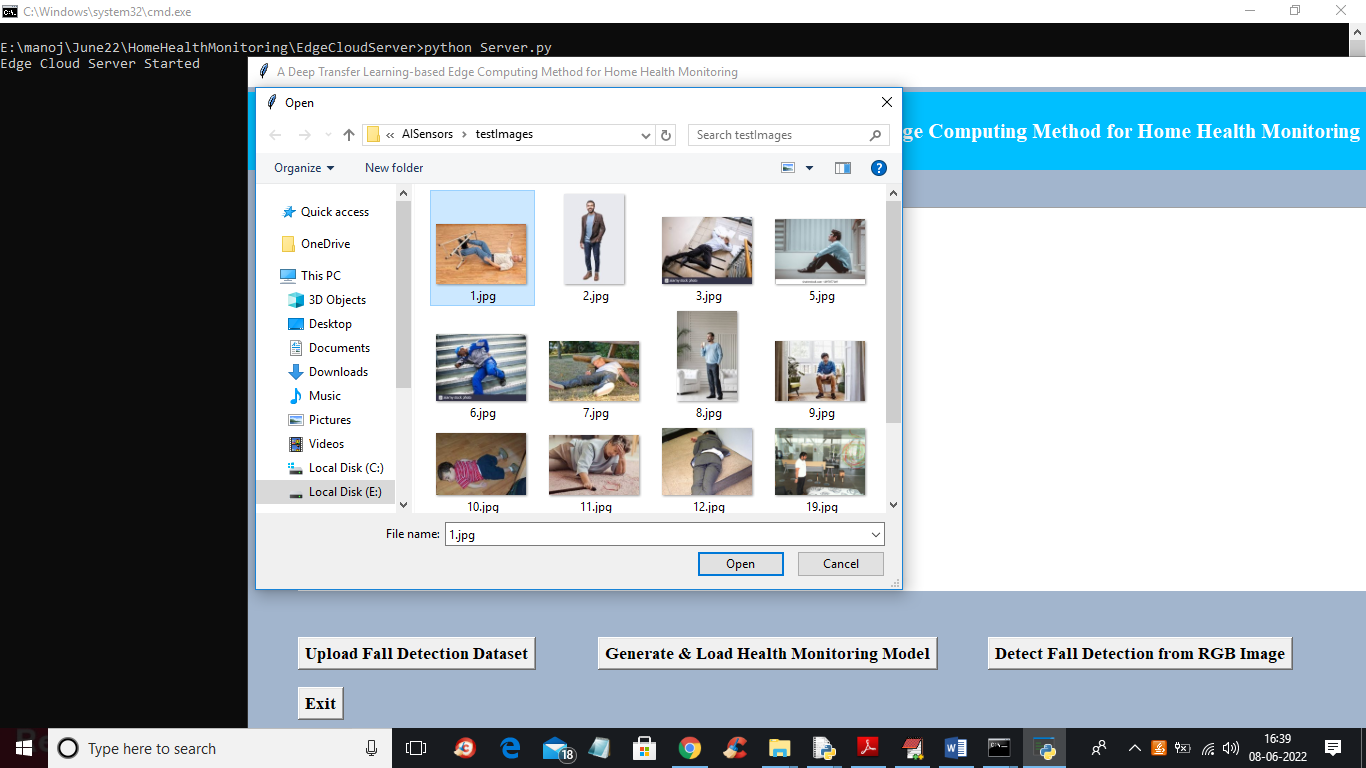
In above screen selecting and uploading ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and get below output



