HUMAN ACTIVITY RECOGNITION THROUGH ENSEMBLE LEARNING OF MULTIPLE CONVOLUTIONAL NEURAL NETWORKS

A PROJECT REPORT

Submitted by

LOKESH GURRAM **211420104087**

KAMULURI AARIF AHMED **211420104122**

PONNAGANTI THARAK SRINIVAS MANOJ **211420204194**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **Human Activity Recognition Through Ensemble Learning Of Multiple Convolutional Neural Networks** is the bonafide work of **LOKESH GURRAM(211420104087),KAMULURI AARIF AHMED(211420104122)PONNAGANTI THARAK SRINIVAS MANOJ (211420104194)** who carried out the project work under my super vision

Signature of the HOD with date

DR L. JABASHEELA M.E., Ph.D.,

Professor and Head,
Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai – 123

Signature of the Supervisor with date

S. PREENA JACINTH SHALOM M.E.,

Assistant Professor
Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai – 123

Submitted for the Project Viva – Voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We PONNAGANTI THARAK SRINIVAS MANOJ (211420104194), LOKESH GURRAM (211420104087) KAMULURI AARIF AHMED (211420104122) hereby declare that this project report titled **Human Activity Recognition through Ensemble Learning of Multiple Convolutional Neural Networks**, under the guidance of **MRS.S. PREENA JACINTH SHALOM** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

LOKESH GURRAM (211420104087)

KAMULURI AARIF AHMED (2114201040122)

PONNAGANTI THARAK SRINIVAS MANOJ(211420104194)

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr.P.CHINNADURAI, M.A., Ph.D**., for his benevolent words and fervent encouragement.His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of the project.

We would like to express our sincere thanks to coordinator **Dr. PUGHAZENDI N, M.E., Ph.D.,** and the project guide **S. PREENA JACINTH SHALOM, M.E** and I also thank my parents, friends, and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

LOKESH GURRAM (211420104087)

KAMULURI AARIF AHMED (211420104122)

PONNAGANTI THARAK SRINIVAS MANOJ (211420104194)

ABSTRACT

Recognition of Human Activity is a field concerned with the recognition of physical human activities based on the interpretation of sensor data, including one-dimensional time series data. Traditionally, hand-crafted features are relied upon to develop the machine learning models for activity recognition. However, that is a challenging task and requires a high degree of domain expertise and feature engineering. With the development in deep neural networks, it is much easier as models can automatically learn features from raw sensor data, yielding improved classification results. In this paper, we present a novel approach for human activity recognition using ensemble learning of multiple convolutional neural network (CNN) models. Three different CNN models are trained on the publicly available dataset and multiple ensembles of the models are created. The ensemble of the first two models gives an accuracy of 94% which is better than the methods available in the literature. The main role of intelligent video systems is to instinctively recognize and store the activities that humans are performing in the video stream. Most individual-centered challenges, such as those involving medical and one-on-one support, require inferring a variety of human actions, from basic to complicated. Thus, a clear classification of technologies for detecting is necessary for the methodical development of platforms that recognize behavior by humans. Human activity recognition (HAR) aims to recognize activities from a series of observations on the actions of subjects and environmental conditions. The vision-based HAR research is the basis of many applications including video surveillance, health care, and human-computer interaction (HCI). In the present research, we offer an innovative method that uses ensemble learning of several convolutional neural network (CNN) models to recognize human activities. Using the publicly accessible dataset, three distinct CNN algorithms are trained, and several ensembles of the models are produced. A success rate of 94% is obtained by the ensemble of the first two algorithms, which is higher than the techniques found in the existing literature.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
LCNN	Lookup-based Convolutional Neural Network
RNN	Recurrent Neural Network
DEX	Dalvik Executables
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hyper Text Transfer Protocol
ADT	Android Development Tool

CHAPTER 1

INTRODUCTION

In today's world, computers have become an important aspect of life and are used in various fields however, the systems and methods that we use to interact with computers are outdated and have various issues, which we will discuss a little later in this paper. Hence, a very new field trying to overcome these issues has emerged namely HUMAN COMPUTER INTERACTIONS (HCI). Although computers have made numerous advancements in both fields of Software and Hardware, Still the basic way in which Humans interact with computers remains the same, using a basic pointing device (mouse) and Keyboard or an advanced Voice Recognition System, or maybe Natural Language processing in really advanced cases to make this communication more human and easy for us. Our proposed project is activity recognition of humans based on their pose and movements. The proposed system uses images as objects where the images are broken into 2.5D coordinate axes and detection is based on the learning of the axes frames. Human activities have an inherent hierarchical structure that indicates the different levels of it, which can be considered as a three-level categorization. First, for the bottom level, there is an anatomic element and these action primitives constitute more complex human activities. After the action primitive level, the action/activity comes as the second level. Finally, the complex interactions form the top level, which refers to the human activities that involve more than two persons and objects. In this paper, we follow this three-level categorization namely action primitives, actions/activities, and interactions. This three-level categorization varies a little from previous surveys and maintains a consistent theme. Action primitives are those atomic actions at the limb level, such as "stretching the left arm," and "raising the right leg." Atomic actions are performed by a specific part of the human body, such as the hands, arms, or upper body part. With the upgrades of camera devices, especially the launch of RGBD cameras in the year 2010, depth image-based representations have been a new research topic and have drawn growing concern for years.

1.1 DIGITAL IMAGE PROCESSINCG

Image processing is reckoned as one of the most rapidly involving fields of the software industry with growing applications in all areas of work. It holds the possibility of developing the ultimate machine in the future, which would be able to perform the visual function of living beings. As such, it forms the basis of all kinds of visual automation.

1.2 ACTIVITY DETECTION AND RECOGNITION

A Human activity Recognition System recognizes the Shapes and or orientation depending on implementation to task the system into performing some job. Movement is a form of nonverbal information. A person can make numerous movements at a time. As humans through vision perceive human gestures and through computers, we need an image and a camera, it is a subject of great interest for computer vision researchers such as performing an action based on the activity of the person.

1.2.1 DETECTION

Activity detection is related to the position of a human at a given time in a stiff image or sequence of images i.e. moving images. In the case of a moving sequence, it can be followed by tracking the movement in the scene but this is more relevant the applications such as sign language. The underlying concept of activity detection is that human eyes can detect objects, which machines cannot, with as much accuracy as that of humans. From a machine point of view, it is just like a man fumbling with his senses to find an object. As shown in Figure 1.1 the accuracy of the image processed can vary depending on the lighting condition in which the image was uploaded.

1.2.2 RECOGNITION

Human activity detection and recognition has been a significant subject in the field of computer vision and image processing in the past 30 years. There have been considerable achievements and numerous approaches developed in this field.

Movement Recognition is a topic in computer science and language technology to interpret human actions via mathematical algorithms using images and camera samples. However, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques. However, the typical approach of a recognition system is shown in the below figure:

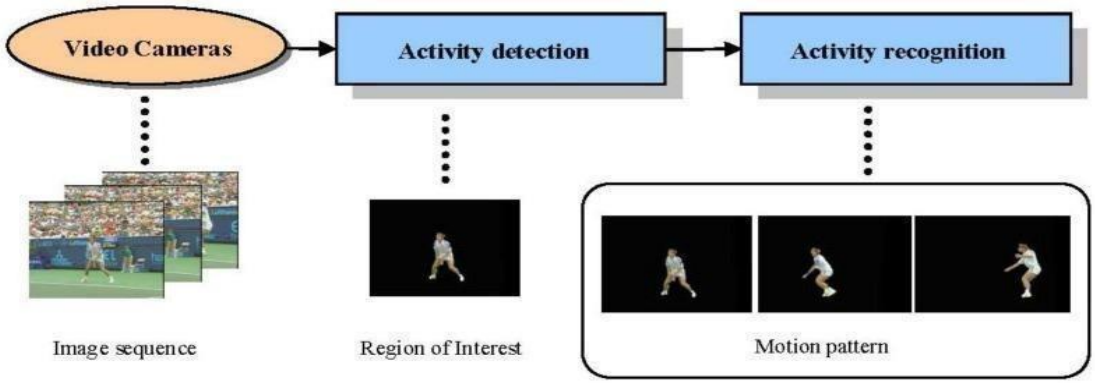


Figure 1.1: Hand Gesture Recognition Flow Chart

1.3 OBJECTIVES

The objectives of the project are:

1. Study and apply the needed tools namely:
    a. Image downloaded from a computer or locally saved.
    b. Flask Deployed Server and Python 3.6.8 Community
    c. Algorithms for computer vision and machine learning.
2. Develop a front-end website to upload images to process.
3. Test the computer application and website running
4. Document the result of the project.

