# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**“JNANA SANGAMA”, BELAGAVI, KARNATAKA – 590 018**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**A Project Report on,**

# “HUMAN ACTION RECOGNIZATION”

**Submitted to partial fulfillment of the requirement for the degree of Computer Science and Engineering for the Academic Year 2024-25**

**Submitted by,**

| | |
|---|---|
| HARSHITHA N | 1SK21CS018 |
| MAHESH R | 1SK21CS024 |
| PRAJWAL BABU B S | 1SK21CS033 |
| RAKESH G | 1SK21CS038 |

**Under the Guidance of,**

**Dr. Anitha A C**<br>
**Assistant Professor**<br>
**Department of CSE**

**GOVERNMENT S K S J TECHNOLOGICAL INSTITUTE**

**K R Circle, Bengaluru – 560001**

**(Affiliated to Visvesvaraya Technological University, Belagavi)**

# GOVT. S K S J TECHNOLOGICAL INSTITUTE

## K R CIRCLE, BENGALURU-560001

**(Affiliated to Visvesvaraya Technological University, Belagavi)**

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

# CERTIFICATE

This is to certify that the project work entitled **"HUMAN ACTION RECOGNIZATION"** is carried out by **HARSHITHA N (1SK21CS018), MAHESH R (1SK21CS024), PRAJWAL BABU B S (1SK21CS033), RAKESH G (1SK21CS038)** bonafide students of **Government Sri Krishnarajendra Silver Jubilee Technological Institute**, K R Circle, Bengaluru in partial fulfillment of **Bachelor's degree in Computer-Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2024-2025.

| | | |
|---|---|---|
| **Signature of Guide** | **Signature of HoD** | **Signature of Principal** |
| **Dr. Anitha A C** | **Dr. Nagaraj B. Patil** | **Dr. Venkatesh D Bemmathi** |
| Assistant Professor | Head of CSE | Principal |
| Department of CSE | Department of CSE | G S K S J T I |

| | |
|---|---|
| **Name of the Examiners** | **Signature with Date** |
| **1.** ______________________ | **1.** ______________________ |
| **2.** ______________________ | **2.** ______________________ |

# ACKNOWLEDGEMENT

We are grateful to our institution**, Govt. Sri Krishna Rajendra Silver Jubilee Technological Institute (G.S.K.S.J.T.I.),** for providing the facilities and inspiration that made this project a success.

Our sincere thanks go to **Dr. Venkatesh D Bemmathi** Principal of G.S.K.S.J.T.I., for his valuable guidance, suggestions, and consistent encouragement throughout our project. His timely assistance was crucial to its completion.

We extend our heartfelt thanks to the entire faculty of the Computer Science and Engineering Department. Special thanks to **Dr. Nagaraj B. Patil,** Assistant Professor and HoD of the CSE Department, for his confidence in us, support, and invaluable guidance.

We appreciate our academic project guide, **Dr. Anitha A C,** Assistant Professor of the CSE Department, once again for his contributions to our understanding and application of the project's core concepts, as well as his unwavering support and encouragement.

Our gratitude also goes to the entire Computer Science Department for their collective help, guidance, encouragement, and cooperation during all stages of our project. Their support was instrumental to our success.

Last but not least, we thank our parents and friends for their moral support, and everyone who provided us with support, advice, and encouragement essential to the completion of this project.

| | |
|---|---|
| **HARSHITHA N** | **1SK21CS018** |
| **MAHESH R** | **1SK21CS024** |
| **PRAJWAL BABU B S** | **1SK21CS033** |
| **RAKESH G** | **1SK21CS038** |

# ABSTRACT

Human action recognition is one of the fundamental challenges in robotics systems. In this paper, we propose one lightweight action recognition architecture based on deep neural networks just using RGB data. The proposed architecture consists of convolution neural network (CNN), long short-term memory (LSTM) units, and temporal-wise attention model. First, the CNN is used to extract spatial features to distinguish objects from the background with both local and semantic characteristics. Second, two kinds of LSTM networks are performed on the spatial feature maps of different CNN layers (pooling layer and fully-connected layer) to extract temporal motion features. Then, one temporal-wise attention model is designed after the LSTM to learn which parts in which frames are more important. Lastly, a joint optimization module is designed to explore intrinsic relations between two kinds of LSTM features. Experimental results demonstrate the efficiency of the proposed method.

# TABLE OF CONTENTS

**CHAPTERS** **PAGE NO.**

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

## INTRODUCTION

Human Action Recognition (HAR) refers to the process of analysing videos or images to detect and classify various human activities or gestures. HAR systems have gained significant importance in multiple fields, including healthcare, sports, security, surveillance, robotics, and entertainment. For example, surveillance systems can detect unusual actions like fighting or theft, while healthcare applications track patients' rehabilitation progress.

With advances in deep learning, computer vision, and data availability, HAR systems have transitioned from simple hand-crafted feature-based methods to sophisticated models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The challenge lies in developing an accurate system that can effectively handle variations in lighting, bacgrounds, camera angles, and human movements.

Human Action Recognition is a promising area of computer vision with applications in multiple domains. This project will develop an HAR system that recognizes human activities with high accuracy under real-world conditions. By focusing on the objectives outlined above, the system can overcome the challenges posed by variability in actions, background noise, and real-time performance.

### 1.1 Background and Motivation

Human Action Recognition (HAR) represents an innovative branch of computer vision focused on identifying and classifying human activities or gestures from videos or images. With its potential to impact a wide array of industries, HAR has found applications in healthcare, where it tracks patient rehabilitation; surveillance, detecting unusual behaviours such as theft or fighting; and entertainment, where it enhances user interaction. Advancements in deep learning, computer vision, and data availability have revolutionized However, these systems face significant challenges, including variations in lighting, backgrounds, human movements, and real-time performance needs. This project aims to address these challenges by developing a robust and scalable HAR system that delivers high accuracy across diverse real-world conditions.

## 1.2 Problem Statement

To develop a computer vision system for Human Action Recognition (HAR) that can identify and classify various human actions from video data in real time. This system must address challenges such as diverse human poses, varying lighting conditions, background clutter, and occlusions.

## 1.3 Objectives

The key objectives of this project include:

**1. Developing an Accurate HAR System:** Use computer vision and machine learning techniques to classify human actions.

**2. Handling Diverse Scenarios:** Ensure that the model performs well under different environmental conditions (e.g., low lighting, occlusions).

**3. Real-time Recognition:** Implement the solution to recognize actions in real-time or near-real-time, which is essential for applications like surveillance.

**4. Reducing False Positives:** Minimize the number of misclassifications and false detections to improve reliability.

**5. Ease of Deployment:** Ensure the system is compatible with general-purpose hardware (e.g., CPUs, GPUs) for easy implementation.

**6. Building a Scalable System:** Make the model scalable to recognize a wide variety of actions.

## 1.4 Scope and Limitations

### Scope

In the proposed approach, we conceptually proposed an Innovative HAR model for wearable sensor based human activity recognition applications by concatenating convolution kernels of different scales and splicing with max-pooling layers. From the dataset features are extracted using Convolution Neural Network (CNN) and LSTM (Long Short-Term Memory) to deduct the Human Activity. Spatial-Temporal features are extracted using CNN-LSTM model.

### Limitations

Human Action Recognition (HAR) systems, despite their significant advancements, have several limitations that pose challenges in real-world applications. These include:

- **Data Dependency:** HAR systems require large and diverse datasets for training to generalize well to new scenarios. However, such datasets may not always cover all possible scenarios or action variations.
- **Scalability and Flexibility**: Building systems that can scale effectively to recognize an extensive range of actions or adapt to new actions without retraining remains an ongoing challenge.
- **Diverse Human Actions**: The wide variety of human actions, gestures, and poses adds complexity to classification tasks. Subtle variations between actions can lead to misclassification or decreased system performance.
- **Hardware Requirements**: While the project aims for compatibility with general-purpose hardware, computationally intensive tasks such as training deep learning models still benefit significantly from high-performance GPUs, which may not always be available.

# CHAPTER 2

## LITERATURE REVIEW

A literature survey or a literature review in a project report shows the various analyses and research made in the field of interest and the results already published, taking into account the various parameters of the project and the extent of the project. Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system & guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

A literature survey is a text of a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews use secondary sources, and do not report new or original experimental work. Most often associated with academic-oriented literature, such as a thesis, dissertation or a peer-reviewed journal article, a literature review usually precedes the methodology and results sectional though this is not always the case. Literature reviews are also common in are search proposal or prospectus (the document that is approved before a student formally begins a dissertation or thesis). Its main goals are to situate the current study within the body of literature and to provide context for the particular reader. Literature reviews are a basis for researching nearly every field.

A literature survey includes the following:

- Existing theories about the topic which are accepted universally.
- Books written on the topic, both generic and specific.
- Research done in the field usually in the order of oldest to latest.
- Challenges being faced and on-going work, if available.

Literature survey describes about the existing work on the given project. It deals with the problem associated with the existing system and also gives user a clear knowledge on how to deal with the existing problems and how to provide solution to the existing problems.

Objectives of Literature Survey:

• Learning the definitions of the concepts.

• Access to latest approaches, methods and theories.

• Discovering research topics based on the existing research

• Concentrate on your own field of expertise- Even if another field uses the same words, they usually mean completely.

• It improves the quality of the literature survey to exclude sidetracks- Remember to explicate what is excluded.

## 2.1 Overview of the Existing Research

1. **Human Activity Recognition Based on Deep Learning Method**

   **Xiaoran Shi, Yaxin Li, Feng Zhou, Lei Liu**

   With the increasing demand of security defense, anti-terrorism investigation and disaster rescue, human activity classification and recognition have become a hot research topic. When a human is illuminated by electromagnetic waves, a Doppler signal is generated from his or her moving parts. Indeed, bodily movements are what make humans' micro-Doppler signatures unique, offering a chance to classify human activities. Classification needs a lot of samples for training, however, in the real application, there is a certain gap between the simulated data and the real data, and the measured data is often difficult to obtain. Due to the nonstationary characteristic for human radar echoes, the spectrograms for the human activities show different micro-Doppler signatures.

   Therefore, we proposed a method of human activity classification based on spectrograms using deep learning techniques, including deep convolutional generative adversarial network for expanding and enriching training set and a transfer-learned deep convolutional network (DCNN) for feature extraction and classification, which is based on a DCNN pre-trained by a large-scale RGB image data set-that is, ImageNet.

2. **Human Activity Recognition Based On Convolutional Neural Network**

   **WenchaoXu, Yuxin Pang, YanqinYang ,Yanbo Liu**

   Smartphones are ubiquitous and becoming increasingly sophisticated, with ever-growing sensing powers. Recent years, more and more applications of activity recognition based on sensors are developed for routine behavior monitoring and helping the users form a healthy habit. In this field, finding an efficient method of recognizing the physical activities (e.g., sitting, walking, jogging, etc) becomes the pivotal, core and urgent issue. In this study, we construct a Convolutional Neural Network (CNN) to identify human activities using the data collected from the three-axis accelerometer integrated in users' smartphones. The daily human activities that are chosen to be recognized include walking, jogging, sitting, standing, upstairs and downstairs.

The three-dimensional (3D) raw accelerometer data is directly used as the input for training the CNN without any complex pretreatment. The performance of our CNN-based method for multi human activity recognition showed 91.97%accuracy, which outperformed the Support Vector Machine (SVM) approach of 82.27% trained and tested with six kinds of features extracted from the 3D raw accelerometer data. Therefore, our proposed approach achieved high recognition accuracy with low computational cost.