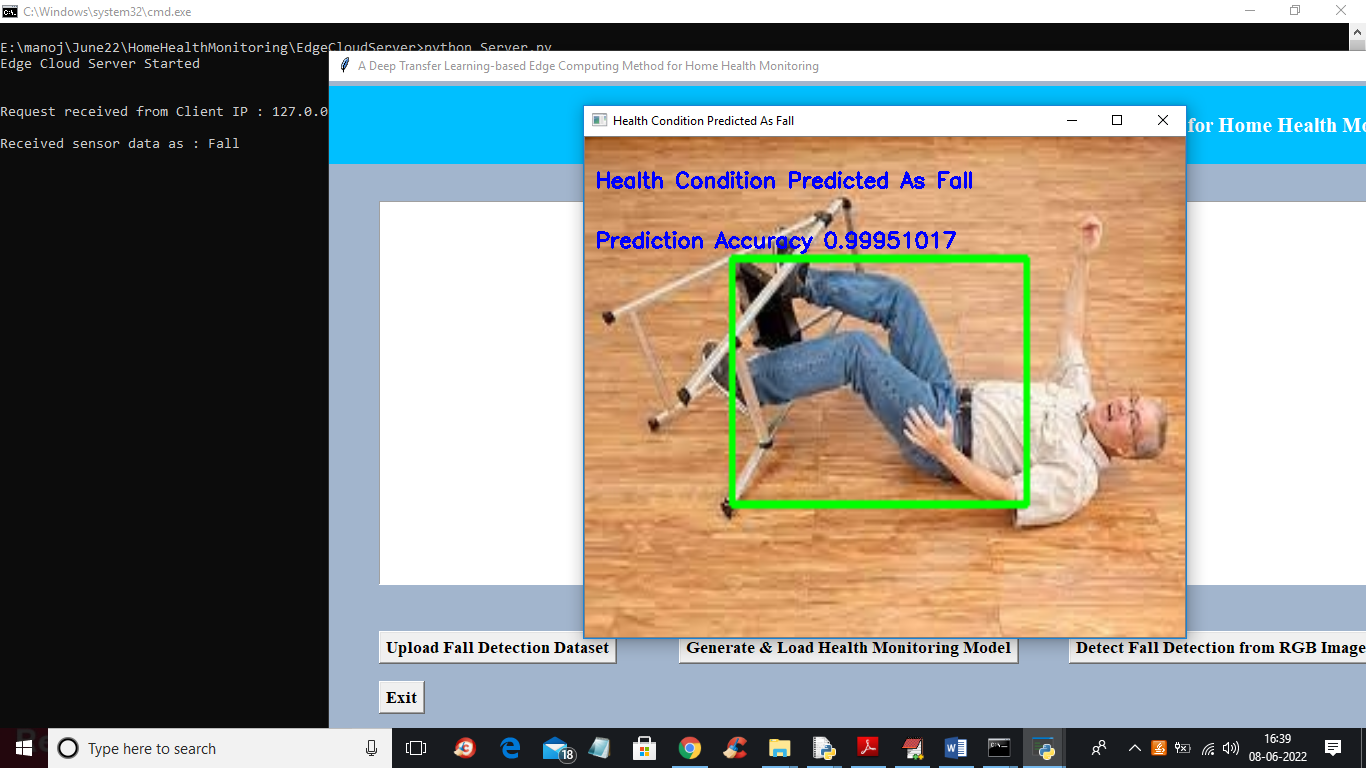
In above screen dataset loaded and now click on ‘Generate & Load Health Monitoring Model’ button to load model and get below output



