A PROJECT REPORT

ON

SKELETON BASED HUMAN ACTION RECOGNITION SYSTEM AND ALERTS

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

BY

SHARVIL BAKSHI [B190138509]
SUNNY BARVE [B190138512]
CHETANA DUSANE [B190138525]
RITESH JADHAV [B190138535]

UNDER THE GUIDANCE OF

PROF. YOGITA H. KHAIRNAR

DEPARTMENT OF INFORMATION TECHNOLOGY
K.K.WAGH INSTITUTE OF ENGINEERING EDUCATION AND RESEARCH,
NASHIK - 422003

2023-2024

K. K. Wagh Education Society's

K.K. Wagh Institute of Engineering Education & Research

NASHIK - 422 003.

Certificate

This is to certify that Sharvil Shailesh Bakshi *and others have / has succesfully completed the project work On* Skeleton based Human Action Recognition System and Alerts *in Partial fulfilment of the Bachelors Degree in* Information Technology *Engineering during the academic year* 2023 - 24

PRINCIPAL

Department of Information Technology
K.K. Wagh Institute of Engineering Education & Research
Nashik -3

2023-2024

CERTIFICATE

This is to certify that the project report entitled **"Skeleton Based Human Action Recognition System and Alerts"** being submitted by **Sharvil Bakshi** (B190138509), **Sunny Barve** (B190138512), **Chetana Dusane** (B190138525), **Ritesh Jadhav** (B190138535) is a record of bonafide work carried out by him/her under the supervision and guidance of **Prof. Yogita H. Khairnar** in fulfillment of the requirement for **BE (Information Technology Engineering) – 2019 course** of Savitribai Phule Pune University, Pune in the academic year 2023-2024.

Date:

Place: Nashik

Prof. Yogita H. Khairnar
Guide

Dr. Preeti D. Bhamre
Head of the Department

This project report has been examined by us as per the Savitribai Phule Pune University, Pune, requirements at K. K Wagh Institute of Engineering Education & Research, Nashik on. . . . . .

(Name & Signature) (Name & Signature) Internal Examiner External Examiner

Date: 22.05.24

Sponsorship Certificate

This is to certify that the project entitled

Skeleton based Human action recognition system and alerts

was sponsored by Inaiways Technologies Pvt. Ltd
(Reg. Add: E2, Omega Heritage Society, DSK Road, Dhayari, Sinhgad Road, Pune 411041)

We Inaiways Technologies Pvt. Ltd have sponsored project idea Skeleton based Human action recognition system and alerts to following students for their final year Engineering (BEIT) graduation project purpose during academic year 2023-24.

1. SHARVIL BAKSHI - B190138509
2. SUNNY BARVE - B190138512
3. CHETANA DUSANE - B190138525
4. RITESH JADHAV - B190138535

They have contributed to best of their extent for implementation for the project idea under supervision of Prof. Yogita H. Khairnar.

Best regards

Inaiways Technology Pvt. Ltd. Pune

Mr Mayur Jadhav
HRD, Inaiways Technologies Pvt. Ltd
Mob. 9823092923

ACKNOWLEDGEMENT

I would like to thank to our Head of my Department Prof. Dr. Preeti D. Bhamre for her valuable and constructive suggestions during development of this project. Then I would like to thank to our Project guide Prof. Yogita H. Khairnar for her valuable guidance which has been the one that helped me patch this project and make it full proof success her suggestions and her instruction has served as the major contributor towards the completion of the project. Then I would like to thank our teachers who helped us in many difficulties to make this project successful. Then I would like to thank my friends who have helped us with their valuable suggestion and guidance has been helpful in various phases of the completion of the project.

Sharvil Bakshi [B190138509]
Sunny Barve [B190138512]
Chetana Dusane [B190138525]
Ritesh Jadhav [B190138535]

Contents

List of Figures

List of Tables

Abstract

In the pursuit of promoting safety and well-being, a camera-based system has been devised to analyse human skeletal movements, particularly focusing on the elderly demographic. The primary objective is to discern and monitor these movements in real-time through the utilization of image or video data. A novel deep convolution long short-term memory (ConvLSTM) network is proposed for skeletal-based activity recognition and fall detection. The Convolution Neural Network LSTM network is a sequential fusion of convolution neural networks (CNNs), long short-term memory (LSTM) networks, and fully connected layers. The acquisition system employs human detection and pose estimation to pre-calculate skeleton coordinates from the image/video sequence. The raw skeleton coordinates, along with their characteristic geometrical and kinematic features, are used to construct novel guided features in the ConvLSTM model. These spatio-temporal guided features are obtained using a trained multi-player CNN-LSTM combination, and a classification head, including fully connected layers, is subsequently applied. Deep learning algorithms are utilized for recognizing and classifying various actions performed by individuals, with an emphasis on timely and efficient detection. Experimental results are compared against the performance of isolated Convolution Neural Networks and LSTM networks. Through the synergy of computer vision and deep learning, safety is enhanced via real-time action recognition and anomaly detection, emphasizing proactive monitoring for the well-being of elderly individuals.

Keywords: Camera-based system, human skeletal movements, ConvLSTM, spatio-temporal guided features, CNN-LSTM, computer vision and anomaly detection.

Chapter 1

Introduction to Human Action Recognition System

This Chapter introduces the project titled Skeleton Based Human Action Recognition and Alerts , outlining its aim and motivation. The scope of the project is defined, delineating its boundaries and key coverage areas. Clear objectives are established, guiding the direction of the project. The report structure and organization are presented, offering a roadmap for the chapters to follow.

1.1 Aim/Motivation

The aim of this project is to enhance safety and well-being, particularly focusing on elderly individuals. Motivated by the imperative to address challenges related to the real-time monitoring of human skeletal movements, the project seeks to develop an innovative camera-based system. This system aims to detect and analyse movements using image or video data, providing timely insights into potential anomalies or fall events. By leverage advanced technologies such as deep convolution neural network long short-term memory (ConvLSTM) networks, the project aspires to contribute to proactive monitoring and improved safety measures for the elderly population.

1.2 Scope

The scope of this project is defined by its commitment to addressing the challenges associated with real-time monitoring of human skeletal movements, with a specific focus on the elderly demographic. The project encompasses the development of a camera-based system capable of analysing image or video data to detect and monitor movements. The system's scope extends to the timely recognition of anomalies and fall events, contributing to the overall safety and well-being of elderly individuals. While the project acknowledges certain limitations, such as dependencies on camera quality and challenges related to data transformation. Also, it

aims to focus on skeleton of the human body so that it will be easy to recognize any complex human body posture.

1.3 Objective(s)

The objectives of this project are strategically outlined to guide the development of a robust camera-based system for real-time monitoring of human skeletal movements, specifically tailored for elderly individuals. The key objectives include:

1. Real-time Movement Detection including algorithms and processes to enable the system to detect and analyse skeletal movements in real- time using image or video data.

2. Anomaly Recognition to implement mechanisms for the timely recognition of anomalies and fall events, with a focus on enhancing the safety and well-being of elderly individuals.

3. Integration of Convolution Neural Network Long Short Term Memory Networks that incorporates deep convolution long short-term memory (ConvLSTM) networks into the system architecture to harness advanced machine learning techniques for improved accuracy in action recognition.

4. An alert generating mechanism which sends alert to the verified user via Short Message System (SMS) on any abnormal activity or fall detected.

5. Development of a human fall detection system which must detect real time falls or any abnormal movements caused to elderly people.

1.4 Report Organization

The organization of this report is structured to provide a comprehensive and coherent exploration of the Skeleton Based Human Action Recognition and Alerts . The following is a brief overview of the report's organization:

1. **Introduction:** Chapter 1 introduces the project, outlining its aim, scope, objectives, and the overall organization of the report. Also, gives a preface about the overall activities carried out int the project.

2. **Literature Review:** Chapter 2 delves into the existing body of knowledge related to Skeleton Based Human Action Recognition and Alerts . It explores related work and existing systems to establish a context for the project. It includes referred papers from different conferences all over the world. Wherein, the existing system identifies the gaps within

the current system that can be addressed.

3. **System Proposed Architecture:** Chapter 3 details the proposed architecture of the system. This includes a precise definition of the problem, the methodology adopted, and a block diagram illustrating the architecture. This chapter flashes the overall skeleton of the project through the block-diagram.

4. **Project Requirement Specifications:** Chapter 4 articulates the specific requirements that the project aims to complete, providing a detailed specification for the development phase. It mentions the hardware and software requirements of the project which are implemented during the project tenure.

5. **High-Level Design of Project ERD/DFD/UML Diagrams:** Chapter 5 focuses on the high-level design aspects of the project, presenting Entity-Relationship Diagrams (ERD), Data Flow Diagrams (DFD), and Unified Modelling Language (UML) diagrams. Chapter briefs about the diagrams and their explanation in the project which provides a detail view of the overall architecture to the user.

6. **Experimental Set-up / Simulation:** Chapter 6 outlines the experimental set-up and simulation procedures undertaken in the project, offering insights into the methodologies employed for testing and validation. This includes the algorithm and the steps to carry out the project implementation.

7. **Project Planning (Semester-I Semester-II):** Chapter 7 provides a comprehensive overview of the project planning, detailing the milestones and activities conducted during both Semester-I and Semester-II. This is the pictorial representation of the project with a diagram called Gantt-Chart.