1.4 SCOPE

The scope of our project is to develop a real-time activity recognition system that ultimately controls the image with a jpg extension and the camera samples of the real-time webcam method. During the project, four gestures were chosen to represent four navigational commands: sitting, standing, bending, and sleeping. A simple computer vision application was written for the detection and recognition of the four gestures and their translation into the corresponding commands for the actions and tracking. There after the program was tested on a webcam with the actual movement of the person in real-time and the results were observed.

CHAPTER 2

LITERATURE SURVEY

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time. Its goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

Project Title: Deep CNN On Multichannel Time Series For Human Activity Recognition
Author Name : Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, Shonali
Year of Publish : 2020

This paper focuses on human activity recognition (HAR) problem, in which inputs are multichannel time series signals acquired from a set of body worn inertial sensors and outputs are predefined human activities. In this problem, extracting effective features for identifying activities is a criticalbut challenging task. Most existing work relies on heuristic hand-crafted feature design and shallow feature learning architectures, which cannot find those distinguishing features to accurately classify different activities. In this paper, we propose a systematic feature learning method for HAR problem. This method adopts a deep convolutional neural networks (CNN) to automate feature learning from the raw inputs in a systematic way. Through the deep architecture, the learned features are deemed as the higher level abstract representation of low level raw time series signals. By leveraging the labelled information via supervised learning, the learned features are endowed with more discriminative and classification are mutually enhanced. All these unique advantages of the CNN make it outperform other HAR algorithms, as verified in the experiments on the Opportunity Activity Recognition Challenge and other benchmark datasets.

Project Title : Deep Learning for Sensor-based Activity Recognition

Author Name : Jindong Wanga,b, Yiqiang Chena,b,_, Shuji Haoc, Xiaohui Penga,b, Lisha

Year of Publish : 2021

Sensor-based activity recognition seeks the profound high-level knowledge about human activities from multitudes of low-level sensor readings. Conventional pattern recognition approaches have made tremendous progress in the past years. However, those methods often heavily rely on heuristic hand-crafted feature extraction, which could hinder their generalization performance. Additionally, existing methods are undermined for unsupervised and incremental learning tasks. Recently, the recent advancement of deep learning makes it possible to perform automatic high-level feature extraction thus achieves promising performance in many areas. Since then, deep learning based methods have been widely adopted for the sensor-based activity recognition tasks. This paper surveys the recent advance of deep learning based sensor-based activity recognition. We summarize existing literature from three aspects: sensor modality, deep model, and application. We also present detailed insights on existing work and propose grand challenges for future research.

Project Title : Human Activity Recognition Using LSTM-RNN Deep Neural Network

Author Name : Schalk Wilhelm Pienaar1, Reza Malekian

Year of Publish : 2019

Using raw sensor data to model and train networks for Human Activity Recognition can be used in many different applications, from fitness tracking to safety monitoring applications. These models can be easily extended to be trained with different data sources for increased accuracies or an extension of classifications for different prediction classes. This paper goes into the discussion on the available dataset provided by WISDM and the unique features of each class for the different axes. Furthermore, the design of a Long Short Term Memory (LSTM) architecture model is outlined for the application of human activity recognition. An accuracy of above 94% and a loss of less than 30% has been reached in the first 500 epochs of training.

Project Title : A Survey of Multiple Classifier Systems as Hybrid Systems

Author Name : Michał Wo´zniaka,_, Manuel Gra˜nab, Emilio Corchadoc

Year of Publish : 2021

A current focus of intense research in pattern classification is the combination of several classifier systems, which can be built following either the same or different models and/or datasets building approaches. These systems perform information fusion of classification decisions at different levels overcoming limitations of traditional approaches based on single classifiers. This paper presents an up-to-date survey on multiple classifier system (MCS) from the point of view of Hybrid Intelligent Systems. The article discusses major issues, such as diversity and decision fusion methods, providing a vision of the spectrum of applications that are currently being developed.

Project Title : CNN Based Approach for Activity Recognition using a WristWorn Accelerometer

Author Name: Madhuri Panwar1, S. Ram Dyuthi1, K. Chandra Prakash1, Dwaipayan Biswas2, Amit Acharyya1, Koushik Maharatna3, Arvind Gautam1, Ganesh R. Naik4

Year of Publish : 2019

In recent years, significant advancements have taken place in human activity recognition using various machine learning approaches. However, feature engineering have dominated conventional methods involving the difficult process of optimal feature selection. This problem has been mitigated by using a novel methodology based on deep learning framework which automatically extracts the useful features and reduces the computational cost. As a proof of concept, we have attempted to design a generalized model for recognition of three fundamental movements of the human forearm performed in daily life where data is collected from four different subjects using a single wrist worn accelerometer sensor. The validation of the proposed model is done with different pre-processing and noisy data condition which is evaluated using three possible methods. The results show that our proposed methodology achieves an average recognition rate of 99.8% as opposed to conventional methods based on K means clustering, linear discriminant analysis and support vector machine.

Project Title : LSTM Networks for Mobile Human ActivityRecognition

Author Name : Yuwen Chen*, Kunhua Zhong, Ju Zhang, Qilong Sun and Xueliang Zhao

Year of Publish : 2020

A lot of real-life mobile sensing applications are becoming available. These applications use mobile sensors embedded in smart phones to recognize human activities in order to get a better understanding of human behavior. In this paper,we propose a LSTM-based feature extraction approach to recognize human activities using tri-axial accelerometers data. The experimental results on the (WISDM) Lab public datasets indicate that our LSTM-based approach is practical and achieves 92.1% accuracy.

Project Title: LSTM-CNN Architecture for Human Activity Recognition

Author Name : Kun Xia, Jianguang Huang, And Hanyu Wang

Year of Publish : 2020

In the past years, traditional pattern recognition methods have made great progress. However, these methods rely heavily on manual feature extraction, which may hinder the generalization model performance. With the increasing popularity and success of deep learning methods, using these techniques to recognize human actions in mobile and wearable computing scenarios has attracted widespread attention. In this paper, a deep neural network that combines convolutional layers with long short-term memory (LSTM) was proposed. This model could extract activity features automatically and classify them with a few model parameters. LSTM is a variant of the recurrent neural network (RNN), which is more suitable for processing temporal sequences. In the proposed architecture, the raw data collected by mobile sensors was fed into a two-layer LSTM followed by convolutional layers. In addition, a global average pooling layer (GAP) was applied to replace the fully connected layer after convolution for reducing model parameters. Moreover, a batch normalization layer (BN) was added after the GAP layer to speed up the convergence, and obvious results were achieved. The model performance was evaluated on three public datasets (UCI, WISDM, and OPPORTUNITY). Finally, the overall accuracy of the model in the UCI-HAR dataset is 95.78%, in the WISDM dataset is 95.85%, and in the opportunity dataset is 92.63%. The results show that the proposed model has higher robustness and better activity detection capability than some of the reported results. It can not only adaptively extract activity features, but also has fewer parameters and higher accuracy