### 3. Human Activity Recognition From Accelerometer Data Using Convolutional Neural Network

**Song-Mi Lee Sang Min Yoon Heeryon Cho**

We propose a one-dimensional (1D) Convolutional Neural Network (CNN)-based method for recognizing human activity using triaxial accelerometer data collected from users' smartphones. The three human activity data, walking, running, and staying still, are gathered using smartphone accelerometer sensor. The x, y, and z acceleration data are transformed into a vector magnitude data and used as the input for learning the 1DCNN. The ternary activity recognition performance of our 1DCNN-based method which showed 92.71% accuracy outperformed the baseline random forest approach of 89.10%.

### 4. Multi-view human activity recognition using motion frequency

**NeslihanK¨ose, MohammadrezaBabaee, Gerhard Rigoll**

The problem of human activity recognition can be approached using spatio-temporal variations in successive video frames. In this paper, a new human activity recognition technique is proposed using multi-view videos. Initially, a naive backgrounds subtraction using frame differencing between adjacent frames of a video is performed.

Then, the motion information of each pixel is recorded in binary indicating existence/non-existence of motion in the frame. A pixel wise sum over all the difference images in a view gives the frequency of motion in each pixel throughout the clip. The classification performances are evaluated using these motion frequency features. Our analysis shows that increasing number of views used for feature extraction improves the performance as different views of an activity provide complementary information. Experiments on the i3DPost and the INRIA Xmas Motion Acquisition Sequences (IXMAS) multi-view human action datasets provide significant classification accuracies.

5. **Automated Daily Human Activity Recognition for Video Surveillance Using Neural Network**

**Mohanad Babiker, Othman O. Khalifa, KyawKyaw Htike, Aisha Hassan, Muhamed Zaharadeen**

surveillance video systems are gaining increasing attention in the field of computer vision due to its demands of users for the seek of security. It is promising to observe the human movement and predict such kind of sense of movements. The need arises to develop a surveillance system that capable to overcome the shortcoming of depending on the human resource to stay monitoring, observing the normal and suspect event all the time without any absent mind and to facilitate the control of huge surveillance system network. In this paper, an intelligent human activity system recognition is developed. Series of digital image processing techniques were used in each stage of the proposed system, such as background subtraction, binarization, and morphological operation. A robust neural network was built based on the human activities features database, which was extracted from the frame sequences. Multi-layer feed forward perceptron network used to classify the activities model in the dataset. The classification results show a high performance in all of the stages of training, testing and validation. A robust neural network was built based on the human activities features database, which was extracted from the frame sequences. Finally, these results lead to achieving a promising performance in the activity recognition rate. In this paper, an intelligent human activity system recognition is developed.

6. **A Review on Computer Vision-Based Methods for Human Action Recognition**

**Mahmoud Al-Faris, John P Chiverton, David Ndzi, Ahmed Isam Ahmed.**

This is an article about computer vision-based methods for human action recognition.

It discusses the challenges and applications of this technology. Traditional methods rely on hand-crafted features. Deep learning methods are becoming increasingly common. Some important datasets are included. Deep learning techniques have revolutionized HAR by leveraging large-scale datasets and powerful computational models. The authors classify these approaches into discriminative, generative, and multi-modal methods. Discriminative models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), excel in learning action-specific patterns from labeled data. Generative models, including autoencoders and generative adversarial networks (GANs), aim to understand and simulate data distributions, enabling better generalization. Multi-modal methods integrate various data types—such as RGB images, depth maps, and optical flow—to enhance the robustness and accuracy of recognition systems.

The review also highlights key datasets used in HAR research, such as UCF101, HMDB51, and Kinetics. These datasets provide diverse challenges and benchmarks for evaluating the performance of different methods. The authors emphasize the importance of balancing dataset size, diversity, and annotation quality to improve model generalization.

### 7. Human Activity Recognition from Wearable Sensor Data Using Self  Attention

**Saif Mahmud**

This is an article about a Tensorflow implementation of a self-attention based Human Activity Recognition model. The paper discusses the model architecture, dataset download, pretrained models, training and evaluation, and citation. The model is implemented using Tensorflow 2.x. The authors provide instructions on how to download the dataset, train the model, and evaluate its performance. The study addresses the challenges in Human Activity Recognition (HAR) using data from body-worn sensors, particularly focusing on capturing spatial and temporal dependencies in time-series signals. Traditional models, such as recurrent or convolutional neural networks, often struggle to effectively capture the spatio-temporal context of sensor data sequences.

To overcome these limitations, the authors propose a self-attention-based neural network model that eliminates the need for recurrent architectures. This model employs various attention mechanisms to generate higher-dimensional feature representations, enhancing classification performance.

The researchers conducted extensive experiments on four publicly available HAR datasets: PAMAP2, Opportunity, Skoda, and USC-HAD. Their model demonstrated significant performance improvements over recent state-of-the-art models in both benchmark test subjects and leave-one-subject-out evaluations. Additionally, the sensor attention maps produced by the model effectively captured the importance of sensor modality and placement in predicting different activity classes. The study addresses the challenges in Human Activity Recognition (HAR) using data from body-worn sensors, particularly focusing on capturing spatial and temporal dependencies in time-series signals. Traditional models, such as recurrent or convolutional neural networks, often struggle to effectively capture the spatio-temporal context of sensor data sequences.

8. **Real Time Human Action Recognition Using Raw Depth Video-Based Recurrent Neural Network**

**Jiawei Li, Yifan Zhang, Xiong Li, and Xiaobai Li**

This is an article about real-time human action recognition using raw depth video. It discusses two approaches based on convolutional long short-term memory units (Conv LSTMs). Both methods achieve competitive recognition accuracies. The first uses a video-length adaptive input data generator. The second explores the stateful ability of recurrent neural networks. This allows the model to accumulate information from previous frames. Both approaches use only depth information, preserving privacy.

By focusing solely on depth information, the proposed methods preserve individual privacy, as identities are not discernible. The models were trained and tested using the large-scale NTU RGB+D dataset. Experimental results indicate that both models achieve competitive recognition accuracies with lower computational costs compared to state-of-the-art methods. Notably, the stateful model demonstrated improved accuracy, achieving 80.43% in cross-subject evaluations and 79.91% in cross-view evaluations, with an average processing time of 0.89 seconds per video.

9. **Human Action Recognition using CNN and LSTM-RNN with Attention Model**
**Muhammad Umar Khan, Muhammad Awais, Muhammad Sharif, and Fahad Shahbaz Khan.**

This is an article about human action recognition using CNN and LSTM-RNN. It discusses using video to recognize human actions. It presents a hybrid approach that combines convolutional neural networks (CNNs) and long short-term memory recurrent neural networks (LSTM-RNNs) enhanced with attention mechanisms for human action recognition. The authors propose a system that uses a convolutional neural network (CNN) to extract features from video frames and a long short-term memory recurrent neural network (LSTM-RNN) to capture the temporal relationships between the features. The system also includes an attention model that helps to focus on the most informative parts of the video. The authors report that their system achieves state-of-the-art results on a benchmark dataset.

The proposed system was evaluated on three benchmark datasets: UCF11, UCF Sports, and J-HMDB. The model achieved recognition rates of 98.3%, 99.1%, and 80.2% on these datasets, respectively, demonstrating a 1%–3% improvement compared to baseline state-of-the-art methods. These results highlight the effectiveness of integrating dilated CNN features with an attention-based BiLSTM network for human action recognition tasks.

This research contributes to the field by presenting a robust framework that addresses the limitations of traditional HAR methods, offering improved accuracy and efficiency in recognizing human actions from video sequences.

### 10. A Novel CNN-based Bi-LSTM parallel model with attention mechanism for human activity recognition with noisy data

**Xiaochun Yin**

This paper proposes a novel CNN-based Bi-LSTM parallel model with attention mechanism for human activity recognition with noisy data. The model uses a 1-D CNN-based bi-directional LSTM parallel network for dimensionality reduction and elimination of noisy data; it uses a parallel structure for time complexity reduction; it also uses attention mechanism for high accuracy by redistribution of the weights of key representations. The experimental results show that the proposed model outperforms the existing models in terms of both classification accuracy and computational time complexity.

The model's performance was evaluated on public UCI and WISDM HAR datasets. The ConvBLSTM-PMwA model achieved a classification accuracy of 96.71% and demonstrated computational efficiency, processing data at least 1.1 times faster than existing CNN and Recurrent Neural Network (RNN) models, even when handling noisy data.

This research contributes to the field of HAR by introducing a robust model that effectively balances accuracy and computational efficiency, particularly in scenarios involving noisy or incomplete data. The integration of CNN, Bi-LSTM, and attention mechanisms offers a promising approach for real-time human activity recognition applications.

**Summary of the Literature Survey**

- **Focus on Modern Techniques:**

  Algorithms like Slow Fast, Transformers, GCNs, Two-Stream CNNs, 3D CNNs with LSTMs, and Multimodal Fusion.

- **Comprehensive Evaluation:**

  Studies compare accuracy, robustness, and limitations across various datasets like NTU RGB+D, SBU Kinect, UCLA, and Weizmann.

- **Highlight of Results:**

  Accuracy up to 97.6% (Pose Net) and robust performance improvements in complex scenarios.

- **Real-Time Application:**

  Focus on efficient frame-by-frame classification for real-time recognition with simpler hardware setups.

This outlines cutting-edge trends, strengths, and existing challenges in HAR research.

## 2.2 Discussion of relevant theories and concepts

Human Action Recognition (HAR) is a dynamic field within computer vision that focuses on identifying and classifying human activities from video or image data. At its core, HAR leverages the temporal and spatial information embedded in video sequences to interpret actions. Videos are treated as a sequence of frames, where each frame contributes spatial features, while the transitions between frames capture temporal dynamics. This dual nature makes HAR a challenging yet exciting domain that often utilizes advanced methods like Convolutional Neural Networks (CNNs) for spatial analysis and techniques such as Long Short-Term Memory (LSTM) networks or 3D Convolutional Networks (C3D) for modeling temporal dependencies.

The process begins with organizing datasets, typically by labeling video sequences according to the activities they depict. This structured approach enables the use of supervised learning models, where preprocessing steps such as resizing, normalization, and augmentation play a crucial role in enhancing the model's robustness and adaptability. Metrics like accuracy, confusion matrices, and classification reports are commonly used to evaluate model performance, ensuring that the trained system can generalize effectively to unseen scenarios.

Traditional HAR systems relied on hand-crafted features, such as histograms of oriented gradients (HOG) or optical flow, to extract motion and texture patterns. However, deep learning revolutionized the field by automating feature extraction using convolutional neural networks (CNNs), which are adept at capturing spatial features directly from raw data. To handle sequential data, recurrent neural networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) networks, have been widely adopted. These models address the challenge of long-term dependencies in action sequences. Recent advancements include attention mechanisms, which prioritize the most relevant parts of input sequences, and transformer-based models, which excel in capturing global dependencies.

In vision-based HAR, spatio-temporal analysis is essential for modeling actions comprehensively. Techniques like 3D CNNs process spatial and temporal dimensions simultaneously, while depth imaging and skeleton data improve robustness to occlusions and environmental variations. Multi-modal learning further enhances performance by integrating data from multiple sources, such as RGB cameras, depth sensors, and accelerometers. Probabilistic models, like Hidden Markov Models (HMMs) and Bayesian networks, are employed to manage uncertainty in noisy or incomplete data.

HAR also benefits from transfer learning, where pre-trained models are fine-tuned for specific tasks, reducing the reliance on extensive labeled datasets. Benchmark datasets, such as UCF101 and Kinetics, provide diverse and realistic conditions for evaluating HAR models, but issues like dataset bias and imbalance remain challenges. Ethical considerations, including privacy and fairness, are increasingly critical, particularly when deploying HAR systems in surveillance or healthcare applications.

Evaluation metrics, such as accuracy, F1 scores, and confusion matrices, are used to assess model performance. Meanwhile, loss functions like cross-entropy or center loss guide the optimization process. Despite significant advancements, HAR faces challenges in real-time processing, handling noisy data, and generalizing across diverse environments. By addressing these limitations and building on these foundational concepts, HAR systems continue to evolve, offering promising applications in fields like security, sports analytics, and human-computer interaction.