8. **Conclusions:** Chapter 8 offers conclusive remarks, summarizing the key findings, contributions, and potential avenues for future work. This chapter winds up the project by giving short description about each chapter and the work carried out in that.

In the following chapter, we will conduct an extensive literature review to explore existing research and systems related to Skeleton Based Human Action Recognition, providing a foundation for our project's context and significance.

Chapter 2

Literature Review

This chapter immerses in a comprehensive literature review, unveiling the existing body of knowledge surrounding Human Action Recognition. Through an exploration of related work and existing systems, this chapter establishes the context for our project.

2.1 Related Work Done

1) Lisa Schrader, et al.
Advanced Sensing and Human Activity Recognition in Early Intervention and Rehabilitation of Elderly People (Date: 24th February 2020)

Methodology: - This study aimed to pioneer advanced sensing techniques coupled with human activity recognition for early intervention and rehabilitation in elderly individuals. The methodology involved deploying a network of wireless, battery-less sensors strategically placed in living spaces. These sensors captured detailed human movements, which were then processed using machine learning algorithms. The algorithms focused on recognizing specific activities such as sitting, standing, and walking. The output was a Human Activity Recognition (HAR) system that used wireless sensors and machine learning to monitor the activities of elderly individuals in real-time.[1]

2) Santosh Kumar Yadav Kamlesh Tiwari Hari Mohan Pandey Shaik
" Skeleton-based human activity recognition using Convolution Neural Network LSTM and guided feature learning " (Date: 1st September 2021)

Methodology: - This study proposed a novel methodology for skeleton-based human activity recognition. The Convolution Neural Network LSTM network was employed, representing a fusion of convolution neural networks (CNNs), long short-term memory (LSTM) networks, and fully connected layers. The acquisition system used human detection and pose estimation to pre-calculate skeleton coordinates. These coordinates, along with their geometric and kinematic features, formed the basis for guided feature

learning.[2]

3) Thi Thi Zin , Ye Htet , Yuya Akagi , Hiroki Tamura , Kazuhiro Kondo , Sanae Araki and Etsuo Chosa
" Real-Time Action Recognition System for Elderly People Using Stereo Depth Camera " (Date:- 1 Sept 2021)

Methodology: - This paper contributed to a real-time action recognition system tailored for elderly individuals. The methodology leveraged a stereo depth camera for precise and real-time monitoring. The system detected and classified actions based on the skeletal information extracted from the depth images. The output was a real-time action recognition system that could effectively monitor and respond to the activities of elderly individuals.[3]

4) Anusha Ganesan , Anand Paul , and HyunCheol Seo
Elderly People Activity Recognition in Smart Grid Monitoring Environment (Date: - 22 March 2022)
Methodology: - This work addressed activity recognition for elderly people within a smart grid monitoring environment. The methodology included the use of simplified patterns for skeleton-joint models to analyse small data quantities. A skeleton algorithm was applied, considering accuracy, resilience to noise, and the generation of a linked skeleton for improved representation. The parameters connected to the position and movement of joints were critical in determining the human body's overall position and movement. The methodology aimed at developing an effective activity recognition system within the context of a smart grid monitoring environment.[4]

5) Paka Akshaja, Pushpendra Kumar Pateriya
Health Care Monitoring System using Action Recognition (Date: 5 May 2022) International Journal of Creative Research Thoughts (IJCRT), Volume 10, Issue 5, ISSN: 2320-2882

Methodology: - This paper focused on developing a Health Care Monitoring System based on action recognition. The methodology involved the use of independently-recurrent neural networks (IndRNN) for real-time human action recognition. The system utilized various datasets for training, validation, and testing, aiming to accurately classify different actions. The output was a robust monitoring system capable of identifying and analysing various health-related actions, providing valuable insights for caregivers.[5]

6) Han Sun, Yu Chen.
Real-Time Elderly Monitoring for Senior Safety by Lightweight Human Action Recognition (Date : 21st July 2022)

Methodology: - The research emphasized real-time monitoring of elderly individuals for senior safety through lightweight human action recognition. The methodology incorporated independently-recurrent neural networks (IndRNN) for effective action recognition with minimal computational complexity. The system required a high-quality camera to ensure accurate operation. The output was the introduction of a Real-Time Elderly Monitoring System (REMS), utilizing lightweight action recognition techniques.[6]

7) Ayman Ali, Ekkasit Pinyoanuntapong, Pu Wang, Mohsen Dorodchi
Skeleton based Human Action Recognition via Convolution Neural Networks (CNN) (Date: 31 Jan 2023)

Methodology: - This study introduced a methodology for human action recognition based on skeletal information using Convolution Neural Networks (CNN). The approach involved a combination of CNNs and Graph Neural Networks (GNNs) to effectively capture and represent the human body's skeletal structure. Various data augmentation techniques were applied to enhance the model's robustness. The methodology underscored the importance of training techniques, data augmentations, and optimizers in improving CNN-based action recognition models.[7]

2.2 Existing Systems

The Existing System study introduces the methodology for human action recognition based on skeletal information using CNN. The training of model using Convolution Neural Networks (CNNs) and Graph Neural Networks (GNNs) which made it possible to achieve the goal of fall detection through image and video. Addition of alert generation to the system made it more user friendly. The project clearly aims to address the gaps in the existing human skeleton detection and recognition for alerts which is a quite challenging task as human body postures are different and varies from human to human. Also, alert generation is a noteworthy update that makes user more friendly towards the system.

Moving forward, Next Chapter will elucidate the proposed architecture of the system, delving into the problem definition, methodology, and a detailed block diagram.

Chapter 3

Systems Proposed Architecture

In this Chapter, we delve into the core of our projectthe proposed architecture. Here, we articulate the problem definition, methodology, and present a detailed block diagram that outlines the architecture of our system.

3.1 Problem Statement

The project, titled "Skeleton-Based Real-Time Action Recognition System Alerts," aims to create a real-time action recognition system for elderly individuals using skeleton-based computer vision techniques. The purpose is to develop a real-time action recognition system for elderly individuals and enable it to send alerts when unusual or concerning actions are detected.

3.2 Methodology

This section defines the detailed steps for implementation of this project.

1. **Data Collection:** Information gathering comes first. This stage is crucial since both quantity and quality have an impact on the system's overall performance. In essence, it is a process for gathering data on particular parameters.

2. **Data Preparation:** Data gathering is the first phase, and data preparation is the next. It is a process that turns useless data into information that may be utilized in decision-making. This approach is also known as data purification.

3. **Choose a Model:** One chooses the CNN algorithm for the task to represent pre-processed data in a model. Train the Model: In machine learning (ML), supervised learning is used to train a model to increase prediction or judgement accuracy. A few parameters are required to evaluate the model. The given goals determine the variables.

4. **Evaluate the Model:** The performance of the model in comparison to the previous one must also be recorded. Several parameters are required to evaluate the model. The given goals determine the variables.

5. **Parameter Tuning:** Initialization parameters, distribution, performance, learning rate, and the total number of training steps may all be present at this point.

6. **Making Predictions:** To compare the constructed model to the real world, a prediction on the test dataset is required. If the outcome is in line with the forecasts of subject matter experts or experts who have a close relationship with the outcome, the model can be utilized to generate additional predictions.

3.3 Block Diagram

Figure 3.1 shows the Block Diagram for this Project. It visually describes the methodological steps. In the first part of the project the input is taken from Camera through the timestamped images then the dataset is set for training and testing.

In the Second part of this project, Machine Learning model on input dataset is created, which is further tested and evaluated. At last, Real - time data will be provided to this trained model and it will send alert to the user in the format of e-mail.

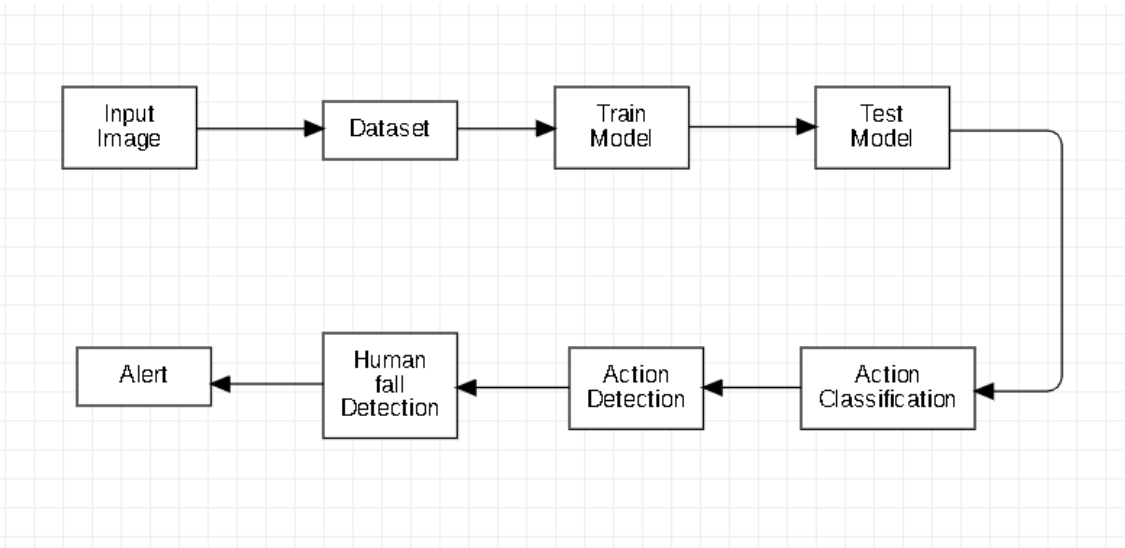

Figure 3.1: Block Diagram

3.4 Algorithm

This section defines the algorithmic steps for execution of this project.