Project Title : Sequential Human Activity Recognition Based on Deep Convolutional Network and Extreme Learning Machine UsingWearable Sensors

Author Name : Jian Sun ,1,2 Yongling Fu,1 Shengguang Li ,2 Jie He ,3 Cheng Xu,3

Year of Publish 2021

Human activity recognition (HAR) problems have traditionally been solved by using engineered features obtained by heuristic methods. These methods ignore the time information of the streaming sensor data and cannot achieve sequential human activity recognition. With the use of traditional statistical learning methods, results could easily plunge into the local minimum other than the global optimal and also face the problem of low efficiency. Therefore, we propose a hybrid deep framework based on convolution operations, LSTM recurrent units, and ELM classifier; the advantages are as follows: (1) does not require expert knowledge in extracting features; (2) models temporal dynamics of features; and (3) is more suitable to classify the extracted features and shortens the runtime. All of these unique advantages make it superior to other HAR algorithms. We evaluate our framework on OPPORTUNITY dataset which has been used in OPPORTUNITY challenge. Results show that our proposed method outperforms deep nonrecurrent networks by 6%, outperforming the previous reported best result by 8%. When compared with neural network using BP algorithm, testing time reduced by 38%.

CHAPTER 3

SYSTEM ANALYSIS

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about existing problems, define objects and requirements, and evaluate the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis, which gives the target for design and development.

Feasibility Study

All systems are feasible when provided with unlimited resources and infinite time. But unfortunately, this condition does not prevail in the practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees, and untold professional embarrassment can be averted if an ill-conceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case, three key considerations involved in the feasibility analysis are:

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system is well within the budget, and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client.The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcome, as he is the final user of the system.

ANALYSIS

PERFORMANCE ANALYSIS

For the complete functionality of the project work, the project is run with the help of a healthy networking environment. Performance analysis is done to find out whether the proposed system. The process of performance analysis and definition must be conducted in parallel.

TECHNICAL ANALYSIS

A system is beneficial only if it can be turned into an information system that will meet the organization's technical requirements. Simply stated this test of feasibility asks whether the system will work or not when developed & installed, and whether there are implementations. Regarding all these issues in technical analysis, there are several points to focus on:

Changes to bring in the system: All changes should be in a positive direction, there would be an increased level of efficiency and better customer service.

Required skills: Platforms & tools used in this project are widely used. Therefore, a skilled workforce is readily available in the industry.

Acceptability: The structure of the system is kept feasible enough so that there should not be any problem from the user's point of view.

ECONOMIC ANALYSIS

Economic analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we need not have any routers, which are highly economical. Therefore, the system is economically feasible enough.

3.1 EXISTING SYSTEM

In the existing system, mostly we use CCTV cameras for surveillance or monitoring. It was a traditional methodto identify what types of work were done. It's hard to watch our CCTV footage for many hours. In the Existing system, there is a lot of manual work to monitor the works. In our proposed system, we decrease the man work.

PROBLEM DEFINITION:

With the advancements in deep learning, this process has been accelerated with unprecedented improvements, especially with the availability of more data and higher computation power. The feature selection process no longer needs to be task-dependent as deep learning models can extract features automatically and they can be applied to a range of classification tasks. With the provision of deep learning, it makes a logical flow to maximize the use of this state-of-the-art research for concentrated implementation of HAR. Therefore, this work aims to develop a systematic process for a more streamlined feature representation for improved activity recognition through ensemble learning of CNN.

3.2PROPOSED SYSTEM

In this proposed system, we propose the convolution neural network method for action recognition in video. The input video will be captured by using the webcam. The input video is converted into several frames. Then the CNN (Convolution Neural Network) algorithm is used to detect the particular part of the frame. Then the maximum weight values are taken from the feature extraction frames by using the Convolution neural network. Finally, the action will be detected in the videos and then the label (action name) will be identified. Then that output is taken to the firebase and the firebase value is given to the user via Android notification

CHAPTER 4

SYSTEM REQUIREMENTS SPECIFICATION

A System Requirement Specification (SRS) is an organization's understanding of a customer or potentialclient's system requirements and dependencies at a particular point before any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives a brief description of the services that the system should provide and the constraints under which, the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will doit. A two-way insurance policy assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an e-commerce website, and so on) must provide, as well as states any required constraints by which the system must abide. SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. The requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing, and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

GENERAL DESCRIPTION

In this section of the presented thesis, the introduction of a software product under consideration has been presented. It presents the basic characteristics and factors influencing the software product or system model and its requirements.

PRODUCT PERSPECTIVE

In this project or research work, we have proposed a highly robust and efficient mechanism for image detection using a human activity system. The proposed system has been emphasized on developing an efficient scheme that can accomplish activity recognition without introducing any training-related overheads. The proposed system takes into consideration of geometrical shape of the human pose and based on defined thresholds and real-time parametric variation, the segmentation for human position is accomplished. Based on the retrieved specific shapes, certain application-oriented commands have to be generated. The predominant uniqueness of the proposed scheme is that it does not employ any kind of prior training and it is functional in real time without having any databases or training datasets. Unlike traditional approaches to images and datasets-based recognition systems; this approach achieves human activity recognition in real-time and responds correspondingly. This developed mechanism neither introduces any computational complexity nor causes any user interferences to achieve the tracing of human gestures.

USER CHARACTERISTICS

The user should have at least a basic knowledge of Windows and web browsers, such as installing software like Python Pycharm, Python 3.6.8, etc. executing a program, and the ability to follow onscreen instructions. The user will not need any technical expertise to use this program.

4.1 HARDWARE AND SOFTWARE SPECIFICATION

4.1.1 HARDWARE REQUIREMENTS

- Hard Disk: 500GB and Above
- RAM: 4GB and Above
- Processor: I3 and Above
- Webcam

4.1.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 10 (64-bit)
- Software: Python and Anaconda
- Tools: Jupyter Note Book and Spyder IDE

FUNCTIONAL REQUIREMENT

- The image used will be uploaded to the website from where the result will be displayed with the machine's accuracy.
- The software will be able to produce multiple frames and display the image in the RGB color space.
- The software will be able to detect the contours of the detected positions.
- The software, which acts as an intermediate in passing these, processed images to control the image sent.

NON-FUNCTIONAL REQUIREMENT

Usability: The user is facilitated with the control section for the entire process in which they can arrange the position of the hand at the center of ROI under consideration, the variation of palm position and respective command generation, etc. can be effectively facilitated using the user interface. The implementation and calibration of the camera and its resolution can also be done as per quality and preciseness requirements. The frame size, flow rate, and command variation concerning the threshold developed and a color component of hand color can be easily calibrated using certain defined thresholds.

Security and support: The application will be permissible to be used only in the secure network so there is less feasibility of insecurity over the functionality of the application. On the other hand, the system functions in a real-time application scenario, therefore the camera, color, and platform compatibility are a must in this case. In the case of command transfer using certain connected devices or wireless communication, the proper port assignment would also be a predominant factor to be considered.

Maintainability: The installation and operation manual of the project will be provided to the user.

Extensibility: The project work is also open for any future modification and hence the work could be defined as one of the extensible works.

External Interface Requirement

An interface description for short is a specification used for describing a software component's interface. IDLsare commonly used in remote procedure call software. In these issues, the machines on moreover last part of the "link" might be utilizing the Dissimilar Interface Description recommends a bridge between the two diverse systems. These descriptions are classified into the following types:

User Interface: The external or operating user is an individual who is interested in introducing a novel Algorithm for shape-based hand gesture recognition in real-time application scenarios. The user interface would be like an axis presenting the real-time movement of the human hand and its relative position concerning defined centroid or morphological thresholds.