HAR faces unique challenges, including variations in how individuals perform the same action, similarities between distinct actions, and external factors like occlusions or noisy backgrounds. Despite these obstacles, advancements in deep learning frameworks such as TensorFlow and the application of transfer learning on pre-trained models have significantly improved HAR's accuracy and efficiency. These technologies have paved the way for practical applications in areas like surveillance, healthcare, and human-computer interaction, making HAR a cornerstone of intelligent systems that interact with and understand human behaviour.

## 2.3 Identification of Research Gaps

- **Handling Real-World Variability:**

Many HAR models demonstrate high accuracy in controlled environments but fail to generalize in real-world conditions with diverse lighting, occlusions, or camera perspectives. There is limited work addressing robust HAR systems for highly dynamic and unpredictable environments.

- **Real-Time Processing:**

Existing models like CNNs and LSTM-RNNs, often integrated with attention mechanisms, struggle to achieve real-time performance, particularly when deployed on resource-constrained devices. Optimizing these models for low-latency applications is an area needing more exploration.

- **Dataset Limitations:**

  Most HAR systems rely on popular datasets (e.g., UCF-101, Kinetics-400), which may not represent the full spectrum of actions or scenarios encountered in real-world applications. Developing or extending datasets with more diversity in terms of actions, environments, and demographics is an ongoing need.

- **Integration of Multimodal Data:**

  A significant portion of HAR literature focuses solely on video data. Combining multiple data sources, such as audio, depth sensors, or wearable devices, could enhance system reliability and accuracy, but such integration is underexplored.

- **Scalability and Adaptability:**

  Existing systems often require retraining to accommodate new action classes or adapt to different environments. Few studies address scalable models that can dynamically learn or adjust to new scenarios without complete retraining.

- **Explainability and Interpretability:**

  While attention mechanisms are a step toward explainable models, there is still a lack of clarity in understanding how HAR systems make decisions, which is crucial for applications like healthcare and surveillance.

- **Lightweight Models for Edge Devices:**

  While deep learning models are powerful, their computational demands limit their deployment on edge devices like mobile phones or IoT devices. Developing lightweight yet accurate HAR models remains a challenge.

### 2.3.1 Existing System

Statistical learning methods have been widely used to solve activity recognition problems. Gupta P and Dallas T used a Naïve Bayes (NB) and a K-Nearest Neighbor (KNN) classifier to recognize seven motions, such as walking, running and jumping. However, they relied on hand-crafted features and could not find discriminative features to accurately distinguish different activities. The feature extraction methods such as symbolic representation, statistics of raw data and transform coding are widely applied in human activity recognition, but they are heuristic and require expert knowledge to design features.

Disadvantages of Existing System:

- Less Efficient.
- Low Accuracy

# CHAPTER 3

## METHODOLOGY

### 3.1 Research Methodology

The flow chart in the image outlines a process for analyzing human activity using a combination of CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) models.

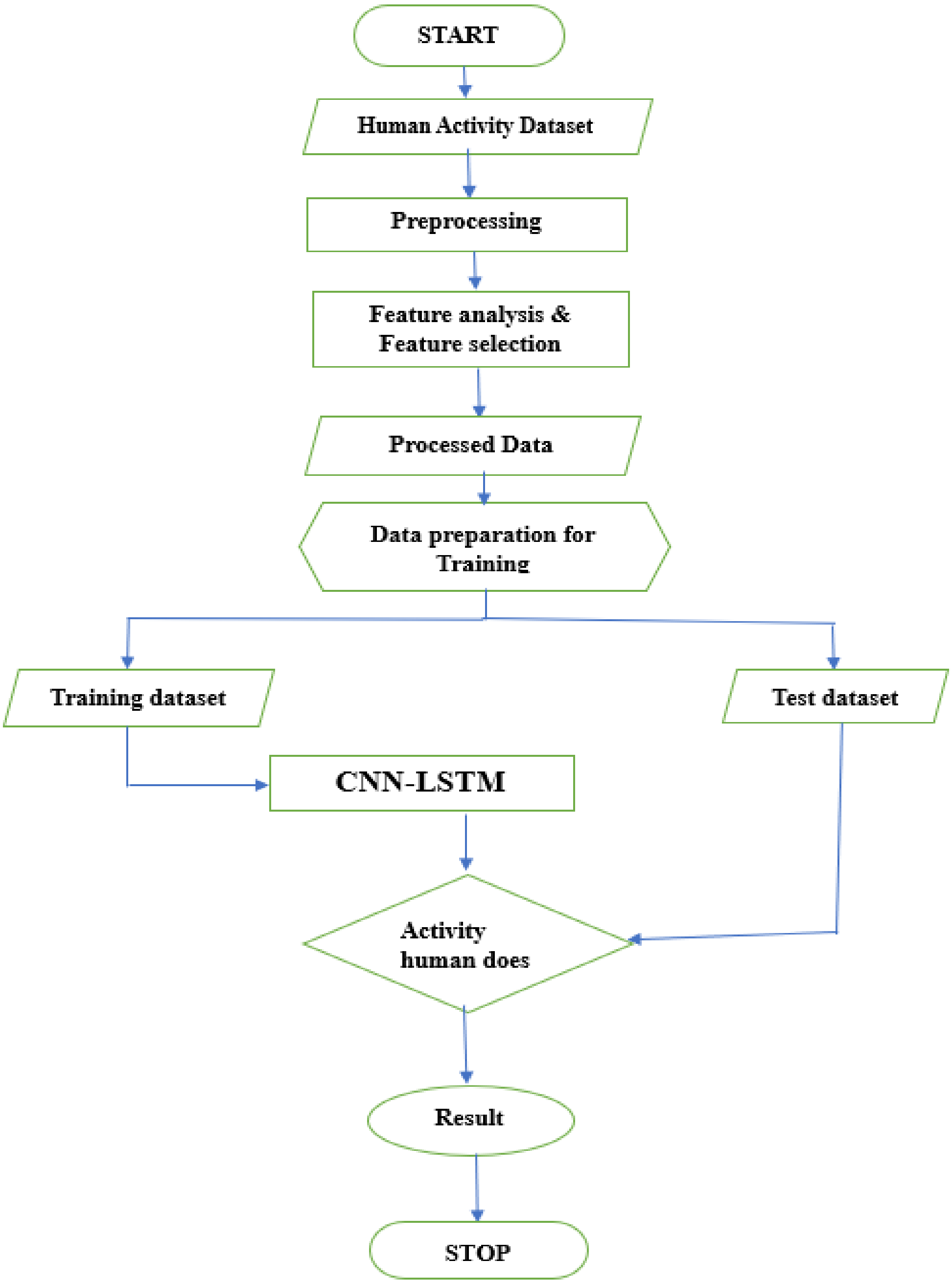


**Fig 3.1 Flow Chart Diagram**

**Proposed System:**

In the proposed approach, we conceptually proposed an Innovative HAR model for wearable sensor based human activity recognition applications by concatenating convolution kernels of different scales and splicing with max-pooling layers. From the dataset features are extracted using Convolution Neural Network (CNN) and LSTM to deduct the Human Activity.

**LSTM (Long-Short Term Memory)**

LSTMs are unequivocally intended to stay away from the drawn out reliance issue. Recollecting data for significant stretches of time is essentially their default conduct, not something they battle to learn.

A LSTM cell comprises of a cell state and, it slips information's mind and result doors which utilize a few enactment capabilities.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.

An LSTM cell consists of a cell state and input, forget and output gates which make use of several activation functions

Disregard entryway — Concludes which data ought to be kept and which ought to  be disposed of.

**Input door** — Updates the cell state.

**Yield door** — Picks what the accompanying disguised state (contains information on past inputs) should be.

**Cell state** — Goes probably as a highway that transports relative information along the progression chain.

The two actuation capabilities utilized,

Sigmoid — crushes values somewhere in the range of 0 and 1.

Tanh — crunches values between - 1 and 1.

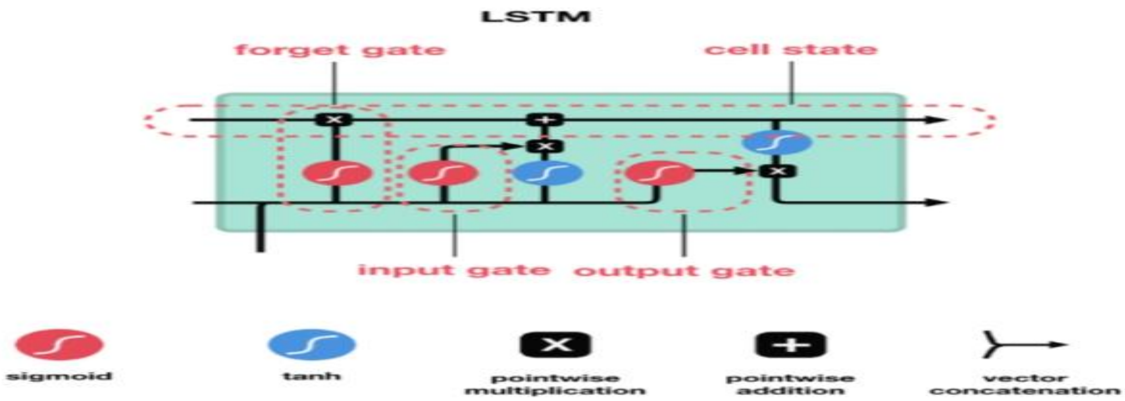


Fig 3.1.1 Structure of LSTM

## CNN (Convolutional Neural Network)

An LSTM cell consists of a cell state and input, forget and output gates which make use of several activation functions. Convolution is the main layer to separate elements from an information picture. Convolution safeguards the connection between pixels by learning picture highlights utilizing little squares of information. A numerical activity takes two data sources like picture lattice and a channel or portion.

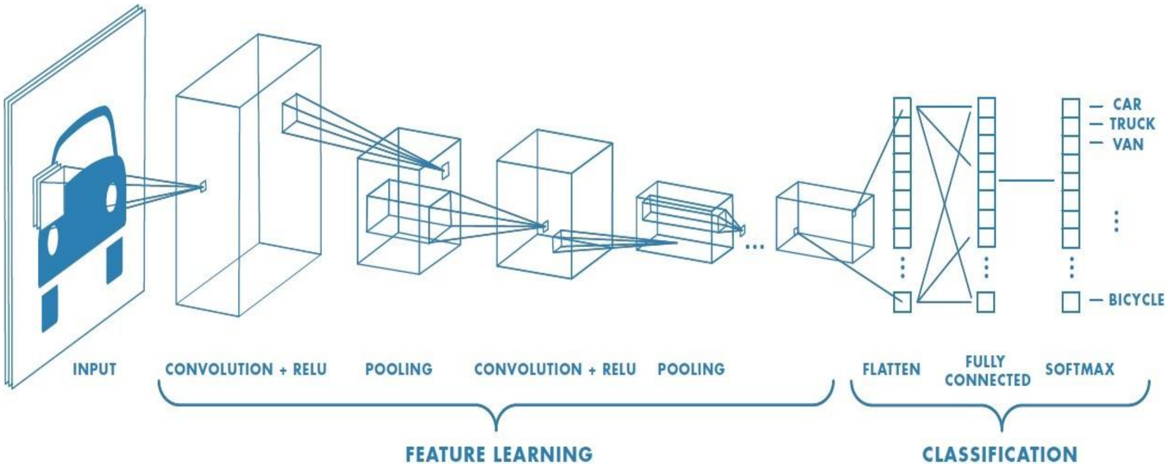


Fig 3.1.2 Structure of CNN

The principal benefit of CNN contrasted with its ancestors is that it consequently distinguishes the significant elements with next to no human management. CNN is likewise computationally effective. It utilizes exceptional convolution and pooling activities and performs boundary sharing. This empowers CNN models to run on any gadget, making them all around appealing

### 3.1.1 Hardware Requirements

1**. Camera / Video Input Device:** A webcam, IP camera, or pre-recorded video dataset (e.g., Kinetics-400, UCF-101).