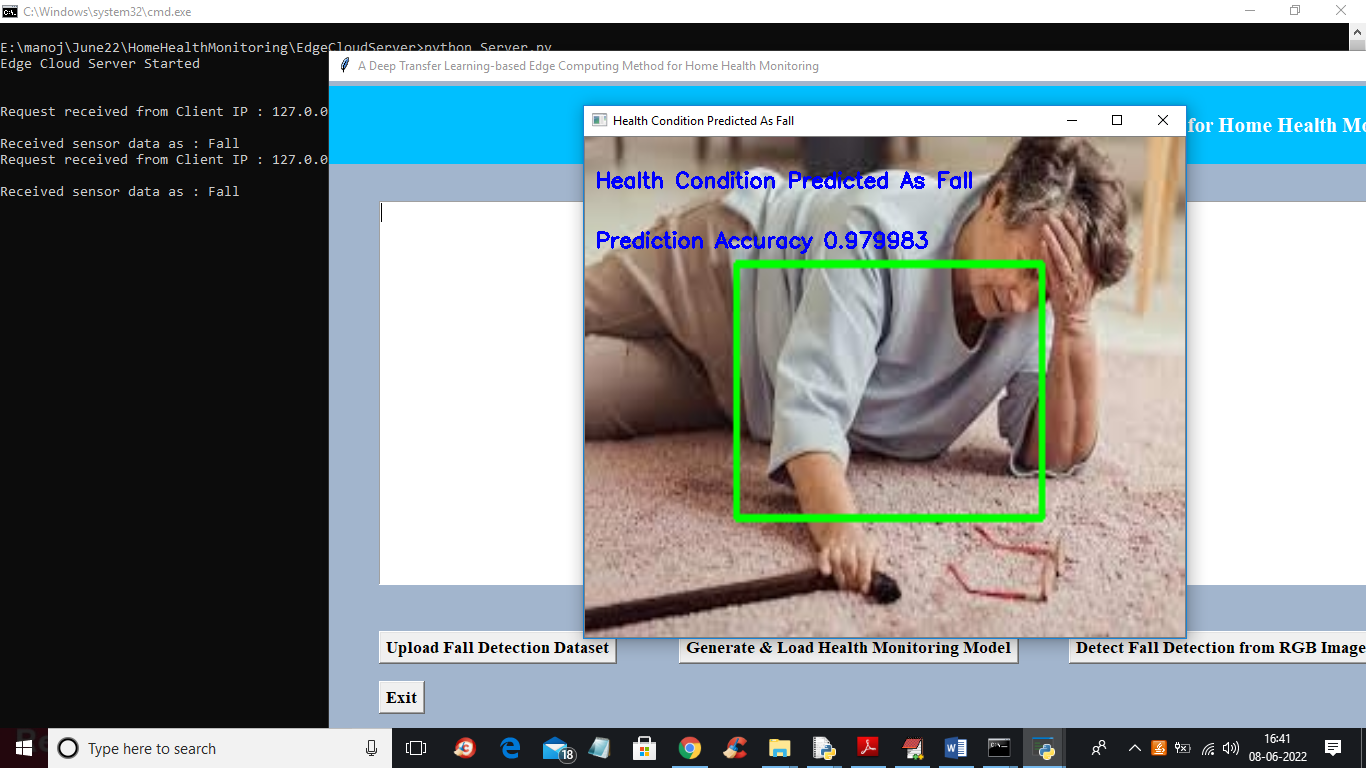
In above screen VGG16 transfer learning model loaded and now click on ‘Detect Fall Detection from RGB image’ button to upload image and get below output



In above screen selecting and uploading ‘1.jpg’ file and then click on ‘Open’ button to get below prediction result



In above screen in uploaded image in blue colour text we can see patient in image condition predicted as Fall with accuracy 0.99 and the same output is report to cloud server which we can see in black console and now test other image