Restoration with Text Removal Software Interface: The Operating System can be any version of Windows, Linux, UNIX, or Mac.

Hardware Interface: In the execution of this project, the hardware interface used is a normal 32/64 bit operating system supported along with better integration with network interface card for better communication with other workstations. For better and more precise outcomes, a high-definition camera with calibrated functioning with. with the correct one, and tweaks are made to their values until the network creates the correct final output each time.

4.2 CONVOLUTIONAL NEURAL NETWORKS (CNN)

Convolutional Neural Networks are a type of neural networks that are generally used for the purpose of image recognition. It utilises the concept of using local receptive fields in order to identify certain aspects or features within the image fed into it which then can help it identify it from a group of different images. A CNN has multiple layers – pooling layer, non-linearity layer, convolutional layer and fully connected layer . The name comes from the mathematical concept of convolution where two functions are combined to form a third function – over here the two functions would be the input combined with a filter of weights which are trained using backpropagation (an algorithm that uses gradient descent for supervised learning of neural networks) which then produce a third function: a feature map. As mentioned earlier, the convolutional layer has parameters (weights) but layers like pooling layer and fully connected layer do not . This greatly improves the performance of convolutional neural networks compared to classic artificial neural networks as this greatly reduces the overall number of parameters within a network. This can allow convolutional neural networks to be used for much larger applications across multiple industries by developers. Structurally, a CNN contains the following three main layers:

Convolutional Layer:

The main responsibility of this layer is to extract useful features such as edges from the input, and as the network starts to grow deeper, more complex features such as shapes and digits are then identified . It consists of two matrices – where one matrix represents a selected portion of the fed input data and the second matrix represents a filter/kernel which is known as a set of learnable parameters. The Kernel matrix slides over the selected portion of the input data matrix to produce a final **feature matrix**. Compared to the original input data matrix, this feature matrix has lesser dimensions and has more clear features than it . Simply put, the convolutional filter (kernel) slides over the input data matrix, producing a dot product at every step and the value from that convolution is then represented in the feature matrix. This feature matrix is then used for all further processing.

Pooling Layer:

The main purpose of this layer is to progressively reduce the size of the feature matrix in order reducing the computation required to process all of the data. Just like features were extracted previously from a much larger matrix, this time those features are ‘pooled’ together. This is done

extracting the dominant features from within sections of the feature matrix and thus the dimensions are reduced .This down sampling of the feature matrix to reduce its complexity in the calculations within further layers can be simply understood by the simple concept of reducing the resolution of an object, for instance, a photo – where the important aspects are preserved for the new image to look like the original image . This reduction in number of parameters can help in terms of removing unwanted noise and controlling overfitting. Maximum Pooling is the most common method used right now, where the maximum of all values within the region being looked at is taken and assigned to the corresponding index of the final matrix produced by the pooling layer.

Fully Connected Layer:

After identifying the complex features within the image and then reducing the size of those features, the identification of the images can finally be done. The image is to be flattened into a single column vector which can then be fed into a feed forward neural network where backpropagation at every step is utilised to ensure that over a series of epochs, the neural network can differentiate between dominant and less dominant features within the data and correctly utilise them for classification . By connecting one or more fully connected layers it is possible for the CNN to establish non-linear dependencies as well. Another important feature within a neural network is the **activation function**. They can be placed within or at the end of a neural network. They allow a neural network to perform non-linear transformation on the linear mapping between the input and the output of the layer it's connected between.In simple words, when applied to an output of 5 different categories, the activation function helps in predicting which category is most probably true and therefore returns a 1 for that category and a 0 for the rest. The activation function has a threshold which, if met, causes it to return a 1, or a 0 otherwise. The three most commonly used activation functions include: sigmoid, Tanh – hyperbolic tangent, and ReLu – Rectified linear units. Rectified Linear Unit has been very commonly used these days. The reason for that is mostly the simple math behind it which makes it easier to apply it to classifiers with a high number of layers. It's simple logic is shown in the following two equations:

$$If\ x < 0: R(x) = 0 \quad (1)$$

$$if\ x \geq 0: R(x) = x \quad (2)$$

This shows that the gradient of the graph of the function at any point will be either 0 or 1. At no point can the gradient saturate, hence this activation function avoids the vanishing gradient problem. Although the one negative aspect is that if a certain neuron never gets activated for any given input, the gradient will always be zero for it and a dead gradient problem may occur.

The limitation with the rectified linear unit activation function is that it should only be used within the hidden layers of a neural network. Hence, for the last output layer of the convolutional neural network a Soft-max layer is used. The soft max function is able to convert the output of the last linear layer of a multi-class neural network into probabilities of which class is most likely to be the correctly predicted one . It produces a probability distribution of the predicted classes – the sum of which should add up to one.When the three main layers – convolution layer, pooling layer and fully connected layer – are combined with an activation function and a soft max layer, the whole process of taking inputs first into the convolutional layer to getting the output at the output layer is called forward propagation. Each individual layer is given the input from the previous layer and processes it as per the activation function and passes each output to the next successive layer. Feed forward networks are important for forward propagation; in order for an output to be generated, it is imperative that no data is fed backwards in the network during the output generation cycle . This takes place at each neuron in a hidden or output layer in two steps: pre-activation phase and activation phase. At the pre-activation phase, the weighted sum of inputs – the linear transformation of the weights with regards to the inputs is available . This aggregated sum is then taken and based on the activation function the neuron decides if this information is to be passed on or not. This calculated sum of weights is then transmitted to the activation phase where non-linearity is added to the network based on the mathematical function of the activation function.The neural networks use a random value of weights when they are first initialised. This might seem counter-intuitive, but in supervised learning it is not where you start off in terms of the weight initialisation of the neurons within each layer. If the network is trained enough times and the classifier is robust and pervasive enough, eventually the correct weights are reached. These weights of each neuron within each layer have to be constantly updated and this is done through backward propagation. Once there is an output from the network in one iteration, the output is compared to the actual value. This is done by the loss function – a generalizable performance metric that enables the user to understand how close the Neural Network's prediction is to the Actual output . The aim of the machine learning program is to always minimise this loss function to zero. The gradient of the loss function is used to optimize the values of the internal weights of the network in order to reduce the value of the loss function. A positive gradient would point at reducing the size of the weights as at that randomly initialised weight if you increase the weight size, the total error would increase. A negative gradient would point at increasing the weights as at this current randomly initialised weight, the total error is decreasing.

The aim is to get to a gradient of 0 after this process of increasing or reducing the size of the weights. This process of finding a minimum is called gradient descent. Appropriate step sizes need to be taken to ensure that the classifier is able to find a global minimum and not a local minimum.

Extensive training and the right epoch sizes can ensure that.As each error is calculated, and each error's gradient is calculated, this is then sent back to the start of the layers. This backpropagation hence allows the weights to be updated with an appropriate value to ensure the performance of the classifier can be increased. As each weight is updated, it is important to ensure that each change to the weight is minute, as a big change in the weights can cause the behaviour of the neural network to be chaotic . Smaller changes can help in identifying the correct value faster as the trajectory of the neural network can be easily projected and localised. Especially since these neural networks usually have lots of non-linearity, and the current gradient that the network would have is localised at the certain point it is being calculated on, bigger changes to the updated weights would make it very difficult to observe and analyse the results of those updates. The methods using which these weights are updated are called optimizers. A commonly used optimizer is the Adam optimizer – which is a variant of stochastic gradient descent. Adam is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. The algorithm works particularly well because it finds individual learning rates for each parameter through the power of adaptive learning rate methods. Adam can be considered a combination of RMSprop and Stochastic Gradient Descent with momentum . The central idea of RMSprop is keep the moving average of the squared gradients for each weight. And then we divide the gradient by square root the mean square. In Stochastic Gradient Descent, using random probability, the algorithm tries to minimize a certain cost function by moving in the direction of the steepest descent as defined by the negative of the gradient.