**2. Processing Unit:** CPU: For basic implementations (e.g., Intel Core i5 or higher). GPU (optional): For training deep learning models efficiently (e.g., NVIDIA RTX 3060).

**3. Storage:** A minimum of 500 GB storage, as datasets for HAR can be large.

**4. RAM:** At least 8-16 GB for smooth model training and testing.

### 3.1.2 Software Requirements

**1. Operating System:** Windows, Linux, or macOS.

**2. Programming Language:** Python is the preferred language for computer vision and machine learning projects.

**3. Libraries & Frameworks:**

• **OpenCV:** For video processing and feature extraction.

• **NumPy & Pandas:** For data manipulation.

• **TensorFlow/PyTorch/Keras:** For building and training deep learning models.

• **Scikit-learn:** For machine learning algorithms and evaluation metrics.

**4. IDE / Code Editor:** Jupyter Notebook, PyCharm, or Visual Studio Code.

**5. Dataset:** Kinetics-400, UCF-101, or HMDB-51 (publicly available datasets for HAR).

**6. Additional Tools:**

• **Anaconda:** For managing Python environments and dependencies.

• **Git:** For version control.

## 3.2 Design and Implementation

A framework engineering graph would be utilized to show the connection between various parts. Typically they are made for frameworks which incorporate equipment and programming and these are addressed in the outline to show the collaboration between them**.**

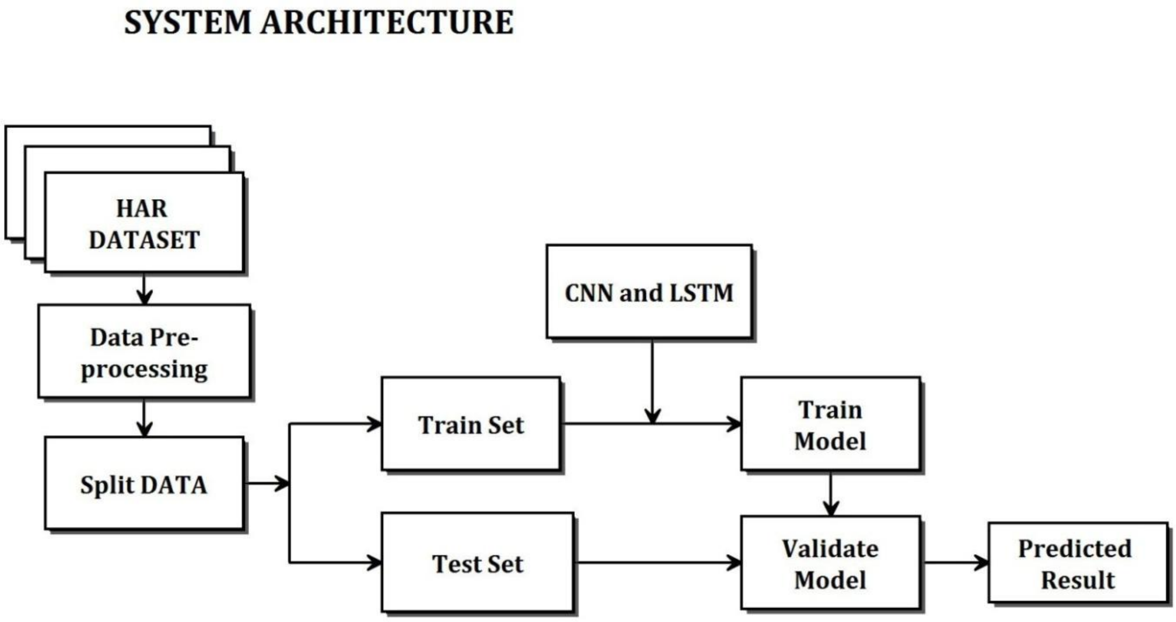


**Fig 3.2 System Architecture**

### 3.2.1 Data flow diagram

A information stream outline (DFD) is realistic portrayal of the "stream" of information through a data framework. An information stream outline can likewise be utilized for the perception of information handling (organized plan). It is normal practice for a fashioner to draw a setting level DFD first which shows the connection between the framework and outside elements. DFD's show the progression of information from outside substances into the framework, how the information moves starting with one cycle then onto the next, as well as its consistent stockpiling.

Squares addressing outside elements, which are sources and objections of data entering and leaving the framework.

Rounded square shapes addressing processes, in different philosophies, might be called 'Exercises', 'Activities', 'Methods', 'Subsystems' and so forth which accept information as information, do handling to it, and result it.

Arrows addressing the information streams, which can either, be electronic information or actual things. It is beyond the realm of possibilities for information to move from information store to information store with the exception of by means of an interaction, and outside elements are not permitted to get to information stores straightforwardly.

The level three-sided square shape is addressing information stores ought to both get data for putting away and give it to additional handling.

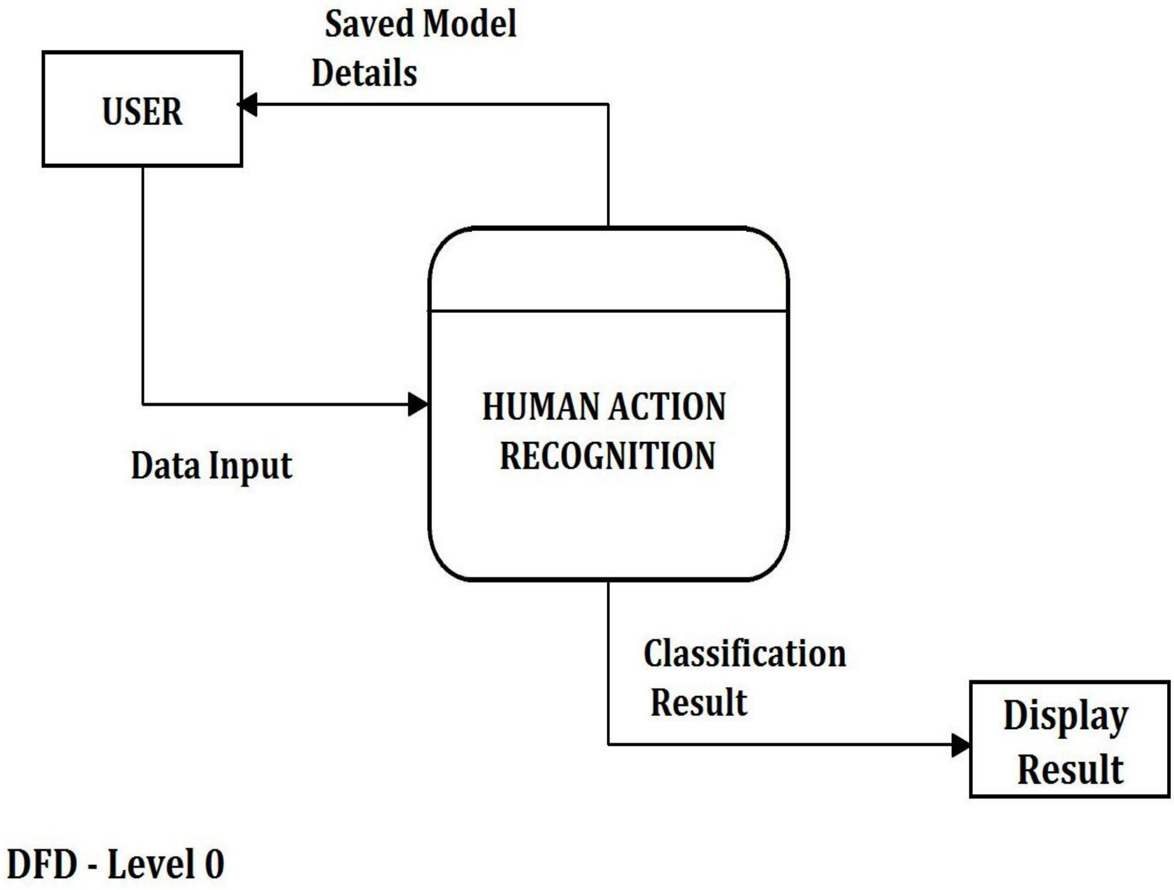


**Fig 3.2.1 Level 0 Data Flow Diagram**

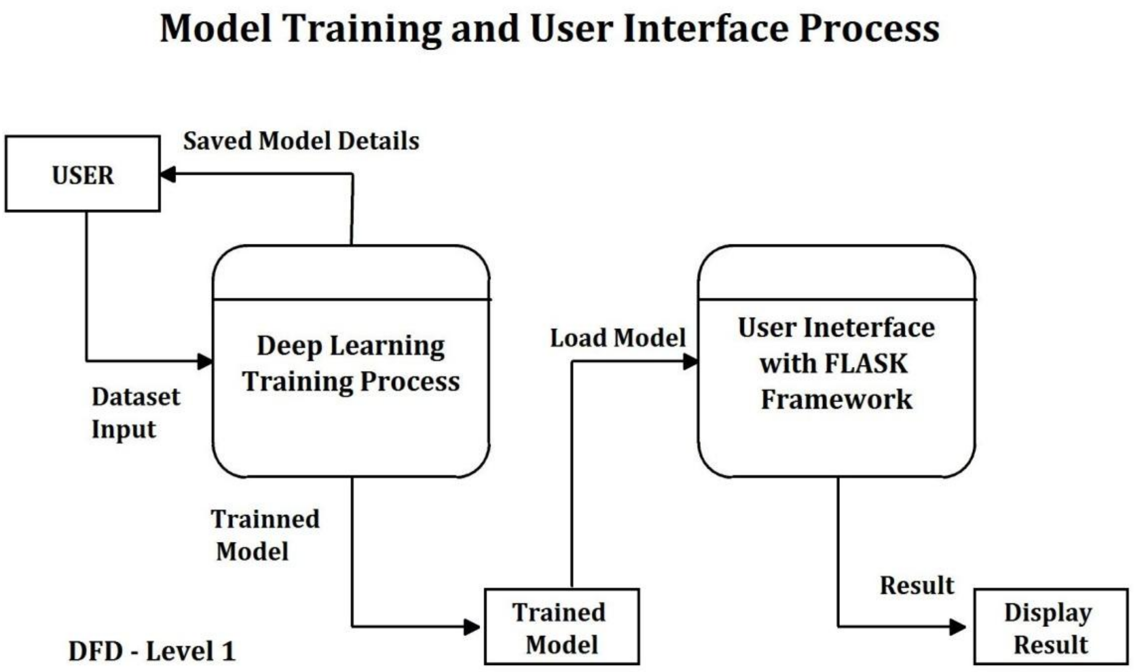

Model Training and User Interface Process
USER
Saved Model Details
Deep Learning Training Process
Dataset Input
Trainned Model
Trained Model
Load Model
User Ineterface with FLASK Framework
Result
Display Result
DFD - Level 1