Step 1 : Start
Begin the fall detection and alert system algorithm.

Step 2 : Data Collection
Gather a dataset of skeleton-based images, consisting of around 140 fall and 140 non-fall images, ensuring diversity in poses and backgrounds.

Step 3 : Preprocessing
Pre-process the images to extract skeleton information using techniques like key-point detection or pose estimation.

Step 4 : Feature Extraction
Use Convolution Neural Network (CNN) models to extract features from the skeleton-based images, capturing important patterns for fall detection.

Step 5 : Model Training (CNN)
Train the CNN model on the preprocessed dataset to classify between fall and non-fall instances, optimizing for accuracy and generalization.

Step 6 : Graph Neural Network (GNN)
Implement a GNN to analyse the spatial relationships between key-points in the skeleton data, enhancing the model's understanding of human poses.

Step 7 : Convolution LSTM (ConvLSTM)
Incorporate ConvLSTM layers to capture temporal dependencies in the skeleton sequence data, enabling the model to recognize falling patterns over time.

Step 8 : Integration
Integrate the CNN, GNN, and ConvLSTM components into a unified model architecture, leveraging their complementary strengths for robust fall detection.

Step 9 : Video Analysis
Extend the model to analyse video streams, detecting instances of falls by processing consecutive frames and identifying abrupt changes in skeletal poses.

Step 10 : Alert Generation
Implement a mechanism to generate alerts when a fall is detected, including timestamp information from the video stream.

Step 11 : Alert Delivery
Send alerts via email to designated recipients, providing timely notifications of potential fall incidents and ensuring prompt response.

Step 12 : End
Finish the fall detection and alert system algorithm.

In the following chapter, we will meticulously outline the specific requirements of our project, establishing a comprehensive specification for the development phase.

Chapter 4

Project Requirement Specifications

This chapter shifts our focus to the specific requirements our project aims to fulfil. Providing a detailed specification for the development phase, this chapter sets the stage for the subsequent design and implementation phases.

4.1 Hardware Requirements

1. Camera (Simple web-cam) :

High-resolution camera(s) capable of capturing clear images. Multiple cameras for comprehensive coverage and redundancy. Optionally, infra-red (IR) cameras for low-light or night-time monitoring.

2. Computer or Processing Unit (Monitor and CPU) :

Powerful CPU or GPU for real-time image processing and deep learning computations. Consider high-performance processors with multiple cores for parallel processing.

3. Internet Connectivity (Network for Alert generation) :

Ethernet switches or routers for network connectivity between cameras and processing units. Ensure sufficient bandwidth for real-time data transmission.

4. Memory (RAM) :

Sufficient RAM to accommodate the processing of large image datasets. Opt for a minimum of 16 GB RAM, with scalability options for future expansions.

4.2 Software Requirements

1. Operating System (Windows 10, Linux, MacOS) :

Choose a stable and reliable operating system compatible with your hardware architecture. Options include Windows, Linux distributions (e.g., Ubuntu, CentOS), or specialized real-time operating systems (RTOS) for embedded systems.

2. Programming Language (Python) :

Python for implementing machine learning algorithms, image processing, and data analysis. Consider using libraries like Tensor-Flow, PyTorch, or OpenCV for deep learning and computer vision tasks.

3. Development Framework (TensorFlow, OpenCV) :

Tensor-Flow or PyTorch for building and training convolution neural networks (CNNs) and long short- term memory (LSTM) networks. Utilize frameworks optimized for GPU acceleration to leverage hardware resources efficiently.

4. Development Environment (IDE) :

Integrated Development Environment (IDE) for software development and programming. Popular choices include Visual Studio Code, PyCharm, or Jupyter Notebook for Python-based development.

5. Communication Tools (for alerts) :

Implement communication protocols such as TCP/IP or MQTT for real-time data exchange between cameras and processing units. Utilize network programming libraries in your chosen programming language for seamless integration.

4.3 Functional and Non Functional Requirements

In the previous chapter, problem definition, detailed methodology or implementation steps and block diagram for the project have been discussed. In this chapter, hardware and software requirements, functional and non-functional requirements have been discussed.

4.3.1 Functional Requirements

1) **Data Collection:** The system must be able to collect a diverse dataset comprising images or video frames depicting both falls and non-fall activities.Data collection must cover various environments, lighting conditions, and fall scenarios relevant to the elderly population.

2) **Data Preprocessing:** The system must pre-process the collected data, including noise removal, image normalization, and labelling of falls and non-falls.

3) **Model Architecture:** Implement a CNN architecture suitable for image-based fall detection. Consider the complexity and efficiency of the model.

4) **Training:** Train the CNN model on the preprocessed dataset. Implement techniques to avoid over-fitting and ensure robust performance across diverse scenarios.

5) **Testing and Validation:** Implement comprehensive testing procedures, including unit tests, integration tests, and real-world validation to ensure the system's accuracy and reliability.

6) **Real-Time Detection:** The system must be capable of real-time fall detection, analysing frames or images as they are captured.

7) **Alerting and Notification:** Implement an alerting mechanism to notify caregivers, emergency services, or relevant authorities in the event of a detected fall.

8) **User Interface:** Develop a user-friendly interface for monitoring the system's status, accessing alerts, and configuring settings.

4.3.2 Non-Functional Requirements

1) **Performance:** The system should achieve a minimum accuracy of 95% in fall detection under various conditions. Fall detection should occur within a maximum latency of 2 seconds from the occurrence of the fall event.

2) **Usability:** The user interface should be intuitive and accessible, catering to users with limited technical knowledge.

3) **Security:** Ensure data privacy with unique user credentials. Implement security measures(SMS Verification)to prevent unauthorized access and tampering of the system.

4) **Scalability:** Design the system architecture to be scalable, allowing for the addition of more sensors or devices if required.

5) **Reliability:** The system should operate reliably without frequent failures or false alarms, ensuring continuous monitoring.

Moving forward in the next chapter, we will shift our focus to the high-level design of the project, incorporating UML diagrams to provide a visual representation of our system's architecture.

Chapter 5

High Level Design of Project

High-Level Design refers to the process of creating abstract representations of a system before the actual implementation begins. As we progress, this chapter focuses on the high-level design aspects of our project. Through the lens of UML diagrams, we present a visual representation of our system's architecture.

5.1 UML Diagrams

Unified Modelling Language (UML) diagrams serve as a visual language for understanding, designing, and communicating complex systems. Widely employed in software engineering, UML diagrams provide a standardized and systematic way to represent various aspects of a system architecture and behaviour. These diagrams serve as a bridge between technical and non-technical stakeholders, facilitating effective communication and comprehension throughout the software development life-cycle.

UML offers a diverse set of diagram types, each catering to specific facets of system design and functionality. These diagrams range from depicting the high-level structure of a system to capturing the dynamic interactions between its components. The primary goal of utilizing UML is to enhance clarity, foster collaboration, and establish a shared understanding among project stakeholders. As a visual modelling tool, UML empowers developers, designers, and decision-makers to conceptualize, plan, and implement software systems with a unified and standardized approach. In the subsequent chapters, we will delve into specific UML diagrams, each playing a unique role in unravelling the intricacies of our project.

These diagrams act as a visual guide, enabling us to navigate the complexities of system design, from the overarching structure to the intricate details of interactions and behaviours. Through the exploration of UML diagrams, we aim to enhance comprehension, streamline communication, and pave the way for the successful development and implementation of our project.

5.1.1 Use-Case Diagram

The Use Case Diagram serves as the project high-level blueprint, offering a comprehensive view of the system functionalities from an end-user perspective. It outlines the interactions between external actors and the system, defining the various use cases that propel our project work-flow. The camera takes input from the dataset and then it is checked whether the dataset is valid or not. The system simultaneously trains the model based on the algorithm of convolution neural network and classifies the images for further testing. The user now can fetch alerts based on the human fall detected or not.