The whole process of forward and backward propagation is then repeated a number of times to get to a point where the neural network is adequately trained on the training data to be able to make predictions on similar information. It takes several iterations for the neural network to reach the right parameters to learn . Each iteration is called an epoch. The number of epochs depends on a number of factors: quality of data, learning rate of the algorithm, complexity of the layers, optimization method used, and even the random initialization of the weights.

4.3 LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK

LSTM RNN are a set of Neural Networks whose functions and way of working is inspired by the human brain. A Recurrent Neural Network differentiates from a normal Artificial Neural Network in the way that it specifically makes use of sequential information. If two images of a cat and dog are fed into an artificial neural network, the ANN will recognize a few patterns and predict an image to be of a dog or a cat. At no point did the ANN's prediction depend on previous images.

Each prediction of a dog or a cat would be based on the image itself and not the data before it. However, there can be multiple situations where the previous data would be important in terms of making a prediction on the current data. For example, predicting the financial market value of a company's stocks, predicting the next word in a sentence etc. In such situations, a neural network – like the recurrent neural network – is needed where it can have 'memory' to store sequential information and utilize that when making a certain prediction. In traditional neural networks, it is generally assumed that the input and outputs are independent of each other. But this can become detrimental when, for example, predicting a word in a sentence as mentioned before. The word 'recurrent' is used for this neural network because it performs the same task on every element of a sequence of data, with each output being dependent on the previous data. As can be seen from the following photo where x_t represents the input, A represents the activation function of the hidden layer, and h_t represents the value of the current state.

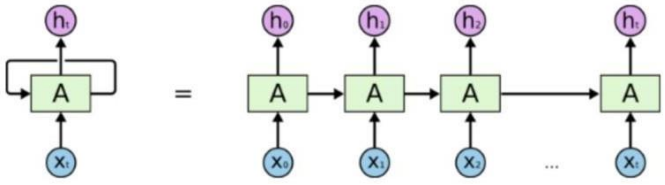


Figure 4.1. How an RNN utilizes its 'memory'

There are many different types of RNNs: one to one, one to many, many to one, many to many and bidirectional many to many. Many to one is the kind of RNN where it takes a sequence of multiple data inputs and provides one output of a fixed size. The following photo describes the architecture of a many to one RNN where red arrows represent the inputs and the green and blue arrows represents the outputs of the hidden layers:

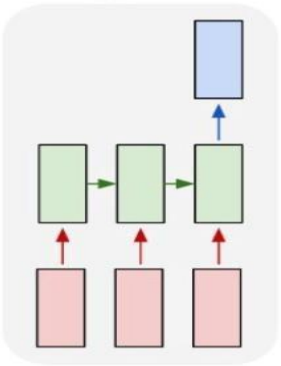


Figure 4.2. Many to one RNN

The weights and biases applied to the hidden layers of an RNN are always the same in order for it to memorize the information processed through them.

In Figure 4.3 below, the formula to find the current state would be the following equation:

$$h_t = f(h_{t-1}, x_t) \qquad (3)$$

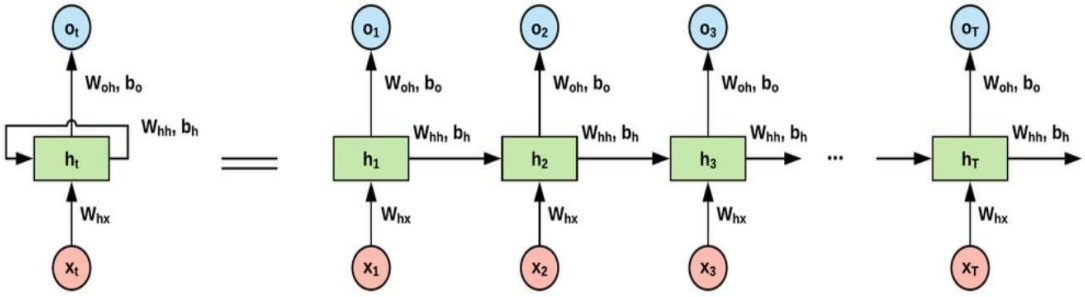


Figure 4.3. Information retention of an RNN

O_t represents the output state, h_t would be the current time stamp, h_{t-1} would the previous time stamp and x_t is passed as the input to the next stage. The tanh activation function is applied within each candidate cell to determine the internal state value and that is then used to update the hidden state's value. This is described in the following equation:

$$h_t = \tanh\left(W_{hh}h_{t-1} + W_{xh}x_t\right) \qquad (4)$$

In equation 4 W, is the weight, with W_{hh} representing the weights from the previous hidden layer, and W_{xh} being the weights at the current input state [24]. The tanh activation function is used because it implements a non-linearity where the output values are within the range -1 to 1. Its non-linear nature allows for multiple LSTM layers to be stacked on top of each other. The output at the final stage is described best through the following equation:

$$y_t = W_{hy} h_t \qquad (5)$$

In equation (5) y_t represents the output state and W_{hy} represents the weight at the output state.

The LSTM RNN utilizes backpropagation to improve the value of the weights and biases assigned at each layer during the training process to reach a more accurate output state. The loss function helps to calculate the difference between the current output state predicted and the real output state. Mean squared error is a common loss function used. The gradient of the loss function is used to optimize the values of the internal weights to try and reduce the value of the loss function down to zero.

As the weights are the same for all hidden layers in a recurrent neural network, the gradient calculated for each time step can be combined together. The weights of the recurrent neuron and the output dense layer can then be updated together. There are two kinds of problems one can face during backpropagation: vanishing gradient and exploding gradient. In simple words, **vanishing gradient** is a problem that occurs when the contribution from earlier steps becomes insignificant in the gradient descent step. This can happen when the error between the predicted output and the real output has a partial derivative with respect to the weight that is less than 1. This value is then multiplied with the learning rate – which is already a very insignificant value – causing this final value to be even smaller. Then as the weights are updated using this value, there won't be any significant changes in the weights and thus the vanishing gradient problem is faced. It causes the RNN to forget the long-term dependencies as those value isn't fed to the next iterations. Similarly, exploding gradient is the problem where if the partial derivative of the error with respect to the weights is extremely high, a stupidly high value is assigned to the weights and thus the gradient explodes.

These problems are solved by adding LSTM to RNNs: Long Short-Term Memory. Within the LSTM cell states, long term dependencies and relations are encoded in state vectors and it is this cell state derivative that can prevent the LSTM gradients from vanishing . LSTMs consist of three steps: Forget gate, Input gate, Output gate. This is shown in the image below:

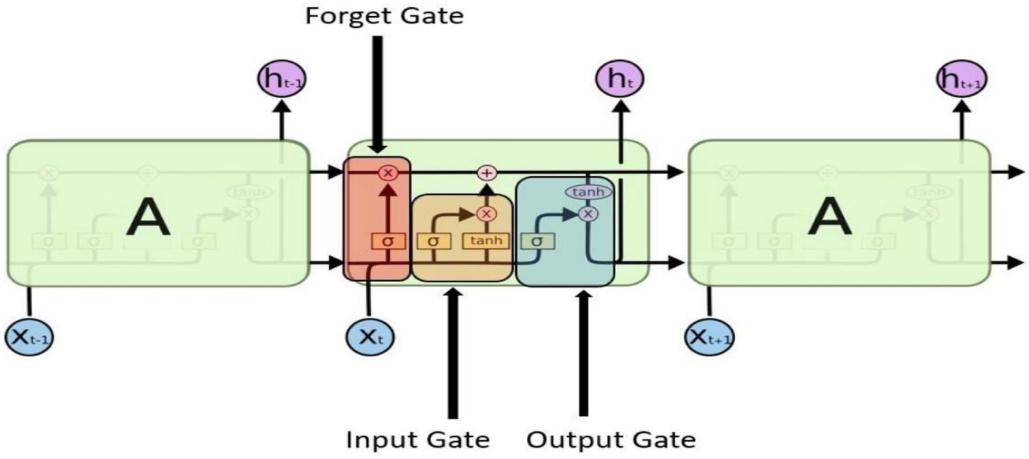


Figure 4.4. Structure of a LSTM cell

- **Forget Gate:** Essentially, this gate lets the LSTM cell know how much it is supposed to remember and how much it is supposed to forget from the information fed to it . At every time stamp this gate figures out which information is to be left out from the cell.This is done using a **sigmoid function**. It looks at the previous data input and the previous state, and then outputs a value between 0 and 1. 0 meaning remove this information and 1 meaning keep this information. The forget gate's output equation is given as following:

$$f_t = \sigma[W_f . (h_{t-1}, x_t)] \qquad (6)$$

- **Input Gate:** The input gate is also called the update gate. It decides how much of this unit is then going to be added to the current state. The sigmoid function decides which values to let through by outputting 0 or 1 . The tanh function assigns a weightage to these values on a scale of -1 to 1 based on their level of importance. The input gate's formula is reflected in the following equation:

$$\tanh\ (W_c . [h_{t-1}, x_t]) \otimes \sigma(W_i . h_{t-1}, x_t) \qquad (7)$$

- **Output Gate:** This gate is responsible for deciding which part of the current cell is going to make it to the output. The sigmoid function is used to decide which values are going to go through, tanh

gives the weightage to these values on a scale of -1 to 1 and then the value from the sigmoid and tanh function are multiplied together. The formula for the output of the output gate can be represented in the following equation:

$$o_t = \sigma(W_o.[h_{t-1}, x_t]) \qquad (8)$$

The output from the LSTM cell can then be represented by the following equation:

$$h_t = o_t \otimes \tanh(c_t) \qquad (9)$$

As mentioned previously, the error term within RNNs is a sum of all the gradients calculated within the layers as all weights and biases are the same for all hidden layers. If the derivative of the error with respect to the weight is represented as a sum of sub-gradients through the following equation:

$$\frac{\partial E}{\partial w} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial w} \qquad (10)$$

For the vanishing gradient problem to occur, the sum of these sub-gradients would need to go to zero . To prevent this, the simplest way is to prevent some of these sub-gradients from converging to a value of 0. They must converge to a non-zero value. When an LSTM cell is added to an RNN, the LSTM cell's state vector is added to the error term. If the partial derivatives of the gradient term mentioned in equation (10) is then taken, the partial derivative of the LSTM cell's state vector contains the forget gate's vector of activations which allows the network to better control the gradient value at each iteration using suitable parameter updates of the forget gate. This presence of the forget gate enables the function to decide at each time step which information should not be left out and it can then update the model's parameters accordingly.

In addition to the forget gate's activation function, the shape and form it is present inside the gradient of the error is important as well. After the LSTM cell's state vector is differentiated, it consists of four different additive terms. As these four different terms are now present inside the gradient of the error, at each iteration when these four terms are updated, it is not likely that all four of them will converge to zero. In contrast to a normal RNN's error gradient function which only consists of one term, with LSTM cells added to the RNN, the additive terms make it more unlikely that the function will converge to zero.

In this way, due to the presence of the forget gate's activation function and the additive property of the function, the vanishing gradient problem is avoided. The exploding gradient problem occurs

in the same way as the vanishing gradient. When the sub-gradient terms are combined, if their values are higher than one, the exploding gradient problem occurs. However, the LSTM cell solves this problem in the same manner as it solves the vanishing gradient problem.

There are a few more features that are necessary to implement the neural network:

Dropout: At every layer of LSTM, dropout is added to each layer. In simple terms, dropout is leaving out a few units with a set of units during the training phase. Leaving out a few units or neurons means these particular units or neurons are not looked at during forward or backward propagation. These nodes are dropped out with the probability 1-p or kept with the probability p in order for the overall network to be reduced . The incoming and outgoing edges to a dropped-out node are also removed .Dropout is necessary in order to prevent over-fitting of the network. Overfitting occurs if a model trains 'too well' on the training data. As most of the parameters are occupied by the fully connected layer, the neurons within it develop a co-dependency amongst each other during the training phase. This hampers the individual power of each neuron leading to over-fitting of the training data. Overfitting is detrimental to the model as it learns the pattern and the noise in the training data to such an extent that performance of the model on a new dataset would be negatively impacted.

During the training phase, on every iteration, at each hidden layer a random fraction, p, of nodes and their activation functions will be ignored. In the testing phase, all activations will be used but will be reduced a factor of p to account for the missing activations during the training. The way the inter-dependencies of the neurons are removed can be seen in the following figure:

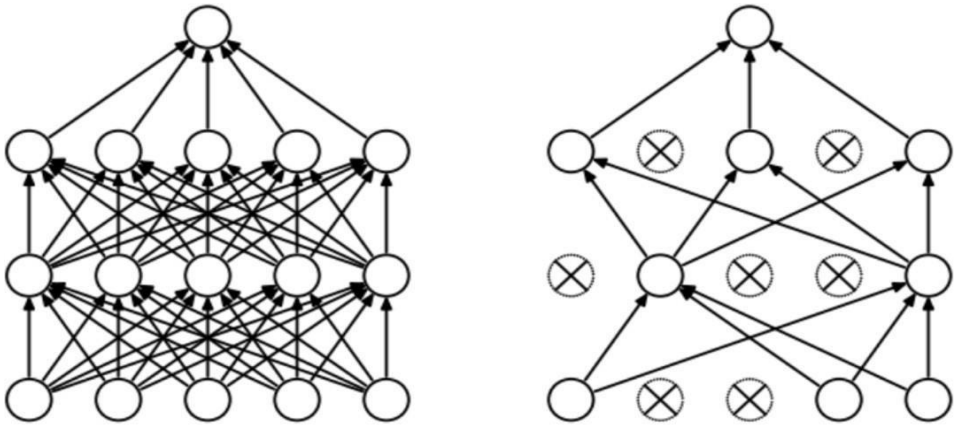


Figure 4.5. Reduction of inter-dependencies of neurons before and after dropout

CHAPTER 5

5.1 Design and Implementation Constraints

5.11 Constraints in Analysis

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

5.1.2 Constraints in Design

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

5.1.3 Constraints in Implementation

Hierarchical structuring of relations may result in more classes and a more complicated structureto implement. Therefore, it is advisable to transform the hierarchical relation structure to a simplerstructure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object-oriented methods.

5.2 Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and access to the web service;
- Server Tier: The server side contains the main parts of the functionality of the
- Proposed architecture. The components at this tier are the following.

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

5.2.1 Safety Requirements

1 The software may be safety-critical. If so, there are issues associated with its integrity level

2 The software may not be safety-critical although it forms part of a safety-critical system. For example, the software may simply log transactions.

3 If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.

4 There is little point in producing 'perfect' code in some languages if hardware and system soft- ware (in the widest sense) are not reliable.

5 If a computer system is to run software of a high integrity level then that system should notat the same time accommodate software of a lower integrity level.