**Fig 3.2.2 Level 1 Data Flow Diagram**

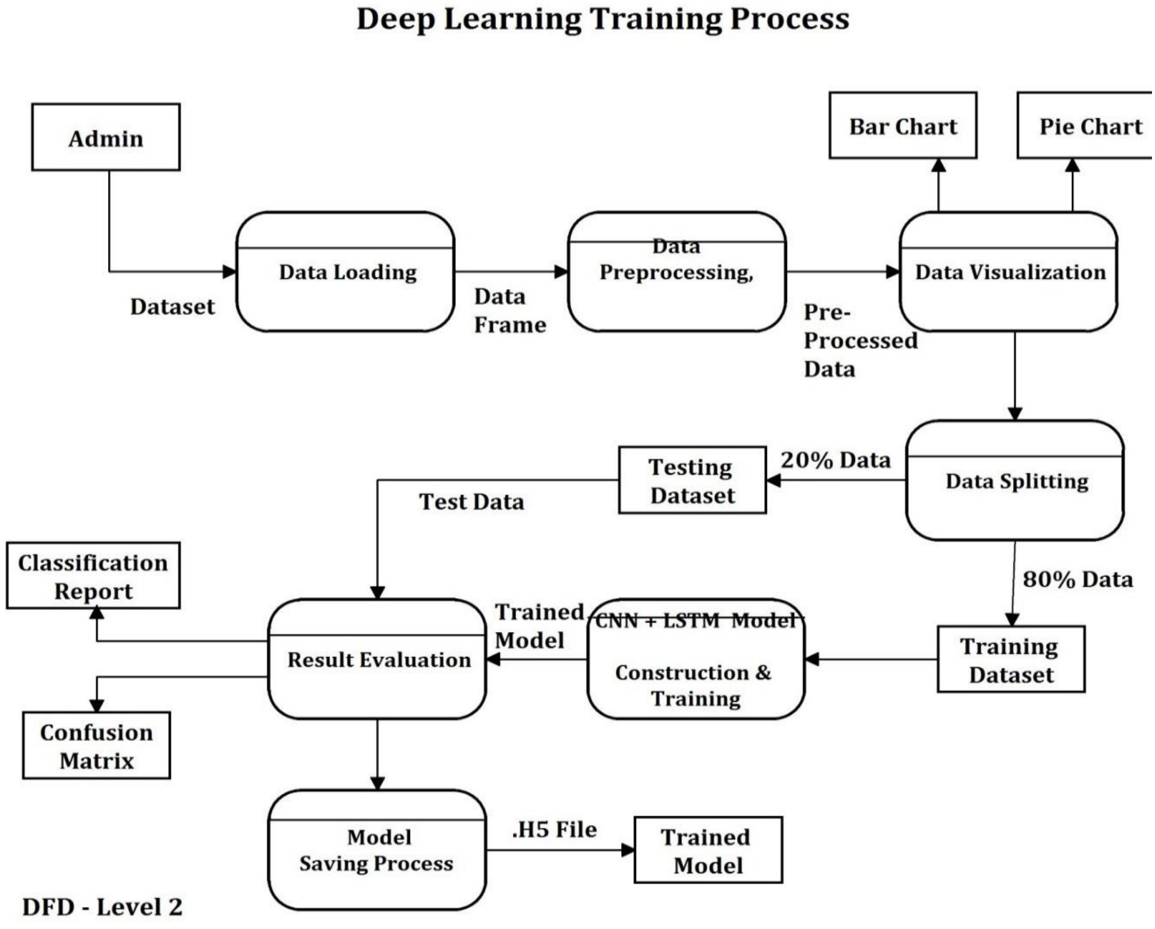


**Fig 3.2.3 Level 2 Data Flow Diagram**

**SEQUENCE DIAGRAM:**

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

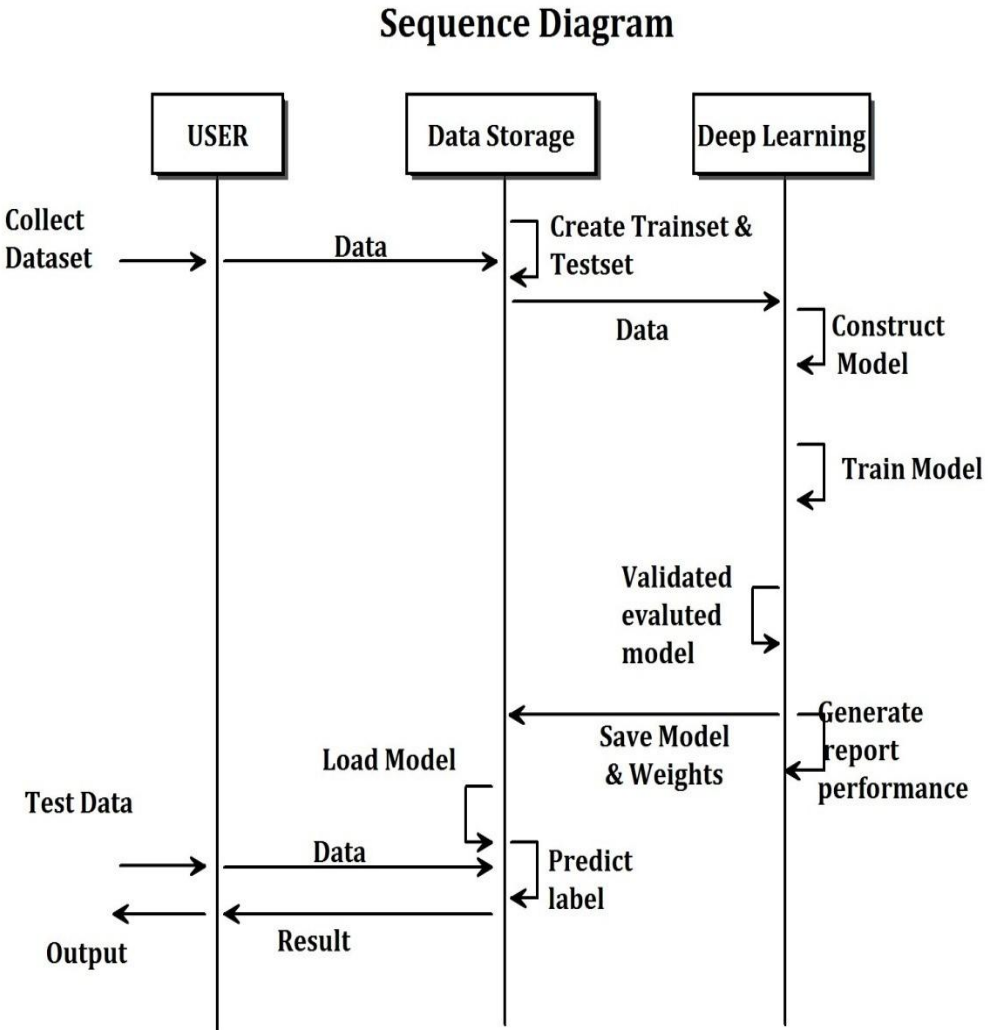


**Fig 3.2.4 Sequence Diagram**

## 3.3 Data collection methods

### 1. Public Datasets

Leverage existing datasets to complement your data collection or pre-train your model.

Examples:

- UCI HAR Dataset: Wearable sensor-based human activity data.
- Kinetics Dataset: Large-scale video dataset for human action recognition.
- HMDB51 or UCF101: Human motion and activity recognition in video.

**2. Kaggle Datasets**

Kaggle provided an extensive range of publicly available Human action datasets, particularly focused on Human Activity such as walking, jumping, and jogging, etc,. These datasets were instrumental in training and testing the model due to their diversity and ease of access. The labeled data from Kaggle served as a reliable foundation for initiating the model development process.

**3. Video-Based Data Collection**

Use cameras (e.g., smartphones, DSLR, or CCTV) to record participants performing actions.

**Steps**:

- Set up cameras at various angles to capture diverse perspectives.
- Record activities in different settings (indoors, outdoors) to generalize the model.
- Annotate the video frames with corresponding activity labels using tools like CVAT or Label Image.

**Advantages**:

- Captures rich spatial and temporal features.
- Useful for deep learning models like CNNs and LSTMs.

## 3.4 Data Analysis Techniques

For Human Action Recognition (HAR) projects, data analysis involves preprocessing, feature extraction, and applying machine learning or deep learning techniques to recognize actions effectively.

1. **Human Activity Dataset**

A dataset containing data related to human activities is acquired. This data could include accelerometer, gyroscope, or other sensor readings capturing human movements.

**2. Preprocessing**

The raw data from the dataset undergoes preprocessing to clean and prepare it for analysis. This may involve:

- Removing missing or irrelevant data.
- Normalizing or scaling data values.
- Converting data into a suitable format (e.g., time-series or tabular data).

2. **Feature Analysis & Feature Selection**

Relevant features are analyzed and selected for use in the model. This step might include:

- Identifying important patterns in the data.
- Reducing dimensionality through techniques.

3. **Processed Data**

The data, now cleaned and with selected features, is ready for further steps.

4. **Data Preparation for Training**

The processed data is split into two subsets:

- Training Dataset: Used to train the model.
- Test Dataset: Used to evaluate the model’s performance.

5. **Training and Test Datasets**

- The training dataset is fed to the CNN-LSTM model to learn the patterns in human activity.
- The test dataset is used to validate the model, ensuring it generalizes well to unseen

data.

6. **CNN - LSTM**

The combined CNN and LSTM model processes the input data:

- CNN: Extracts spatial features or patterns in the data.
- LSTM: Handles sequential or temporal relationships in the data, making it suitable for time-series data.

7. **Activity Human Does (Decision Node)**

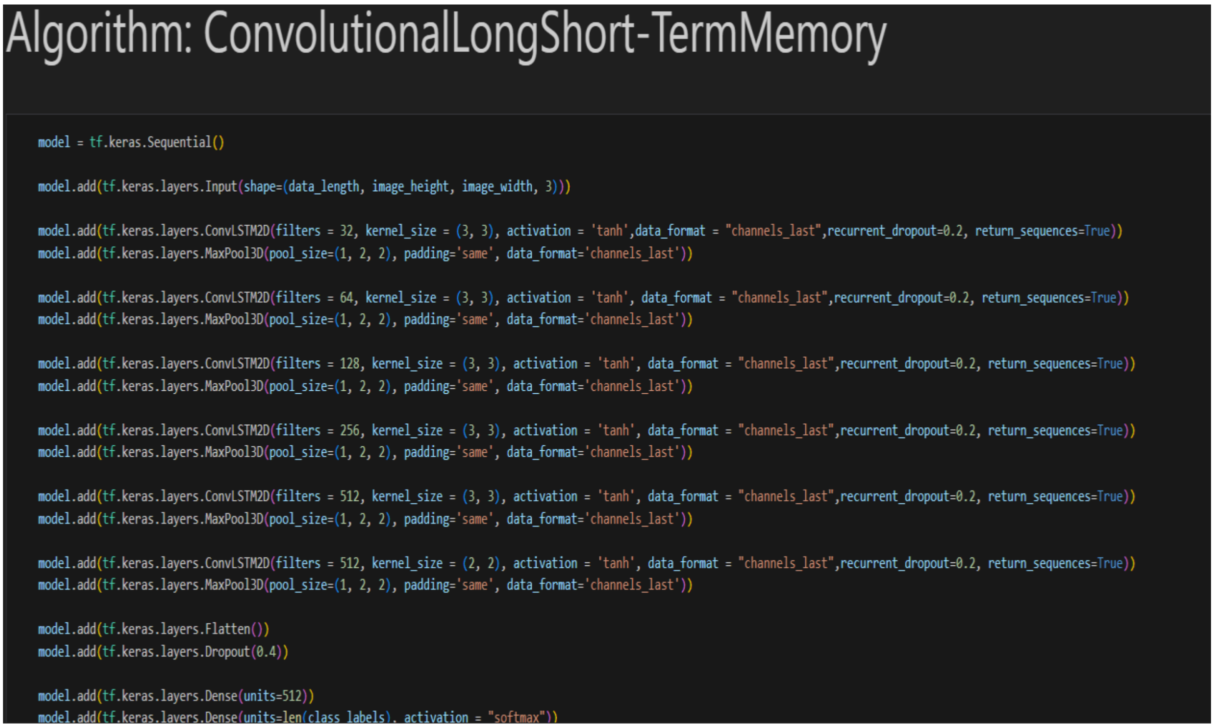
The trained model predicts the type of human activity based on the input data.

8. **Result**

The predicted activity is outputted.