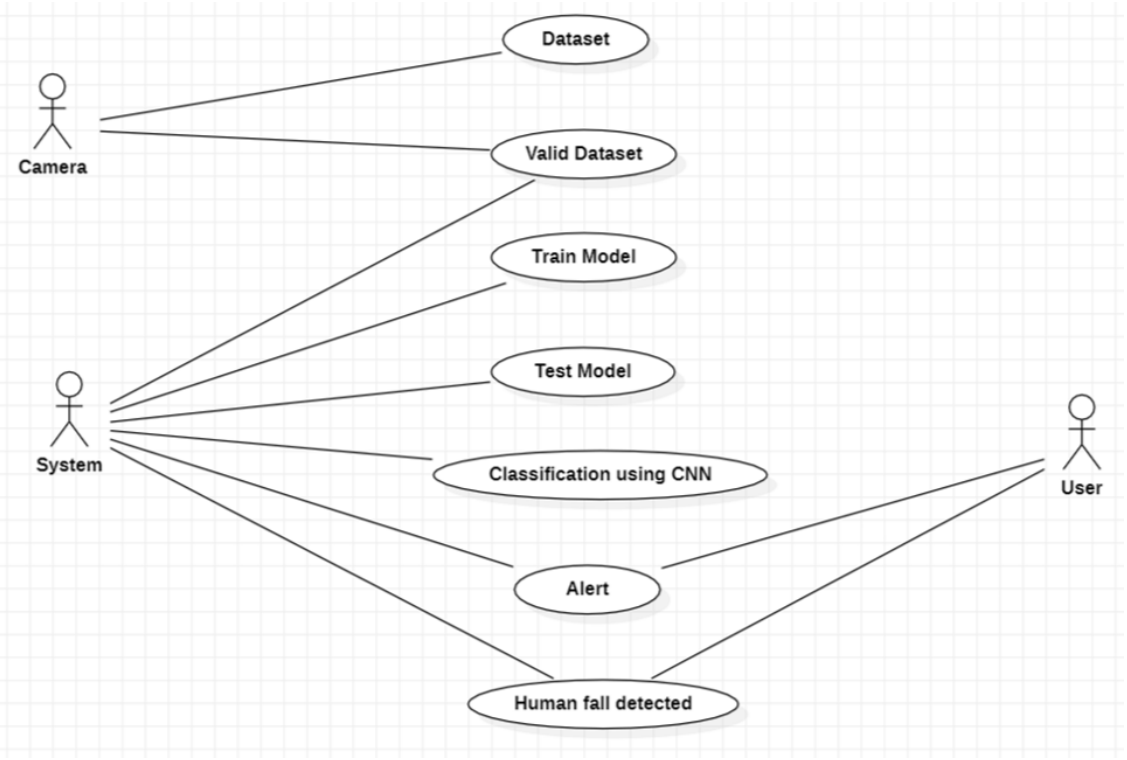

Figure 5.1: Use Case Diagram

Use case diagram is used to describe the function requirements of the system by using the use cases and the actors. In the Figure user and database are the actors into the system. Use cases involved into the system.

5.1.2 Class Diagram

The Class Diagram delves into the structural anatomy of our system, portraying the relationships among classes, their attributes, and associated behaviours. Within this diagram, components such as configuring the Camera, System and User processing emerge as distinct classes, each encapsulating properties and methods essential for seamless system operation.

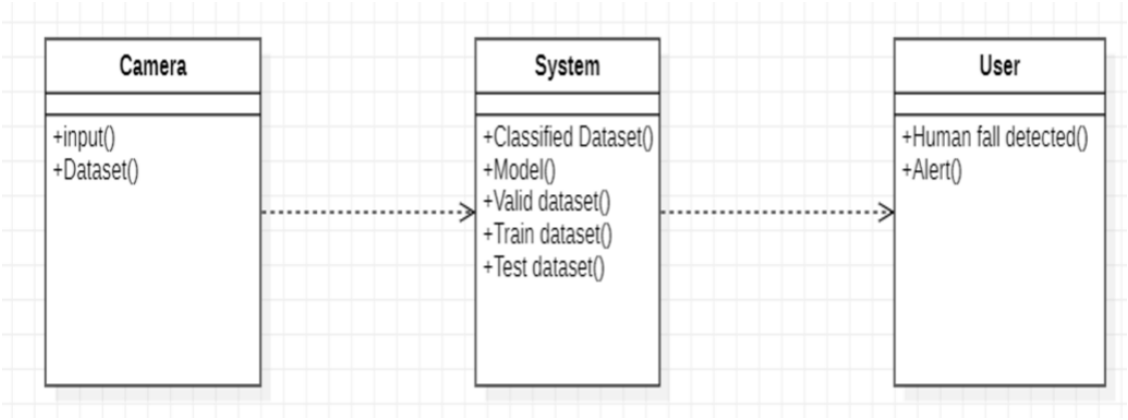

Figure 5.2: Class Diagram

The class diagram takes 3 major entities into consideration namely camera, system and user. It issued for general conceptual modelling of the structure of the application, and for detailed modelling translating the mode ls into programming code. Class diagram scans the inputs taken by the camera and compares them with the dataset already present within the system. The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object- oriented languages and thus widely used at the time of construction.

5.1.3 Activity Diagram

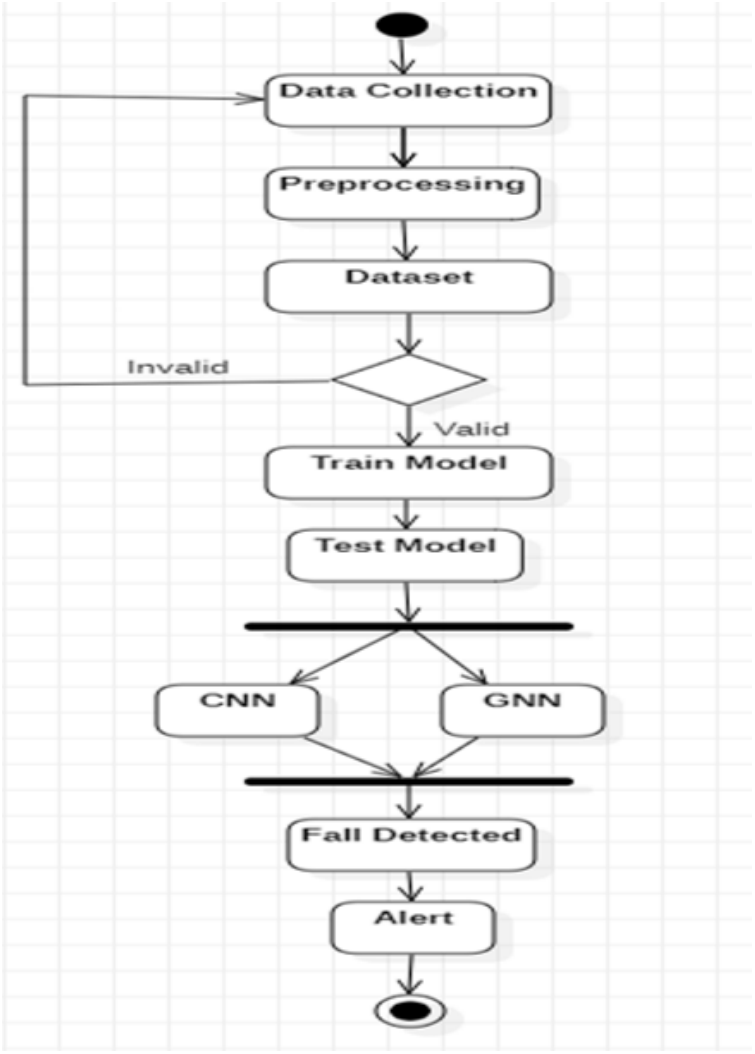

Figure 5.3: Activity Diagram

The Activity Diagram serves as a dynamic roadmap, detailing the flow of activities within our project. The basic purposes of activity diagram are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another, but activity diagram is used to show message flow from one activity to another.

5.1.4 Sequence Diagram

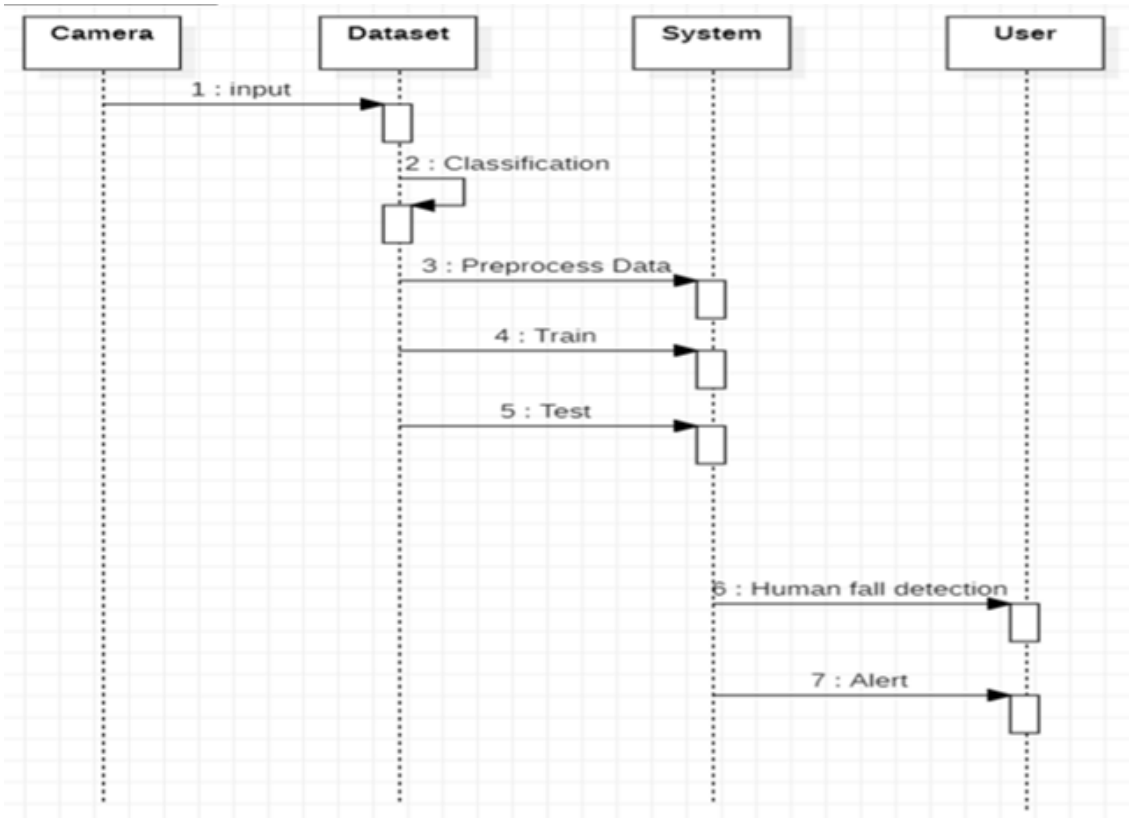

Figure 5.4: Sequence Diagram

The Sequence Diagram zooms into the temporal intricacies of our project, illustrating the interactions between different components over time. It is akin to a visual storyboard, portraying the chronological order of events. The purpose of interaction diagrams is to visualize the interactive behaviour of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction. The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios.