6 Systems with different requirements for safety levels must be separated.

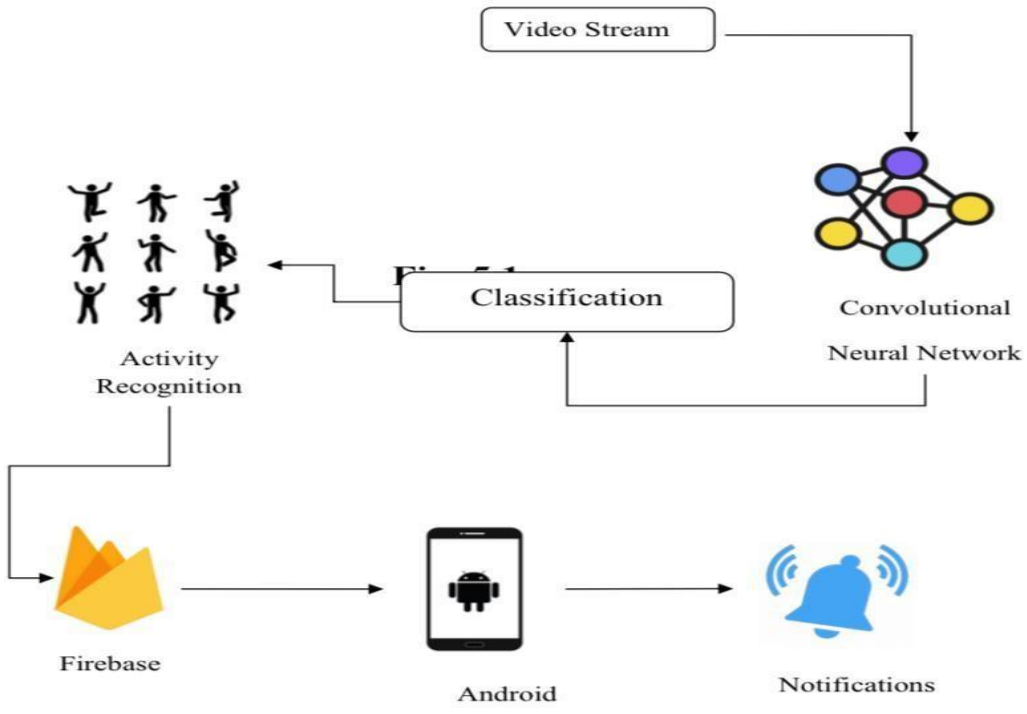
7 Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

CHAPTER 6

SYSTEM DESIGN

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the development of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in Software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system design. Design is the perfect way to accurately translate a customer's requirement into the finished software product. The design creates a representation or model and provides details about software data structure, architecture, interfaces, and components that are necessary to implement a system. The logical system design is arrived at because of systems analysis is converted into physical system design.

6.1 Architecture Diagram:



6.1 System Architecture

6.2 SYSTEM DEVELOPMENT METHODOLOGY

The system development method is a process through which a product will get completed or a product gets rid of any problem. The software development process is described as several phases, procedures, and steps that give the complete software. It follows a series of steps, which are used for product progress. The development method followed in this project is the waterfall model.

Model Phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing, and maintenance.

Requirement Analysis: This phase is concerned with the collection of requirements of the system. This process involves generating documents and requirement review.

System Design: Keeping the requirements in mind the system specifications are translated into a software representation. In this phase, the designer emphasizes algorithm, data structure, software architecture ,etc.

Coding: In this phase, the programmer starts his coding to give a full sketch of the product. In other words, system specifications are only converted into machine-readable computer code.

Implementation: The implementation phase involves the actual coding or programming of the software The output of this phase is typically the library, executables, user manuals, and additional software documentation

Testing: In this phase, all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

Maintenance: The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate changes in the external environment, correct errors and oversights previously undetected in the testing phase,

Advantages of the Waterfall Model

- Clear project objectives.
- Stable project requirements.
- Progress of the system is measurable.
- Strict sign-off requirements.
- Helps you to be perfect.
- The logic of software development is clearly understood.
- Production of a formal specification.
- Better resource allocation.
- Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.
- Less human resources are required as once one phase is finished those people can start working on the next phase

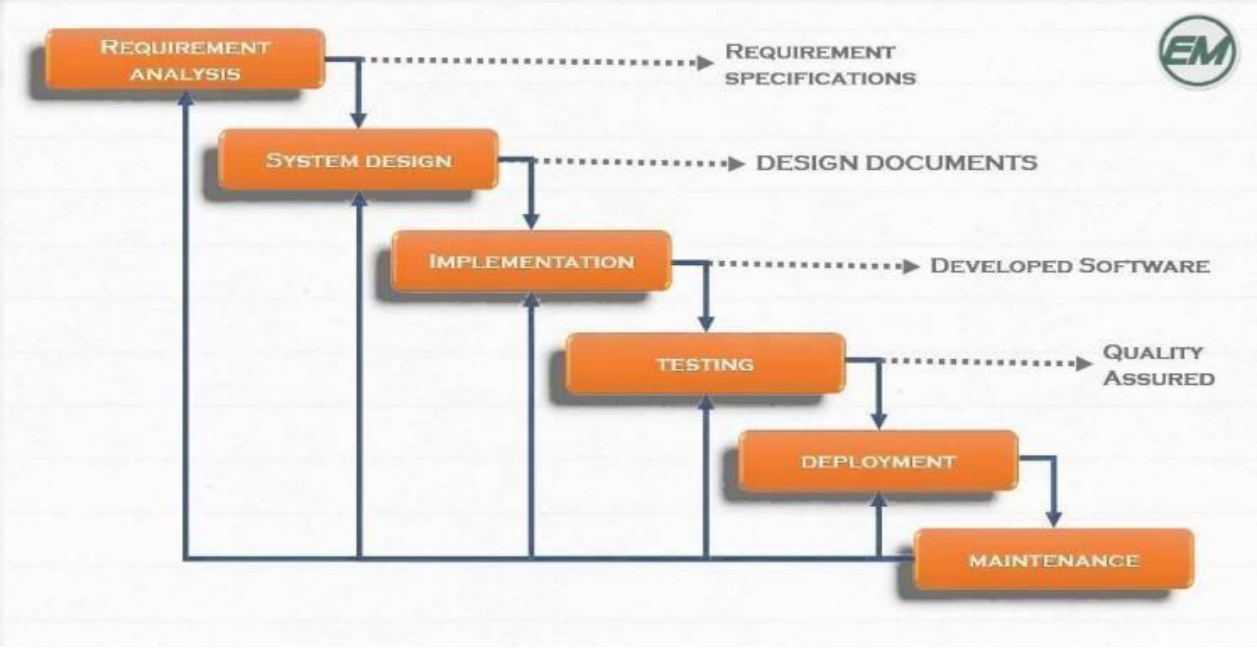


Fig 6.2 Waterfall Model

6.3 Design Using UML

Designing a UML diagram specifies, how the process within the system communicates along with how the objects within the process collaborate using both static as well as dynamic UML diagrams since in this ever-changing world of object-oriented application development, it has been getting harder and harder to develop and manage high-quality applications in a reasonable amount of time. Because of this challenge and the need for a universal object modeling language everyone could use, the Unified Modelling Language (UML) is the Information Industries version of the blueprint. It is a method for describing the system architecture in detail. Easier to build or maintain a system, and to ensure that the system will hold up to the requirement changes.

6.4 Data Flow Diagram

The DFD is also called a bubble chart. It is a simple graphical formalism that can be used to represent system in terms of the input data to the system, various processing carried out on these data, and the output data generated by the system.