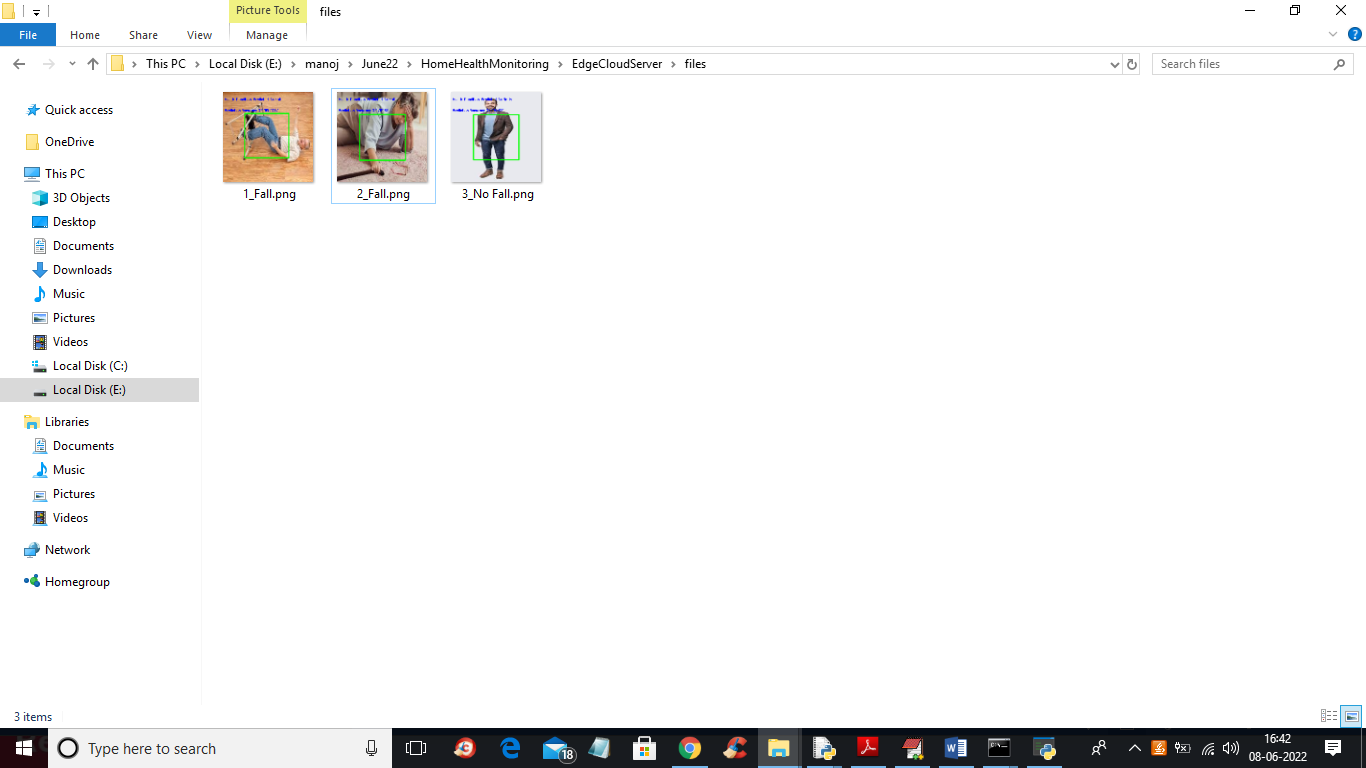


In above image also AI predicted as FALL and now test other images



In above screen patient condition predicted as NO FALL and similarly you can upload and test other images.

All the above uploaded image we can see inside cloud ‘files’ folder with result for future monitoring



In above cloud hospital folder all uploaded images from sensors will be saved.

So by using this application patient can be monitored from home and not require any resources from hospital

**8. CONCLUSION**

To mitigate the health crises in a pandemic or to take care elders in an affordable way, home health monitoring would be very beneficial. In this article, we have proposed a computer vision-based method where a deep transfer learning is used in edge devices as edge computing. In this approach, the raw visual data continuously capture by visual sensor(s) is not required to be sent outside of home. Therefore, privacy, data security as well as latency are not big issues.

**9. REFERENCES**

[1] G. Kapoor, A. Sriram, J. Joshi, A. Nandi, and R. Laxminarayan, “Covid19 in india : State-wise estimates of current hospital beds, intensive care unit (icu) beds and ventilators,” in CDDEP, Princeton University, 2020.

[2] F. Blavin and D. Arnos, “Hospital readiness for covid-19,” 2020. [Online]. Available: https://www.urban.org/sites/default/files/ publication/101864/hospital-readiness-for-covid-19 2.pdf

[3] U. Nations, “World population ageing report,” 2019. [Online]. Available: <https://www.un.org/en/sections/issues-depth/ageing/>

[4] B. Shneiderman, “Human-centered artificial intelligence: Reliable, safe & trustworthy,” International Journal of Human–Computer Interaction, vol. 36, no. 6, pp. 495–504, 2020.

[5] J. He, S. L. Baxter, J. Xu, J. Xu, X. Zhou, and K. Zhang, “The practical implementation of artificial intelligence technologies in medicine,” Nature medicine, vol. 25, no. 1, pp. 30–36, 2019.

[6] C. Xu, M. Dong, K. Ota, J. Li, W. Yang, and J. Wu, “Sceh: smart customized e-health framework for countryside using edge ai and body sensor networks,” in 2019 IEEE global communications conference (GLOBECOM). IEEE, 2019, pp. 1–6.

[7] R. L. Fritz and G. Dermody, “A nurse-driven method for developing artificial intelligence in “smart” homes for aging-in-place,” Nursing outlook, vol. 67, no. 2, pp. 140–153, 2019.

[8] G. Yang, Z. Pang, M. J. Deen, M. Dong, Y. Zhang, N. H. Lovell, and A. M. Rahmani, “Homecare robotic systems for healthcare 4.0: Visions and enabling technologies,” IEEE Journal of Biomedical and Health Informatics, 2020.

[9] L. P. Malasinghe, N. Ramzan, and K. Dahal, “Remote patient monitoring: a comprehensive study,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 1, pp. 57–76, 2019.

[10] S. Sathyanarayana, R. K. Satzoda, S. Sathyanarayana, and S. Thambipillai, “Vision-based patient monitoring: a comprehensive review of algorithms and technologies.