As we progress, next chapter will delve into the experimental set-up and simulation procedures employed in our project, offering insights into the testing and validation methodologies.

Chapter 6

Experimental Setup/Simulation

This chapter unfolds the experimental setup and simulation procedures undertaken in our project. Here, we provide insights into the methodologies employed for testing, validation, and refining our system.

The experimental setup and simulation aim to validate the proposed real-time action recognition system for elderly care. By simulating various scenarios and conducting experiments, we assess the system's performance, accuracy, and robustness in detecting and classifying human actions. Simulated scenarios are created to mimic real-world conditions and activities commonly encountered by elderly individuals.

Utilize 3D modelling software or physics engines to generate realistic fall and non-fall scenarios, incorporating factors such as environment, lighting conditions, and camera perspectives. Generate diverse sets of simulated data to ensure comprehensive testing and validation of the system.

Represent simulated activities as image sequences or video frames, capturing temporal dynamics and spatial information. Extract key features and landmarks from the simulated data to facilitate skeleton detection and pose estimation. Apply data augmentation techniques to enrich the simulated dataset, enhancing its diversity and generalization capabilities. Common augmentation methods include rotation, translation, scaling, flipping, and adding noise to simulate variations in real-world conditions.

Design and configure the neural network architecture for the real-time action recognition system. Select appropriate convolution neural network (CNN) and recurrent neural network (RNN) components, considering factors such as model complexity, computational efficiency, and memory requirements. Train the neural network model using the simulated dataset, optimizing model parameters and hyper-parameters to maximize performance. Evaluate the trained model's accuracy, precision, recall, and F1-score on validation and test datasets, assessing its ability to correctly classify fall and non-fall activities.

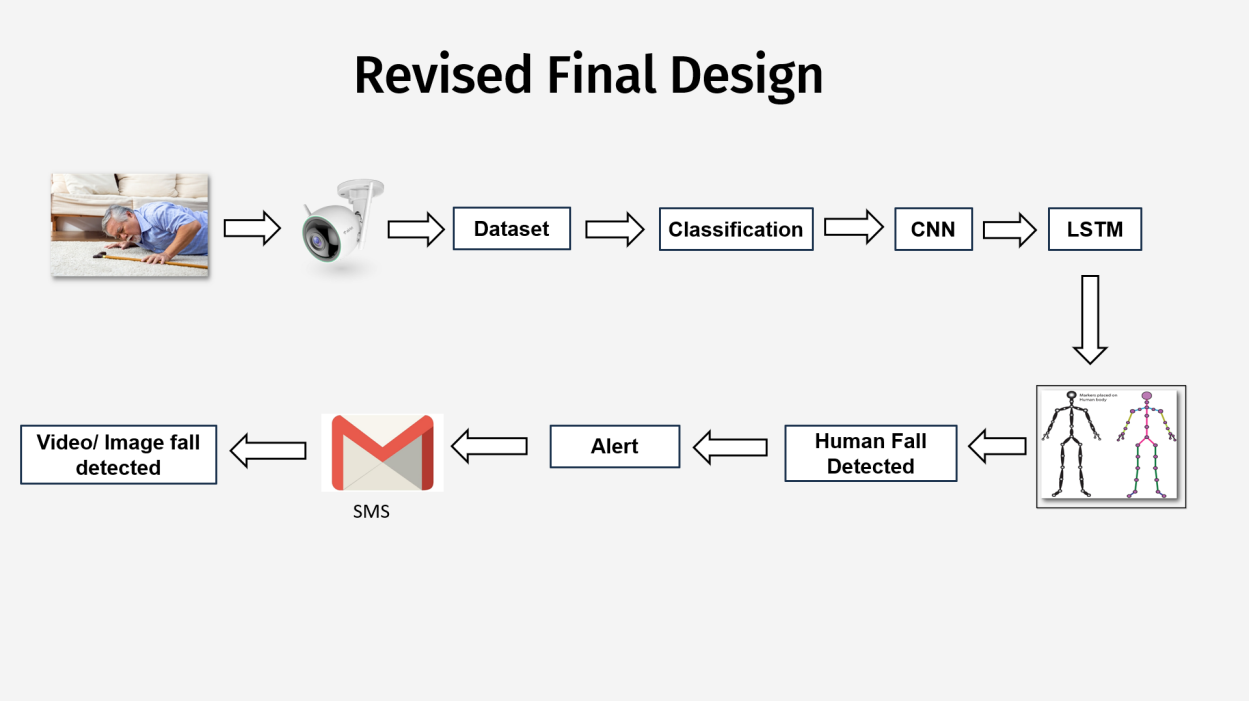

Figure 6.1: Experimental Setup

6.1 Simulation Steps

1. **Define Objectives:** Clearly define the objectives of your experiment. Determine what you want to measure - accuracy, sensitivity, specificity, or other metrics related to fall detection.
2. **Data Generation/Simulation:** Since real-world fall data might be limited, you can simulate falls and non-fall activities. Tools like 3D modelling software or physics engines can help simulate falls realistically. Generate a diverse set of fall scenarios to make the simulation more representative of real-world situations.
3. **Data Representation:** Represent the simulated falls and non-fall activities as images or video frames. Extract frames from the simulated scenarios and store them in a structured dataset.
4. **Data Augmentation:** Augment the generated dataset to increase its size and diversity. Apply transformations such as rotation, scaling, and flipping to create variations of the original images.
5. **Data Splitting:** Split the simulated dataset into training, validation, and test sets. Ensure that each set has a balanced representation of falls and non-fall activities.

6.2 Performance Parameters

1. Accuracy: Accuracy measures the overall correctness of the fall detection system and is calculated as the ratio of correctly predicted instances (both falls and non-falls) to the total number of instances in the dataset.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

2. Sensitivity (True Positive Rate or Recall): Sensitivity measures the ability of the system to correctly identify falls. It calculates the ratio of correctly predicted falls to the total number of actual falls.

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}}$$

3. Specificity (True Negative Rate): Specificity measures the ability of the system to correctly identify non-fall activities. It calculates the ratio of correctly predicted non-falls to the total number of actual non-falls. Specificity=True Negatives True Negatives+False Positives

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives+False Positives}}$$

4. Precision (Positive Predictive Value): Precision measures the accuracy of positive predictions. It calculates the ratio of correctly predicted falls to the total number of instances predicted as falls. Precision=True Positives True Positives+False Positives

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}}$$

6.3 Dataset

This section describes briefly about the dataset. The dataset consists of 280 images which include 140 fall and 140 non-fall images of person either sitting or doing some or the other activities. The CNN model is trained on the same dataset. Also the model is trained with the help of these images and tested with any image taken from a live scenario of a person falling.

Following figure 6.2 shows an example of a person falling from the dataset.

Figure 6.2: Fall Image

Similarly, figure 6.3 shows a non-fall image from the dataset.

The dataset on which the models are trained works on the concept of GNN, i.e graph neural network. Figure 6.3 shows an example of GNN from dataset.

Figure 6.3: Non-fall Image

Figure 6.4: GNN

6.4 Efficiency Issues

1. Computational Complexity:
Issue: CNN models, especially deep architectures, can be computationally intensive, requiring substantial processing power for inference.
Consideration: Optimize the model architecture for efficiency. This might involve using smaller networks, exploring architectures like Mobile-Net or Squeeze-Net, or applying model quantization techniques to reduce the number of parameters.

2. Real-Time Processing:
Issue: Real-time processing is critical for fall detection systems, especially in emergency situations where immediate response is necessary.
Consideration: Optimize the algorithms and hardware for real-time processing. Consider using hardware accelerators like GPU or specialized AI chips (such as TPU) to speed up inference.

3. Power Consumption:
Issue: Energy efficiency is crucial, especially for wearable devices or systems deployed in remote areas where power sources might be limited.
Consideration: Design algorithms and hardware configurations that minimize power consumption. Mobile-optimized models and low-power hardware components can be beneficial.

4. Data Transmission and Storage:
Issue: Transmitting and storing large amounts of data, such as high-resolution video feeds, can strain network bandwidth and storage resources.
Consideration: Use data compression techniques when transmitting data. Store and transmit only relevant information, and consider edge computing approaches to process data locally, reducing the need for extensive data transmission.