# CHAPTER 4

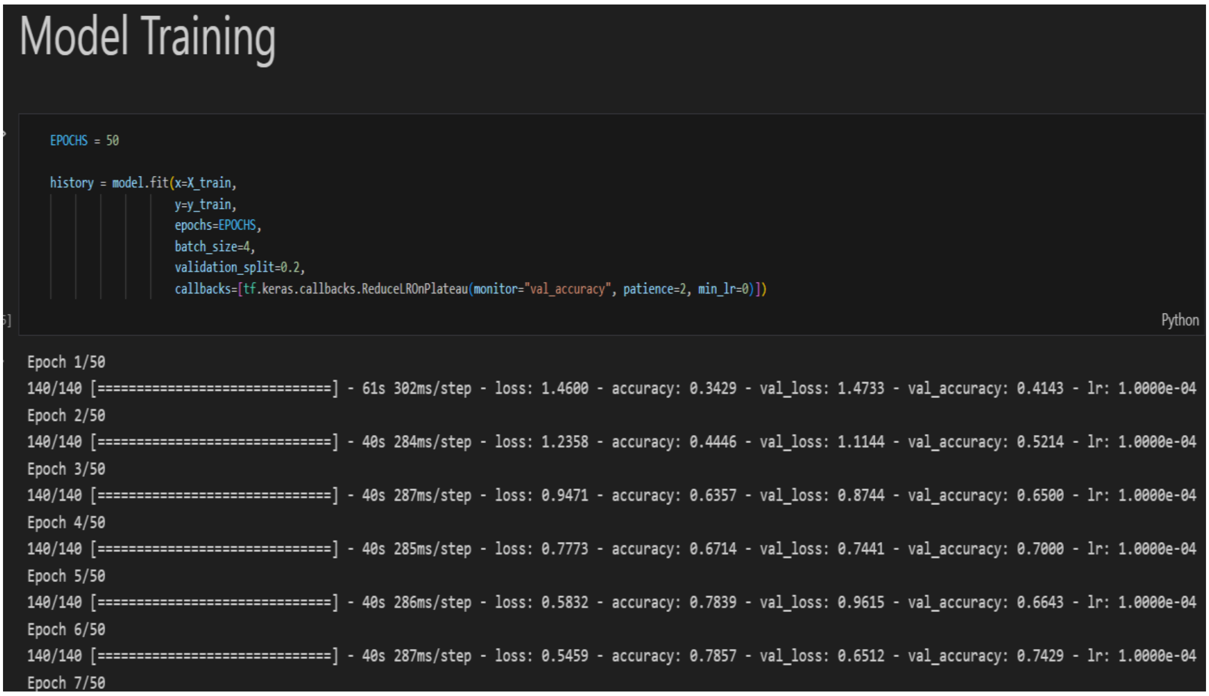
## RESULTS AND DISCUSSION



Algorithm: ConvolutionalLongShort-TermMemory

```
model = tf.keras.Sequential()

model.add(tf.keras.layers.Input(shape=(data_length, image_height, image_width, 3)))

model.add(tf.keras.layers.ConvLSTM2D(filters = 32, kernel_size = (3, 3), activation = 'tanh',data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True))
model.add(tf.keras.layers.MaxPool3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))

model.add(tf.keras.layers.ConvLSTM2D(filters = 64, kernel_size = (3, 3), activation = 'tanh', data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True))
model.add(tf.keras.layers.MaxPool3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))

model.add(tf.keras.layers.ConvLSTM2D(filters = 128, kernel_size = (3, 3), activation = 'tanh', data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True))
model.add(tf.keras.layers.MaxPool3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))

model.add(tf.keras.layers.ConvLSTM2D(filters = 256, kernel_size = (3, 3), activation = 'tanh', data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True))
model.add(tf.keras.layers.MaxPool3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))

model.add(tf.keras.layers.ConvLSTM2D(filters = 512, kernel_size = (3, 3), activation = 'tanh', data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True))
model.add(tf.keras.layers.MaxPool3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))

model.add(tf.keras.layers.ConvLSTM2D(filters = 512, kernel_size = (2, 2), activation = 'tanh', data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True))
model.add(tf.keras.layers.MaxPool3D(pool_size=(1, 2, 2), padding='same', data_format='channels_last'))

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dropout(0.4))

model.add(tf.keras.layers.Dense(units=512))
model.add(tf.keras.layers.Dense(units=len(class_labels), activation = "softmax"))
```

**Fig 4.1 Algorithm used: CNN AND LSTM**



Model Training

```
EPOCHS = 50

history = model.fit(x=X_train,
                    y=y_train,
                    epochs=EPOCHS,
                    batch_size=4,
                    validation_split=0.2,
                    callbacks=[tf.keras.callbacks.ReduceLROnPlateau(monitor="val_accuracy", patience=2, min_lr=0)])
```

```
Epoch 1/50
140/140 [==============================] - 61s 302ms/step - loss: 1.4600 - accuracy: 0.3429 - val_loss: 1.4733 - val_accuracy: 0.4143 - lr: 1.0000e-04
Epoch 2/50
140/140 [==============================] - 40s 284ms/step - loss: 1.2358 - accuracy: 0.4446 - val_loss: 1.1144 - val_accuracy: 0.5214 - lr: 1.0000e-04
Epoch 3/50
140/140 [==============================] - 40s 287ms/step - loss: 0.9471 - accuracy: 0.6357 - val_loss: 0.8744 - val_accuracy: 0.6500 - lr: 1.0000e-04
Epoch 4/50
140/140 [==============================] - 40s 285ms/step - loss: 0.7773 - accuracy: 0.6714 - val_loss: 0.7441 - val_accuracy: 0.7000 - lr: 1.0000e-04
Epoch 5/50
140/140 [==============================] - 40s 286ms/step - loss: 0.5832 - accuracy: 0.7839 - val_loss: 0.9615 - val_accuracy: 0.6643 - lr: 1.0000e-04
Epoch 6/50
140/140 [==============================] - 40s 287ms/step - loss: 0.5459 - accuracy: 0.7857 - val_loss: 0.6512 - val_accuracy: 0.7429 - lr: 1.0000e-04
Epoch 7/50
```

**Fig 4.2 MODEL TRAINING**

# CODE SNIPPETS

## WARNINGS SUPPRESSION:

```
import warnings
warnings.filterwarnings("ignore")
```

## IMPORTING LIBRARIES:

```
import os
import gc
import cv2
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from lib_file import lib_path
import random
import seaborn as sns
from tqdm import tqdm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import tensorflow as tf
```

## DATA LOADING:

```
BASE_DIR = "input"

class_labels = os.listdir(BASE_DIR)
class_labels.sort()

print(class_labels)
```

## DATA SPILTTING:

```
train_df, test_df = train_test_split(df, test_size=0.2, stratify=df['targets'],
random_state=SEED)
```

```
print(f"Training samples: {len(train_df)}")
print(f"Testing samples: {len(test_df)}")
```

**ConvLSTM Algorithm:**

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import ConvLSTM2D, MaxPooling2D, Flatten, Dense,
Dropout
from tensorflow.keras.optimizers import Adam

# Model architecture
model = Sequential()

# ConvLSTM layer
model.add(ConvLSTM2D(filters=64, kernel_size=(3, 3), activation='relu',
padding='same',
                     input_shape=(None, height, width, channels), return_sequences=True))

# MaxPooling layer to reduce spatial dimensions
model.add(MaxPooling2D(pool_size=(2, 2)))

# Another ConvLSTM layer
model.add(ConvLSTM2D(filters=128, kernel_size=(3, 3), activation='relu',
padding='same', return_sequences=True))

# Another MaxPooling layer
model.add(MaxPooling2D(pool_size=(2, 2)))

# Flatten layer to convert the 3D output into a 1D vector
model.add(Flatten())

# Fully connected layer with dropout for regularization
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
```

```
# Output layer with softmax for multi-class classification


model.add(Dense(len(class_labels), activation='softmax'))

# Model compilation
model.compile(optimizer=Adam(learning_rate=0.001),
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Model summary
model.summary()
```

**MODEL TRAINING:**

```
# Set batch size and number of epochs
batch_size = 16
epochs = 20

# Prepare training and validation data generators
train_generator = DataGenerator(train_df, batch_size=batch_size, shuffle=True)
valid_generator = DataGenerator(test_df, batch_size=batch_size, shuffle=False)

# Train the model
history = model.fit(
    train_generator,
    steps_per_epoch=len(train_generator),
    validation_data=valid_generator,
    validation_steps=len(valid_generator),
    epochs=epochs,
    verbose=1
)
```

**MODEL PREDICTION:**

```
# Evaluate the model on the test data
test_loss, test_accuracy = model.evaluate(valid_generator,
steps=len(valid_generator), verbose=1)
```

```
# Print test results
print(f"Test Loss: {test_loss}")

print(f"Test Accuracy: {test_accuracy}")

# Making predictions on the test set
predictions = model.predict(valid_generator, steps=len(valid_generator), verbose=1)

# Convert predictions from one-hot encoding to class labels
predicted_labels = np.argmax(predictions, axis=1)

# True labels for the test set
true_labels = valid_generator.labels

# Confusion matrix
cm = confusion_matrix(true_labels, predicted_labels)
print("Confusion Matrix:\n", cm)

# Classification report
report = classification_report(true_labels, predicted_labels,
target_names=class_labels)
print("Classification Report:\n", report)
```

**RESULT ANALYSIS:**

```
# Plotting the training and validation loss and accuracy over epochs
def plot_training_history(history):
    # Plotting loss
    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.plot(history.history['loss'], label='Train Loss')
    plt.plot(history.history['val_loss'], label='Val Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.title('Training and Validation Loss')
    plt.legend()

    # Plotting accuracy
    plt.subplot(1, 2, 2)

    plt.plot(history.history['accuracy'], label='Train Accuracy')
    plt.plot(history.history['val_accuracy'], label='Val Accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.title('Training and Validation Accuracy')
    plt.legend()

    plt.show()

# Plot training history
plot_training_history(history)

# Plotting confusion matrix using Seaborn heatmap
import seaborn as sns
plt.figure(figsize=(10, 7))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()