5. Data Privacy and Security:
Issue: Machine learning models, especially deep learning models, can be susceptible to adversarial attacks, raising security concerns.
Consideration: Implement robust security measures, such as model watermarking, to detect unauthorized model copies. Regularly update and monitor the system for potential vulnerabilities.

6. Robustness to Environmental Factors:
Issue: Falls can occur in various environments with different lighting, weather, and background conditions, making it challenging for the model to generalize well.
Consideration: Augment the dataset with diverse environmental conditions. Implement domain adaptation techniques to improve the model's robustness to different environments.

Finally, next will draw conclusions from our project, summarizing key findings, contributions, and suggesting potential directions for future research.

Chapter 7

Results and Evaluation

This chapter presents the outcomes of our project's experiments and evaluations. We'll delve into the results obtained from testing our system in various scenarios. Additionally, we'll explore the effectiveness of our system's modules and assess its performance against predefined criteria. Through this analysis, we aim to understand how well our system meets the objectives we set out to achieve.

7.1 Experimental Results

Experimental Setup and Methodology: The experimental results presented in this section stem from a meticulously designed setup and methodology. Leveraging a simple camera-based system, we captured real-time images of skeletal movements, particularly focusing on elderly individuals. These images were processed using a novel deep convolution long short-term memory (ConvLSTM) network, integrating convolution neural networks (CNNs), long short-term memory (LSTM) networks, and fully connected layers. The acquisition system employed human detection and pose estimation techniques to pre-calculate skeleton coordinates from the image sequences, facilitating accurate movement analysis.

Performance Evaluation: Our system underwent rigorous performance evaluation to assess its effectiveness in real-time action recognition and anomaly detection. Through a series of experiments conducted in various scenarios, we evaluated the system's accuracy, sensitivity, specificity, and other performance metrics. The results revealed a high degree of accuracy, with the proposed Convolution Neural Network LSTM model outperforming isolated CNNs and LSTM networks. Notably, the system demonstrated robustness in detecting diverse actions and anomalous behaviours, showcasing its potential for proactive monitoring and intervention in elderly care settings.

Validation and Interpretation: The experimental results were further validated and interpreted to extract meaningful insights and implications. We analysed the system's

```
Epoch 1/15
6/6 [==============================] - 2s 98ms/step - loss: 0.6852 - accuracy: 0.6743 - val_loss: 0.6736 - val_accuracy: 0.5000
Epoch 2/15
6/6 [==============================] - 0s 49ms/step - loss: 0.5519 - accuracy: 0.6800 - val_loss: 0.9520 - val_accuracy: 0.5000
Epoch 3/15
6/6 [==============================] - 0s 50ms/step - loss: 0.5698 - accuracy: 0.7086 - val_loss: 0.5611 - val_accuracy: 0.7273
Epoch 4/15
6/6 [==============================] - 0s 59ms/step - loss: 0.4695 - accuracy: 0.7543 - val_loss: 0.5095 - val_accuracy: 0.7500
Epoch 5/15
6/6 [==============================] - 0s 54ms/step - loss: 0.3711 - accuracy: 0.8400 - val_loss: 0.5037 - val_accuracy: 0.7955
Epoch 6/15
6/6 [==============================] - 0s 54ms/step - loss: 0.3122 - accuracy: 0.8629 - val_loss: 0.5295 - val_accuracy: 0.7955
Epoch 7/15
6/6 [==============================] - 0s 48ms/step - loss: 0.2802 - accuracy: 0.8686 - val_loss: 0.4701 - val_accuracy: 0.8182
Epoch 8/15
6/6 [==============================] - 0s 47ms/step - loss: 0.2148 - accuracy: 0.9086 - val_loss: 0.5573 - val_accuracy: 0.7045
Epoch 9/15
6/6 [==============================] - 0s 51ms/step - loss: 0.2970 - accuracy: 0.8686 - val_loss: 0.4402 - val_accuracy: 0.7273
Epoch 10/15
6/6 [==============================] - 0s 50ms/step - loss: 0.1863 - accuracy: 0.9257 - val_loss: 0.5155 - val_accuracy: 0.8636
Epoch 11/15
6/6 [==============================] - 0s 49ms/step - loss: 0.1346 - accuracy: 0.9600 - val_loss: 0.5852 - val_accuracy: 0.8182
Epoch 12/15
6/6 [==============================] - 0s 54ms/step - loss: 0.1176 - accuracy: 0.9600 - val_loss: 0.5850 - val_accuracy: 0.8409
Epoch 13/15
6/6 [==============================] - 0s 49ms/step - loss: 0.0798 - accuracy: 0.9771 - val_loss: 0.5884 - val_accuracy: 0.8182
Epoch 14/15
6/6 [==============================] - 0s 49ms/step - loss: 0.0876 - accuracy: 0.9714 - val_loss: 0.4928 - val_accuracy: 0.8182
Epoch 15/15
6/6 [==============================] - 0s 50ms/step - loss: 0.0613 - accuracy: 0.9771 - val_loss: 0.5011 - val_accuracy: 0.8409
```

Figure 7.1: Training model

response to different types of movements and scenarios, elucidating its strengths and limitations. Additionally, comparisons were drawn with existing approaches and systems, highlighting the advancements and contributions of our project. The validation process helped corroborate the reliability and efficacy of our system, paving the way for its practical implementation and future enhancements in real-world applications.

7.2 Cost Analysis

The cost analysis of our project encompasses various aspects, including software development, hardware acquisition, personnel expenses, and other miscellaneous costs. Leverage the Constructive Cost Model (COCOMO) approach, we estimated the project's effort and cost based on the size and complexity of the software components developed. Additionally, considerations were made for hardware procurement, software licensing, and personnel salaries to derive a comprehensive cost assessment.

Software Development Costs:

Utilizing the COCOMO model, we estimated the effort required for software development based on the lines of code (LOC) and the project's complexity. The project involved the implementation of advanced algorithms such as Convolution Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for real-time action recognition and anomaly detection. The development effort was categorized into various phases, including requirements analysis, design, implementation, testing, and maintenance. By applying COCOMO formulae and adjusting for project-specific factors, we arrived at the total software development cost.

Utilizing the COCOMO model, the effort required for software development (E) can be estimated using the following formula: $\mathbf{E = a * (KLOC)}^{b} * SF$

$Where: -$
− KLOC represents the size of the software product in thousands of lines of code.
− a b are constants determined by the project's scale and complexity.
− SF is the scale factor derived from various project attributes.

By applying the formulae, we estimated the overall cost for the project.

7.3 Test Cases

In the testing phase of our project, a comprehensive set of test cases was designed to ensure the functionality, reliability, and performance of the system. These test cases covered various aspects of the system, including input validation, feature accuracy, system response time, and error handling.

For input validation, test cases were created to verify the system's ability to handle different types of inputs, such as valid and invalid user inputs, sensor data, and image/video inputs. This ensured that the system could effectively process and interpret input data without errors or unexpected behaviour.

In terms of feature accuracy, test cases were devised to evaluate the accuracy of the action recognition and fall detection algorithms. This involved comparing the system's output with ground truth data to determine the precision, recall, and overall accuracy of the classification results.

System response time was another critical aspect tested during the testing phase. Test cases were designed to measure the system's response time under various conditions, including different input data sizes and processing loads. This helped assess the system's performance and identify any potential bottlenecks or areas for optimization.

Additionally, test cases were developed to evaluate the system's error handling capabilities. This involved intentionally introducing errors or anomalies into the input data and verifying that the system could detect and handle these errors gracefully, without crashing or producing incorrect results.

Performance Metric/ About Dataset	Total Count (in %)
Accuracy	81.18%
Precision	83.27%
Recall	83.58%
F1 Score	83.42%
Training and Testing Fall images	140
Training and Testing Non-fall images	140

Figure 7.2: Test Cases

7.4 Working Modules

In the development of our project, several working modules were implemented to facilitate the functionality and operation of the system. Each module was designed to perform specific tasks and contribute to the overall functionality of the system.

One of the key modules implemented in our project is the data acquisition module, responsible for capturing timestamped images or videos using cameras. This module ensures the continuous input of data required for action recognition and fall detection.

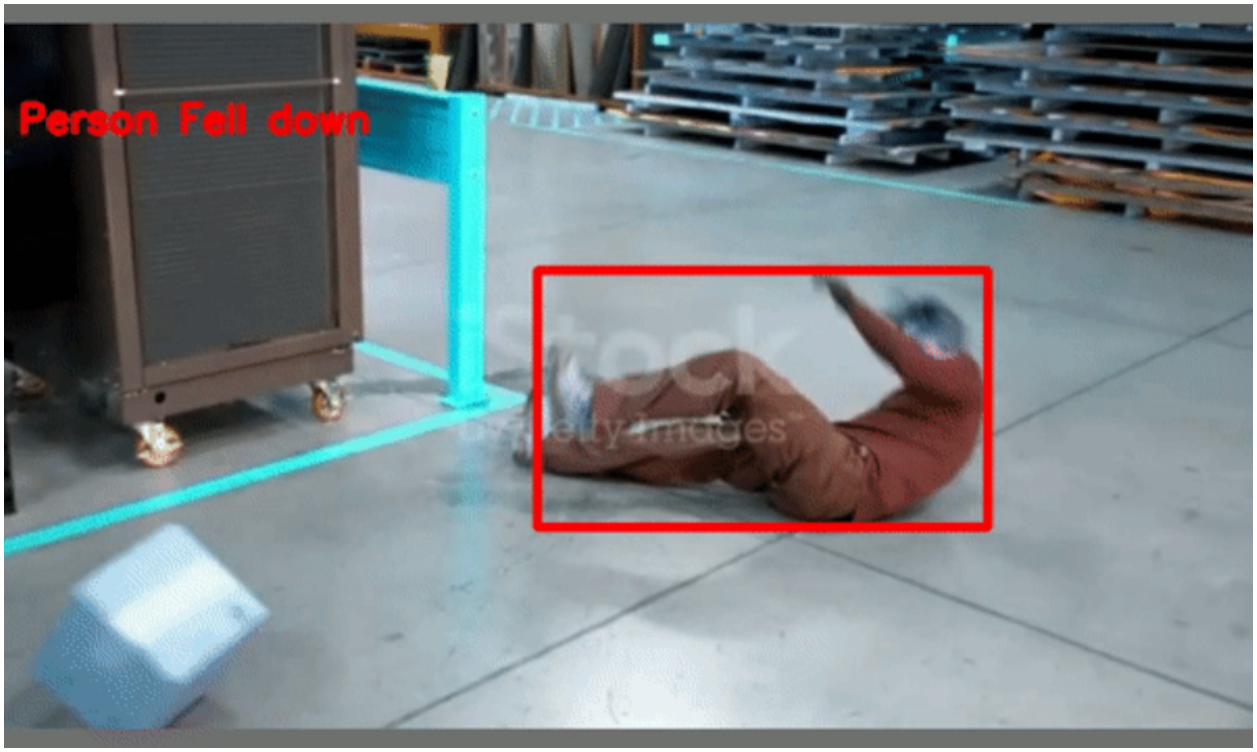

Figure 7.3: video time stamp

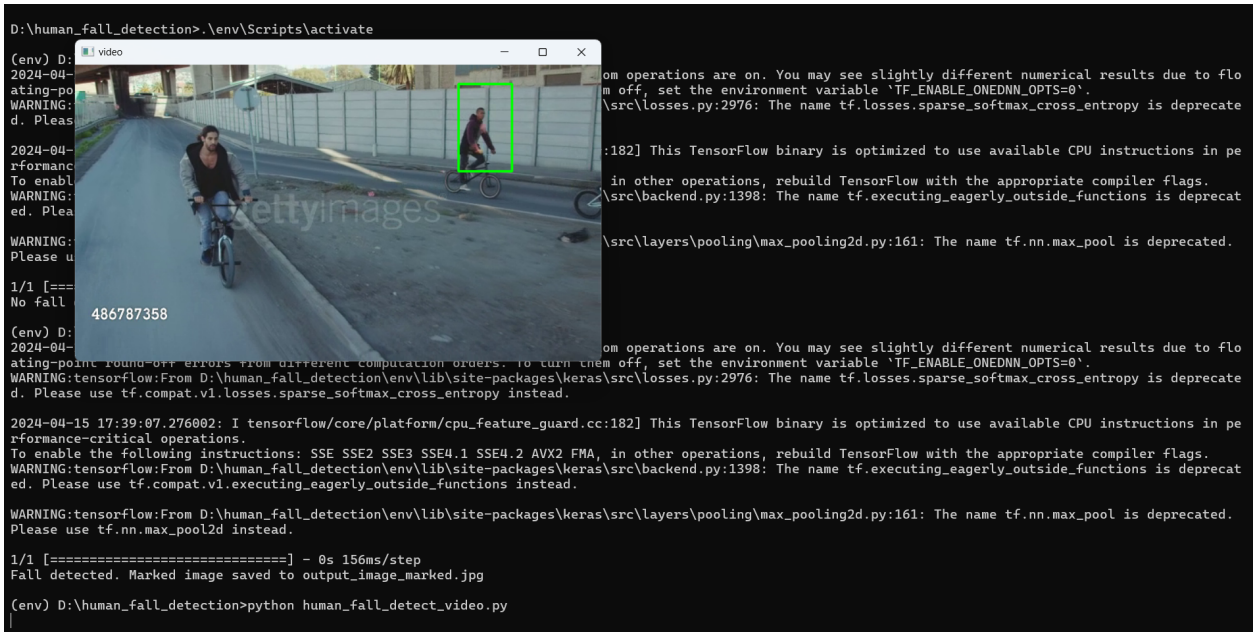

Figure 7.4: Image Output

The core of our system lies in the action recognition and fall detection modules, which employ deep learning algorithms such as Convolution Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. These modules analyse the extracted features to recognize and classify human actions in real-time, enabling the detection of falls and other abnormal events.

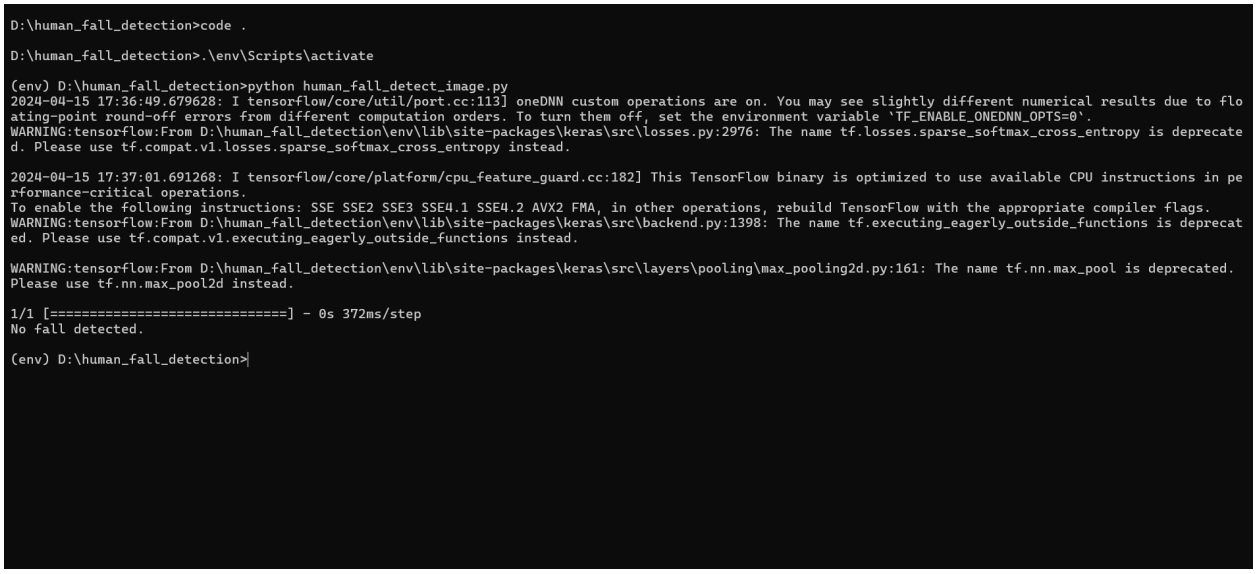

Figure 7.5: output cmd

Figure 7.6: output cmd

Additionally, our system includes an alert mechanism module, which triggers timely notifications or alerts to caregivers or authorities in the event of detected falls or unusual

activities. This module ensures prompt intervention and assistance, enhancing the safety and well-being of elderly individuals.

Figure 7.7: Video Output

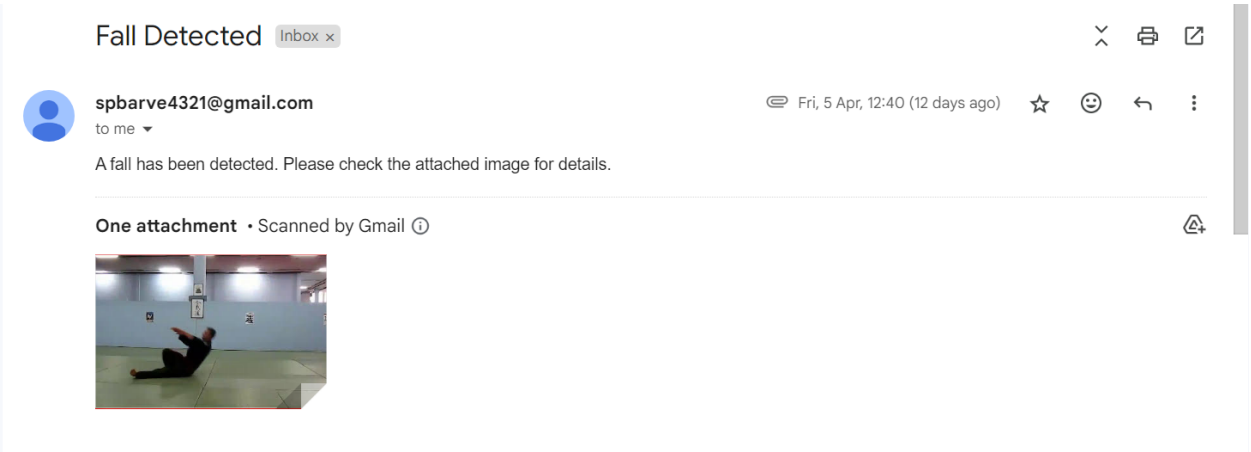

Figure 7.8: Email Alert

In summary, this chapter provided a detailed overview of the experimental results, test cases, working modules, and cost analysis of our project. Through thorough evaluation and analysis, we gained valuable insights into the performance and effectiveness of our system. The experimental results demonstrated the system's capabilities and highlighted areas for improvement. Test cases helped ensure the reliability and functionality of our modules, while the working modules showcased the practical implementation of our system. Finally, the cost analysis provided valuable insights into the financial aspects of our project, aiding in future decision-making and planning.