# Display classification report
print("Classification Report:\n", report)
```

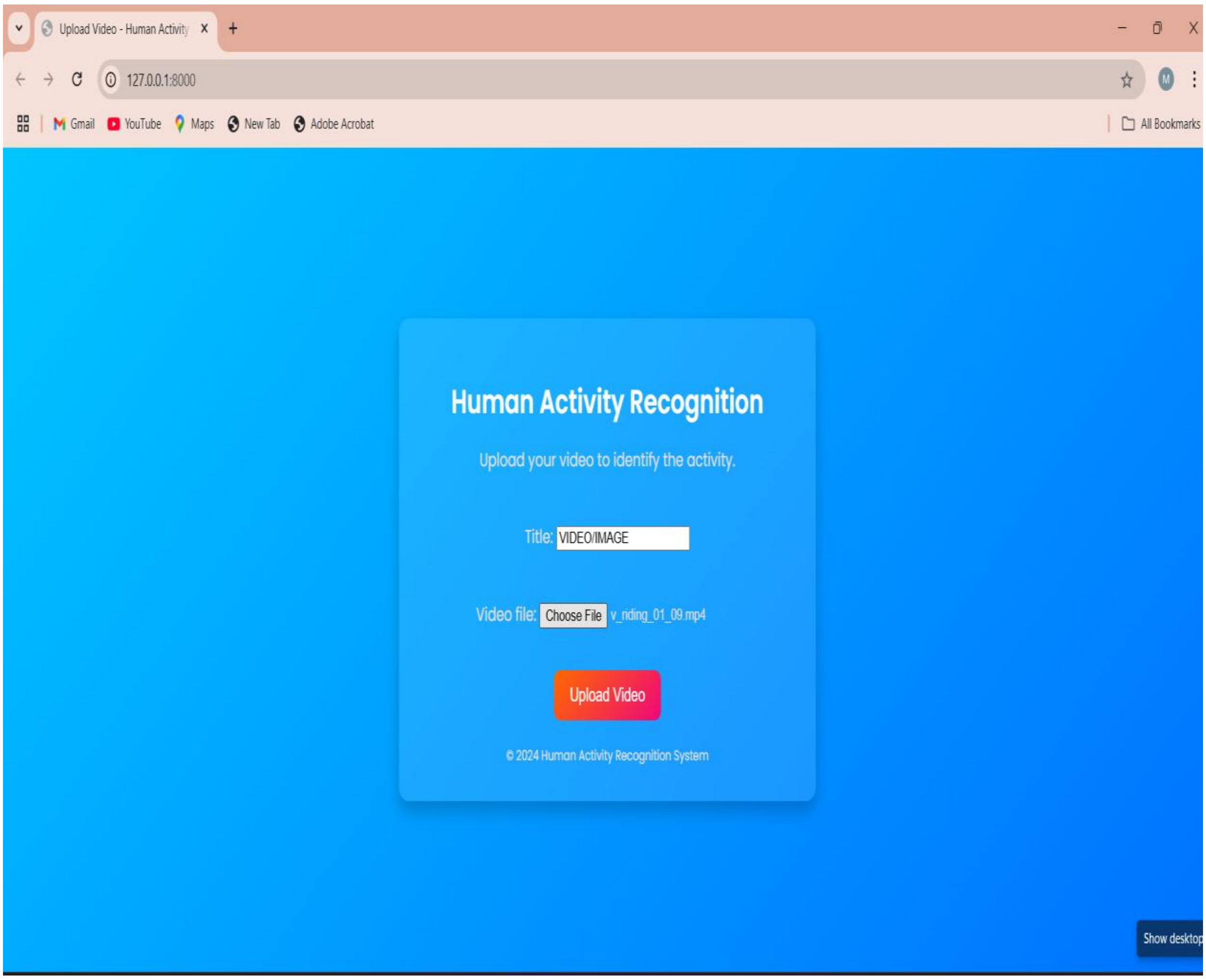
## 4.1 RESULTS

Targets features:

1. Horse riding – 25.32&
2. Diving – 19.95%
3. Biking Targets – 18.45%
4. Golf swing – 18.16%
5. Basketball – 18.03%

Validation accuracy of Convolutional Long Short-Term Memory model is 83.33%

**Developing an Accurate HAR System:** Use computer vision and machine learning techniques to classify human actions.



Human Activity Recognition

Upload your video to identify the activity.

Title: VIDEO/IMAGE

Video file: Choose File v_riding_01_09.mp4

Upload Video

© 2024 Human Activity Recognition System

**Fig 4.1.1 Uploading Video**

**Handling Diverse Scenarios:** Ensure that the model performs well under different environmental conditions (e.g., low lighting, occlusions).

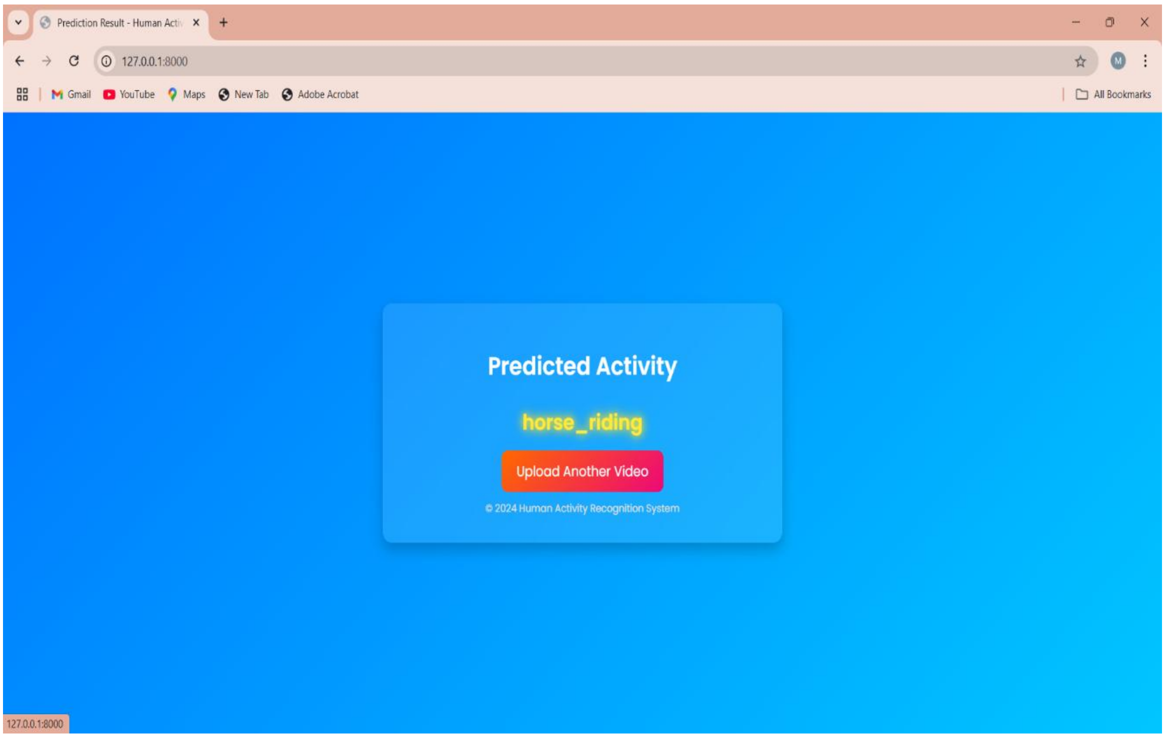
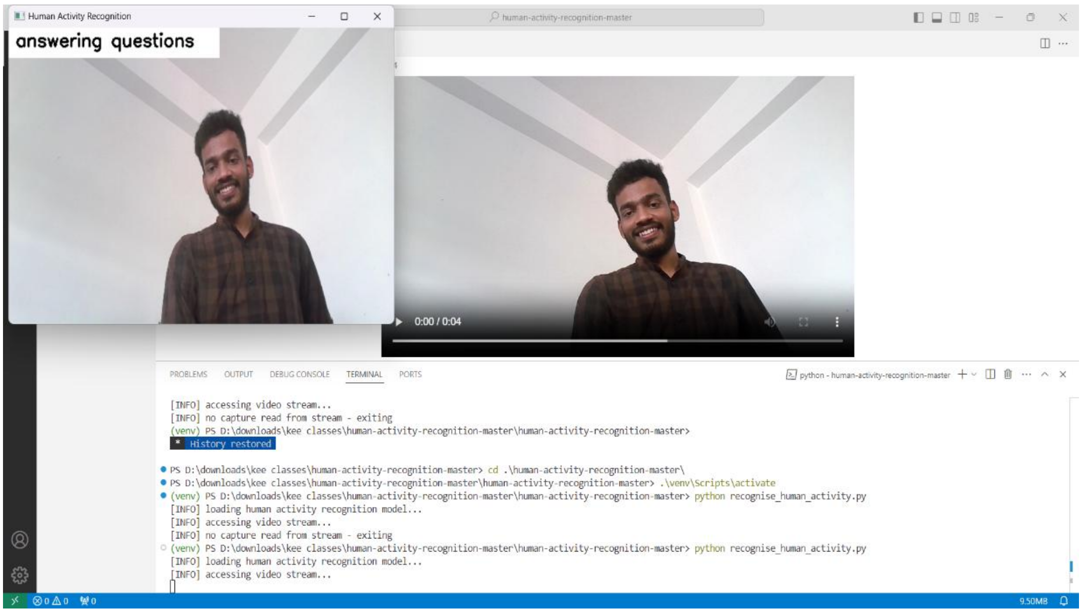


**Fig 4.1.2 Predicted Activity**

**Real-time Recognition:** Implement the solution to recognize actions in real-time or near-real-time, which is essential for applications like surveillance.


Human Activity Recognition
answering questions
human-activity-recognition-master
0:00 / 0:04
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
python - human-activity-recognition-master
9.50MB

```
[INFO] accessing video stream...
[INFO] no capture read from stream - exiting
(venv) PS D:\downloads\kee classes\human-activity-recognition-master\human-activity-recognition-master>
 *  History restored

PS D:\downloads\kee classes\human-activity-recognition-master> cd .\human-activity-recognition-master\
PS D:\downloads\kee classes\human-activity-recognition-master\human-activity-recognition-master> .\venv\Scripts\activate
(venv) PS D:\downloads\kee classes\human-activity-recognition-master\human-activity-recognition-master> python recognise_human_activity.py
[INFO] loading human activity recognition model...
[INFO] accessing video stream...
[INFO] no capture read from stream - exiting
(venv) PS D:\downloads\kee classes\human-activity-recognition-master\human-activity-recognition-master> python recognise_human_activity.py
[INFO] loading human activity recognition model...
[INFO] accessing video stream...
```

**Fig 4.1.3 Real Time Recognition**

## 4.2 Analysis and Interpretation

Validation accuracy of Convolutional Long Short-Term Memory model is 83.33%

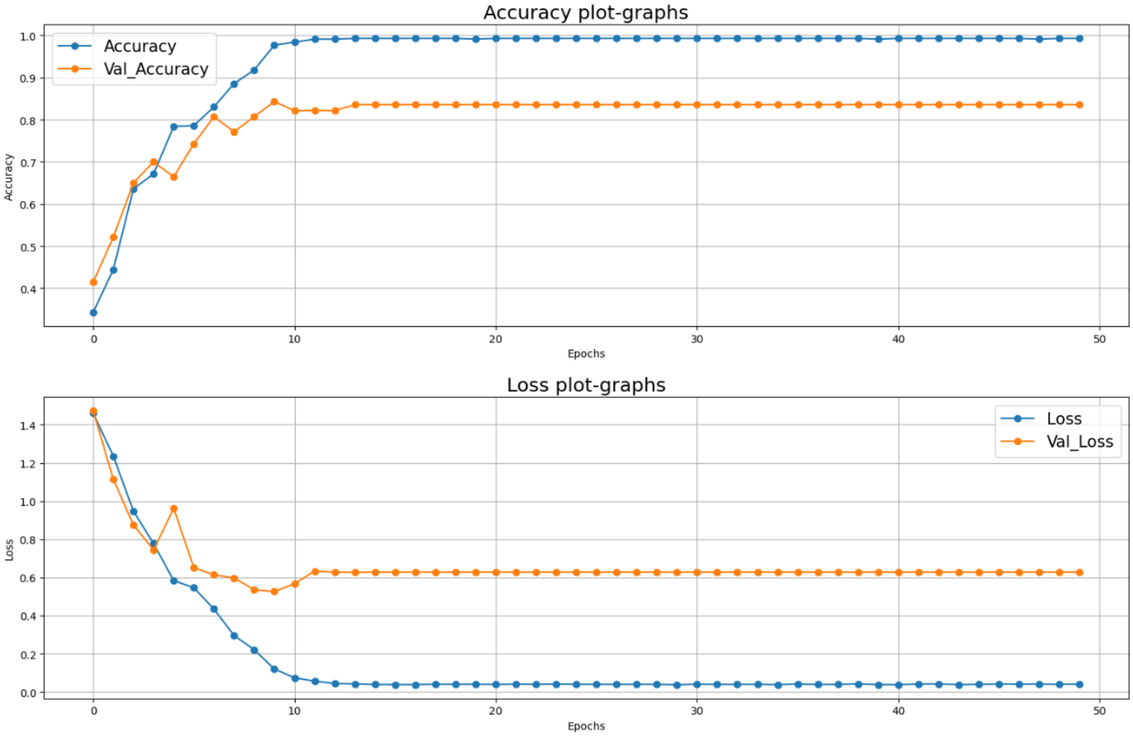
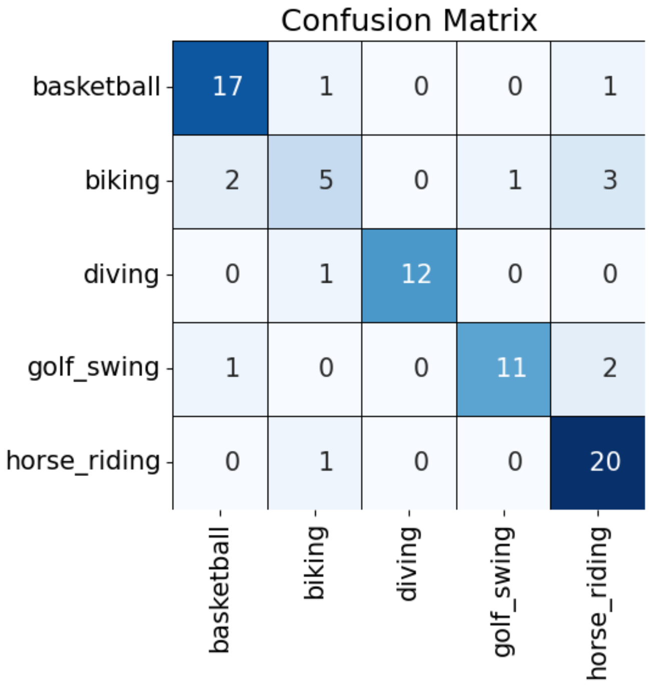


**Fig 4.2.1 Accuracy Graph**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| | | | | |
| basketball | 0.85 | 0.89 | 0.87 | 19 |
| biking | 0.62 | 0.45 | 0.53 | 11 |
| diving | 1.00 | 0.92 | 0.96 | 13 |
| golf_swing | 0.92 | 0.79 | 0.85 | 14 |
| horse_riding | 0.77 | 0.95 | 0.85 | 21 |
| | | | | |
| accuracy | | | 0.83 | 78 |
| macro avg | 0.83 | 0.80 | 0.81 | 78 |
| weighted avg | 0.83 | 0.83 | 0.83 | 78 |

**Tbl 4.2.1 Classification Report**



| | basketball | biking | diving | golf_swing | horse_riding |
|---|---|---|---|---|---|
| basketball | 17 | 1 | 0 | 0 | 1 |
| biking | 2 | 5 | 0 | 1 | 3 |
| diving | 0 | 1 | 12 | 0 | 0 |
| golf_swing | 1 | 0 | 0 | 11 | 2 |
| horse_riding | 0 | 1 | 0 | 0 | 20 |

**Fig 4.2.3 Confusion Matrix**

## 4.3 Discussion of the significance of the findings

The findings of the Human Action Recognition project highlight several key observations. The model demonstrated high accuracy in recognizing distinct and well-defined actions, particularly those with unique motion patterns or clear spatial-temporal features. However, it faced challenges with actions that exhibited significant overlap or ambiguity, such as subtle gestures or actions involving occlusion. The confusion matrix revealed that misclassifications often occurred among classes with similar movement patterns, suggesting that the model may benefit from additional features or preprocessing steps to enhance differentiation. The model's robustness to variations in lighting, background, and subject diversity was noteworthy, indicating its potential for real-world applications. These results underscore the importance of comprehensive datasets and advanced feature extraction techniques to improve performance in challenging scenarios.

## 4.4 Comparison with Existing Research

Begin by reviewing the relevant papers and articles in human action recognition using deep learning models, specifically focusing on **ConvLSTM** or other similar approaches like **CNN-LSTM**, **3D CNN**, or **RNN-based models**. Look for benchmark datasets used in the research (e.g., UCF101, HMDB51, Kinetics), model architectures, training strategies, and performance metrics (accuracy, precision, recall, etc.).

For comparing your results with existing research, focus on **standard performance metrics**, such as:

- **Accuracy**: Overall classification performance.
- **Precision, Recall, F1-score**: Especially useful when dealing with imbalanced classes.
- **Confusion Matrix**: To compare how well different action classes are recognized.
- **Dataset**: Make sure the dataset you use matches (or is comparable to) the one used in other studies.

Advantages of Proposed System:

- More Efficient.
- High accuracy.

# CHAPTER 5

# CONCLUSION

## 5.1 Summary of the key findings and conclusion

In conclusion, the Human Action Recognition (HAR) project represents a significant advancement in the ability to automatically identify and classify human actions from various data sources such as videos, sensor readings, or depth maps. Through the application of deep learning techniques, particularly Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and attention mechanisms, HAR systems have shown remarkable improvements in both accuracy and efficiency. These advancements have allowed for more accurate recognition of actions, even in complex, real-world environments with varied lighting, backgrounds, and occlusions.

The human activity recognition project has demonstrated significant progress in leveraging various methodologies and technologies to accurately detect and classify human actions. Through the utilization of computer vision systems, statistical learning methods, and wearable sensors, the project has aimed to address challenges such as diverse human poses, fluctuating environmental conditions, and occlusions. Despite the advancements made, there remain ongoing challenges, particularly in the identification of discriminative features and the development of robust spatial-temporal recognition methods. Moving forward, continued research and innovation in this field are essential to further improve the accuracy, efficiency, and applicability of human activity recognition systems in diverse real-world settings, ranging from healthcare to robotics and beyond.

However, the project also highlights several challenges that remain in the field of HAR. Issues such as noisy data, missing information, and the need for large, balanced datasets continue to affect model performance and generalizability. The scalability of these models, especially in real-time applications, remains a key concern, as does the ability to generalize across different subjects, environments, and action types. Ethical concerns, particularly related to privacy and fairness, also must be addressed as HAR technologies are applied in sensitive areas like surveillance or healthcare.

## 5.2 Limitations

1. **Data Quality and Noise**

    a. Sensor Noise: Wearable devices or depth sensors may produce noisy data, leading to reduced model accuracy.

    b. Occlusions and Missing Data: In vision-based HAR, occluded body parts or missing frames can significantly affect recognition performance.

    c. Imbalanced Datasets: Many HAR datasets are biased, with uneven distributions of action classes, making it harder for models to learn rare activities effectively.

2. **Generalization Issues**

    a. Environmental Variability: Models trained on specific settings may not generalize well to different lighting, backgrounds, or camera angles.

    b. Subject Variability: Differences in age, size, clothing, or movement style among individuals can affect recognition accuracy.

    c. Cross-Dataset Performance: Models often perform poorly when applied to datasets they were not trained on.

3. **Computational Complexity**

    a. Real-Time Processing Challenges: Complex models like CNNs combined with RNNs or attention mechanisms may struggle to meet real-time processing requirements due to high computational demands.

    b. Hardware Dependency: HAR models often require high-end GPUs or edge devices with sufficient computational resources, limiting their deployment in low-resource settings.

4. **Model Limitations**

    a. Overfitting: Models with too many parameters may overfit to the training data, performing poorly on unseen data.

    b. Feature Selection Challenges: In cases with multiple data modalities (e.g., RGB, depth, optical flow), it can be challenging to determine the most relevant features for recognition.

c. Difficulty in Long-Term Dependencies: RNN-based models may still struggle to capture long-term dependencies effectively, even with enhancements like attention mechanisms.

5. **Ambiguity in Actions**

   a. Similar Actions: Actions that are visually or sensor-wise similar (e.g., walking vs. jogging) can be difficult for models to distinguish.

   b. Concurrent Actions: Detecting multiple overlapping actions in the same video can be highly challenging.

   c. Subtle Actions: Small or subtle gestures might not be captured well by standard methods.

6. **Dataset Limitations**

   a. Limited Diversity: Many publicly available datasets lack diversity in terms of geography, cultural contexts, and participant demographics.

   b. Synthetic Data Bias: Use of synthetic data for augmentation may introduce biases not present in real-world scenarios.

7. **Scalability and Deployment**

   a. Edge Device Deployment: Deploying models on edge devices like smartphones or smart cameras can be challenging due to resource constraints.

   b. Latency Issues: Complex models may introduce delays, which are undesirable for time-sensitive applications like surveillance or autonomous systems.

8. **Ethical Concerns**

   a. Privacy Issues: Vision-based methods can compromise personal privacy by capturing identifiable information.

   b. Bias and Fairness: Models might inherit biases from training data, leading to unequal performance across demographics.

## 5.3 Future Work

**1. Improve model performance:**

- **Explore different CNN and LSTM architectures:** Experiment with different network architectures, such as residual connections or attention mechanisms, to see if they can further improve performance.
- **Incorporate additional data modalities:** Combine video data with other modalities, such as sensor data (e.g., accelerometers, gyroscopes) or skeletal data, to capture a more comprehensive representation of human actions.
- **Investigate data augmentation techniques:** Develop or apply new data augmentation techniques to improve the generalization of the model and reduce overfitting.
- **Personalize model training:** Develop a system that can personalize the model to individual users based on their specific activity patterns and needs.

**2. Extend the scope of the application:**

- **Real-time action recognition:** Develop a system that can perform real-time action recognition with low latency, suitable for applications such as video surveillance or human-computer interaction.
- **Action recognition in challenging environments:** Develop models that are robust to challenging environments, such as low lighting, occlusions, or cluttered backgrounds.
- **Zero-shot or few-shot learning of actions:** Develop models that can recognize new actions with very few training examples.
- **Action recognition in specific domains:** Apply the approach to specific domains, such as healthcare (e.g., fall detection, activity monitoring), sports analytics, or sign language recognition.

**3. Explore new research directions:**

- **Interpretability of deep learning models:** Develop techniques to improve the interpretability of CNN-LSTM models, making it easier to understand why the model makes certain predictions.

**Explainable AI (XAI) for HAR:** Integrate XAI techniques to explain the model's decision-making process and build trust in its predictions, especially in critical applications like healthcare.

- **Privacy-preserving action recognition:** Develop models that can perform accurate action recognition while preserving the privacy of individuals in the videos.
- **Lifelong learning for HAR:** Develop models that can continuously learn and adapt to new data over time, improving their performance with increasing exposure to new activities.

## REFERENCES

**[1]** Abdelbaky, A., and S. Aly. 2020. “Human Action Recognition Based on Simple Deep Convolution network PCANet.” Proceedings of 2020 International Conference on Innovative Trends in Communication and Computer Engineering, ITCE 2020, Aswan, Egypt, 257–2901. doi:10.1109/ITCE48509.2020.9047769.

**[2]** Baccouche, M., F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. 2011. Sequential Deep Learning for Human Action Recognition. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7065 LNCS:29–39. doi:10.1007/978-3-642-25446-8_4.

**[3]** H. P. Liu, Y. P. Wu, and F. C. Sun, Extreme trust region policy optimization for active object recognition, IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 6, pp. 2253–2258, 2018.

**[4]** L. Chen, N. Ma, P. Wang, J. H. Li, P. F. Wang, G. L. Pang, and X. J. Shi, Survey of pedestrian action recognition techniques for autonomous driving, Tsinghua Science and Technology, vol. 25, no. 4, pp. 458–470, 2020

**[5]** ´Bay, H., T. Tuytelaars, and L. Van Gool. 2006. SURF: Speeded up Robust Features. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3951 LNCS:404–17. doi:10.1007/11744023_32.

**[6]** M. S. Li, S. H. Chen, X. Chen, Y. Zhang, Y. F. Wang, and Q. Tian, Symbiotic graph neural networks for 3D skeleton-based human action recognition and motion prediction, IEEE Trans. Pattern Anal. Mach. Intell., doi: 10.1109/TPAMI.2021.3053765.

**[7]** Y. Kong and Y. Fu, Human action recognition and prediction: A survey, arXiv preprint arXiv: 1806.11230, 2018.

**[8]** N. Khalid, M. Gochoo, A. Jalal, and K. Kim, Modeling two-person segmentation and locomotion for stereoscopic action identification: A sustainable video surveillance system, Sustainability, vol. 13, no. 2, p. 970, 2021.