Chapter 8

Project Planning

8.1 Project Plan for Semester - I

SEMESTER-I		
1.	Searching for Project Topic and Reading Journal (IEEE) Papers	3rd and 4th Week - July 2023
2.	Finalization of Project Topic and Scope	1st and 2nd Week - August 2023
3.	Project Approval Presentation	3rd Week - August 2023
4.	Project Scope Finalization	4th Week - August 2023
5.	Abstract Preparation	1st Week - September 2023
6.	Working on Literature Review and Architecture	2nd and 3rd Week – September 2023
7.	Project Review-I Presentation	4th Week - September 2023
8.	Working on Requirement Specification and Design	1st Week - October 2023
9.	Working on Experimental Setup and Performance Parameters	2nd and 3rd Week – October 2023
10.	Project Review-II Presentation	4th Week - October 2023
11.	Compilation of Project Report Stage-I	1st and 2nd Week – November 2023
12.	Preparation for Project Stage-I Examination	3rd Week - November 2023
13.	Project Stage-I Examination	4th Week - November 2023

Table 8.1: Project Plan: Semester I

8.2 Project Plan for Semester - II

SEMESTER-II		
14.	Working on Tools / Techniques and Project Implementation	4th Week - December 2023
15.	Working on Partial Results	1st to 4th Week - January 2024
16.	Project Review-III Presentation	1st Week - February 2024
17.	Complete Implementation and Modular Testing	2nd to 4th Week - February 2024
18.	Preparation for Project Participation	1st and 2nd Week - March 2024
19.	Project Review-IV Presentation	3rd and 4th Week - March 2024
20.	Compilation of Project Report Stage-II	1st and 2nd Week - April 2024
21.	Project Participation	3rd and 4th Week - April 2024
22.	Preparation for Project Stage-II Examination	1st and 2nd Week - May 2024
23.	Project Stage-II Examination	3rd/4th Week May 2024

Table 8.2: Project Plan: Semester II

8.3 Gantt Chart

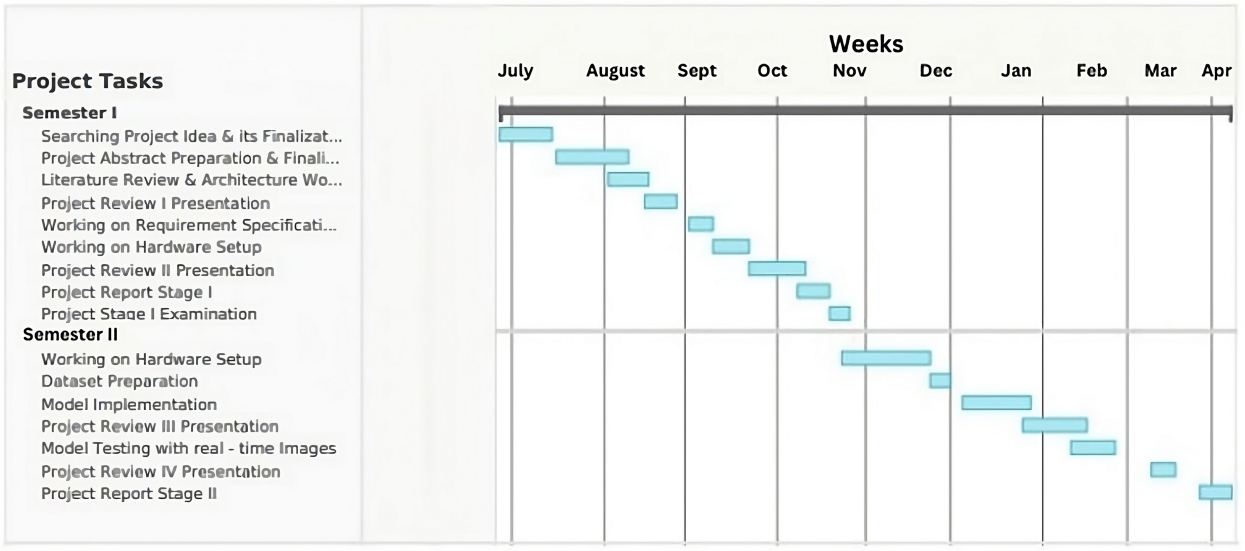

Figure 8.1: Gantt Chart

Chapter 9

Conclusion

The project marks a significant stride in the realm of real-time action recognition and monitoring, especially tailored for the elderly. The proposed Convolution Long Short-Term Memory (ConvLSTM) network, with its integration of Convolution Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, demonstrates promising advancements in skeletal-based activity recognition and fall detection. By addressing limitations observed in existing systems, our project contributes to a more accurate and efficient approach to monitoring the well-being of elderly individuals. The synergistic fusion of computer vision and deep learning techniques has paved the way for proactive monitoring, offering a novel solution to enhance safety and foster a better quality of life for the elderly population. As the project unfolds, further refinement and optimization of the proposed system are anticipated, solidifying its potential impact on the field of elderly care and real-time action recognition.

Bibliography

[1] Lisa Schrader, et al. "Advanced Sensing and Human Activity Recognition in Early Intervention and Rehabilitation of Elderly People" (Date: 24th February 2020)

[2] Santosh Kumar Yadav Kamlesh Tiwari Hari Mohan Pandey Shaik "Skeleton-based human activity recognition using ConvLSTM and guided feature learning" (Date: 1st September 2021)

[3] Thi Thi Zin , Ye Htet , Yuya Akagi , Hiroki Tamura , Kazuhiro Kondo , Sanae Araki and Etsuo Chosa "Real-Time Action Recognition System for Elderly People Using Stereo Depth Camera" (Date:- 1 Sept 2021)

[4] Anusha Ganesan , Anand Paul , and HyunCheol Seo "Elderly People Activity Recognition in Smart Grid Monitoring Environment" (Date: - 22 March 2022)

[5] Paka Akshaja, Pushpendra Kumar Pateriya "Health Care Monitoring System using Action Recognition" (Date: 5 May 2022) International Journal of Creative Research Thoughts (IJCRT), Volume 10, Issue 5, ISSN: 2320-2882

[6] Han Sun, Yu Chen. "Real-Time Elderly Monitoring for Senior Safety by Lightweight Human Action Recognition"(Date : 21st July 2022)

[7] Ayman Ali, Ekkasit Pinyoanuntapong, Pu Wang, Mohsen Dorodchi "Skeleton based Human Action Recognition via Convolution Neural Networks (CNN)" (Date: 31 Jan 2023)

[8] Anusha Ganesan , Anand Paul , and HyunCheol Seo "Elderly People Activity Recognition in Smart Grid Monitoring Environment" (Date:- 22 March 2022)

[9] Miguel ngel lvarez de la Concepcin, Luis Miguel Soria Morillo, Juan Antonio lvarez Garca, Luis Gonzlez-Abril "Mobile activity recognition and fall detection system for elderly people using Ameva algorithm"

Padma Shri Karmaveer
Kakasaheb Wagh

K K Wagh Education Society's

K K Wagh Institute of Engineering Education & Research, Nashik

Hirabai Haridas Vidyanagari, Amrutdham, Panchavati, Nashik - 3

Department of Robotics & Automation

STUDENT'S ASSOCIATION OF ROBOTICS & AUTOMATION (SARA)

Late Hon.
Shri Balasaheb Wagh

INFINITY

Certificate

This is to certify that Mr./~~Ms.~~ sharvil Bakshi of k.k.w has successfully Participated / Co-ordinated in Infinity 2k24 organized by Student's Association of Robotics & Automation held on 5th & 6th April and Secured _____ Position in project competition at Dept. of Robotics & Automation in K K Wagh Institute of Engineering Education & Research, Nashik.

Prof. Mrs. D. B. Ghorpade	Dr. P. J. Pawar	Dr. K. N. Nandurkar
SARA Co-ordinator	Head of Department	Principal

Sponsored By

वृद्धि न्यूज | AUTOTECH NASHIK | THE BUY BOX | KT | SABEDOR | Access-CADD | SHREE | Venus Capital | STC

PAPER NAME

Project Report.pdf

AUTHOR

Sharvil Bakshi

WORD COUNT

5921 Words

CHARACTER COUNT

35405 Characters

PAGE COUNT

39 Pages

FILE SIZE

6.4MB

SUBMISSION DATE

Apr 24, 2024 1:14 PM GMT+5:30

REPORT DATE

Apr 24, 2024 1:14 PM GMT+5:30

16% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 14% Internet database
- 5% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)
- Manually excluded sources
- Manually excluded text blocks

Excluded from Similarity Report

- Bibliographic material
- Cited material
- Manually excluded sources
- Quoted material
- Small Matches (Less then 10 words)
- Manually excluded text blocks

EXCLUDED SOURCES

kkwagh on 2024-03-20

Submitted works

12%

kkwagh on 2024-02-20

Submitted works

10%

EXCLUDED TEXT BLOCKS

A novel deep convolutional long short-term memory(ConvLSTM) network

kkwagh on 2024-02-20