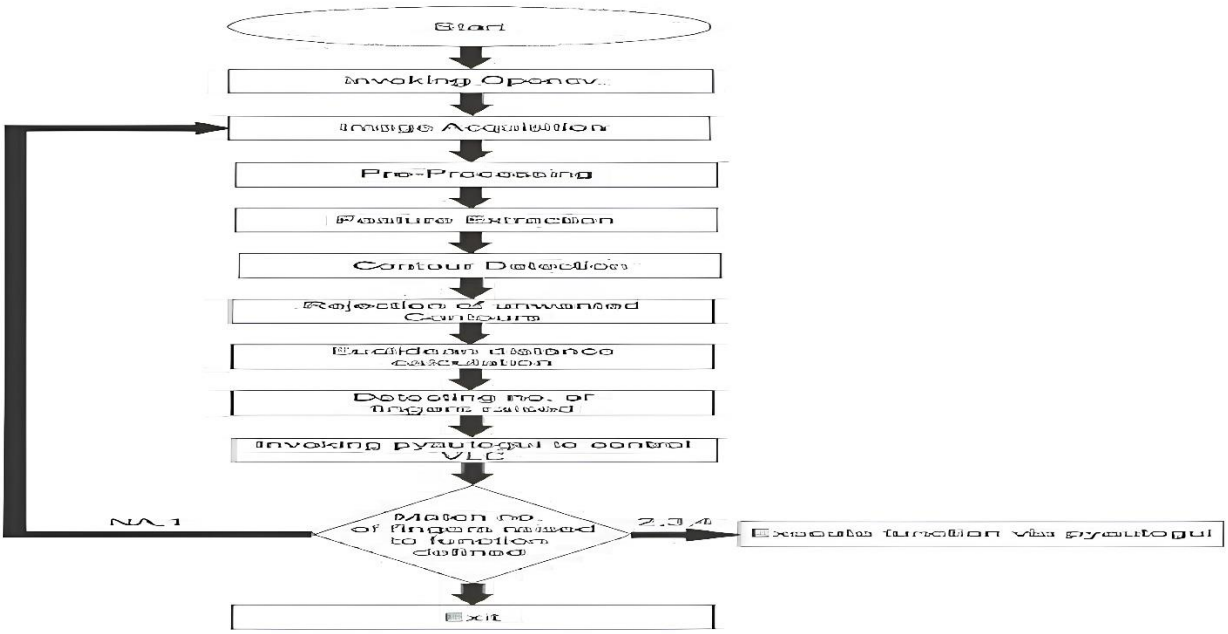


Fig 6.3 Data Flow Model

The data flow diagram essentially shows how the data control flows from one model to another. Unless the input filenames are correctly given the program cannot proceed to the next module. Once the user gives, the correct input filenames parsing is done individually for each file. The required information is taken in parsing and an adjacency matrix is generated for that. From the adjacency matrix, a lookup tables generated giving paths for blocks. In addition, the final sequence is computed with the lookup table and the final required code is generated in an output file. In the case of multiple file inputs, the code for each is generated and combined.

6.5 Component diagram

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

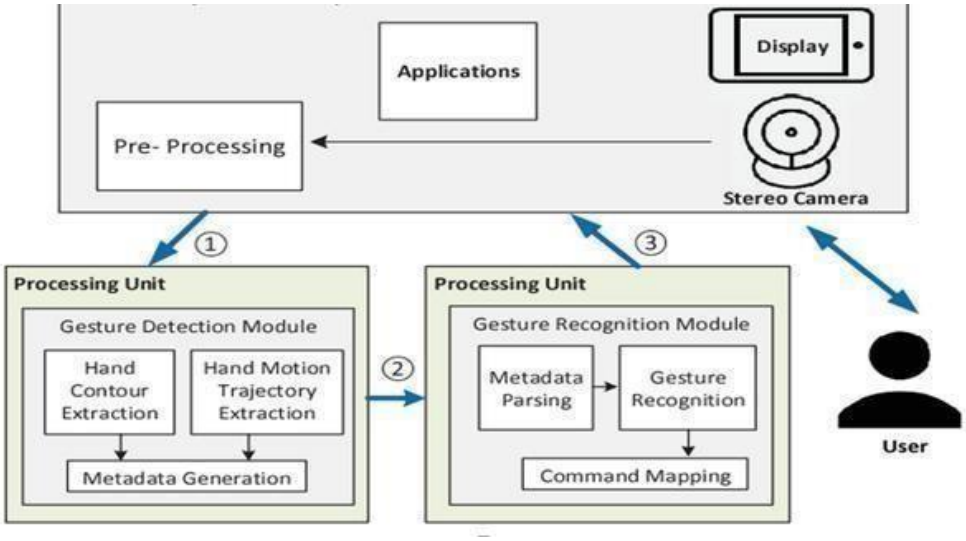


Fig 6.4 Component Diagram

The component diagram for the decentralized system ideally consists of different modules that are represented together via a common module for the user. The user is required to have the input files in the current folder where the application is being used.It is interesting to note that all the sequences of activities that are taking place are via this module itself, i.e.the parsing and the process of computing the final sequence. The parsing redirects across the other modules until the final code is generated.

6.6 Use Case Diagram

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities, which interact with the system, are its actors. A set of use cases describes the complete functionality of the system at a particular level of detail and the use case diagram can graphically denote it. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. In software and systems engineering, a use case is a list of steps, typically defining interactions between a role (known in Unified Modelling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be captured in Systems Modelling Language (SysML) or as contractual statements. The Sequence ofactivities that are carried out is the same as in the other diagrams. The use case for this module indicates the user's interaction with the system as a whole rather than individual modules. All the encryption mechanisms are carried out via the login page that redirects the user to the particular functionality that he or she wishes to implement.

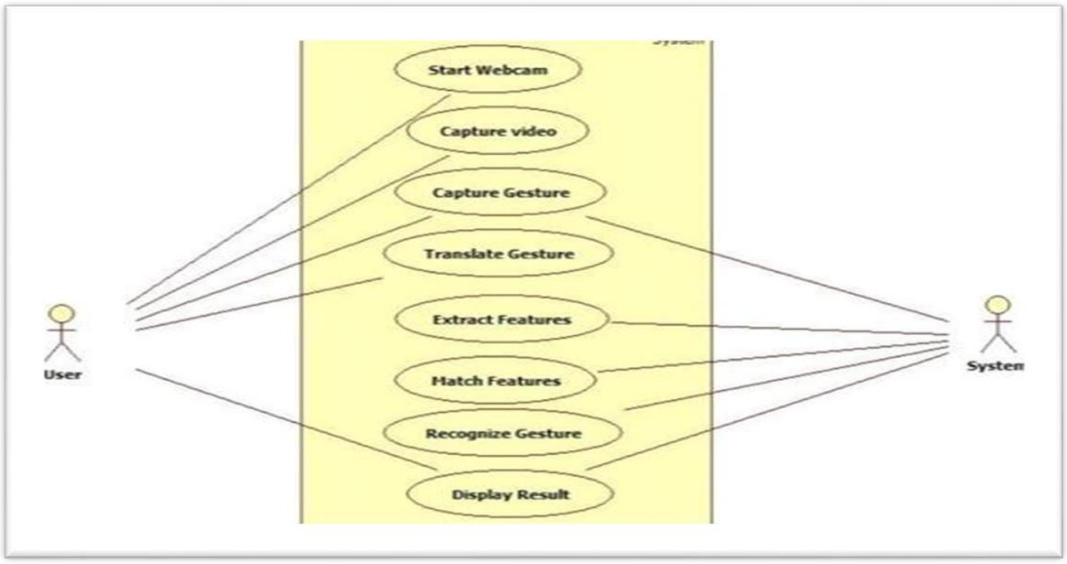


Fig 6.5 Use Case Diagram

6.7 Activity Diagram

An activity diagram shows the sequence of steps that make up a complex process. An activity is shown as a round box containing the name of the operation. An outgoing solid arrow attached to the end of the activity symbol indicates a transition triggered by the completion. Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial state) of the workflow;
- An encircled black circle represents the end (final state).

The basic purposes of activity diagrams are similar to the other four diagrams. It captures the dynamic behavior of the system. The other four diagrams are used to show the message flow from one object to another but the activity diagram is used to show the message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing inactivity diagram is the message part.

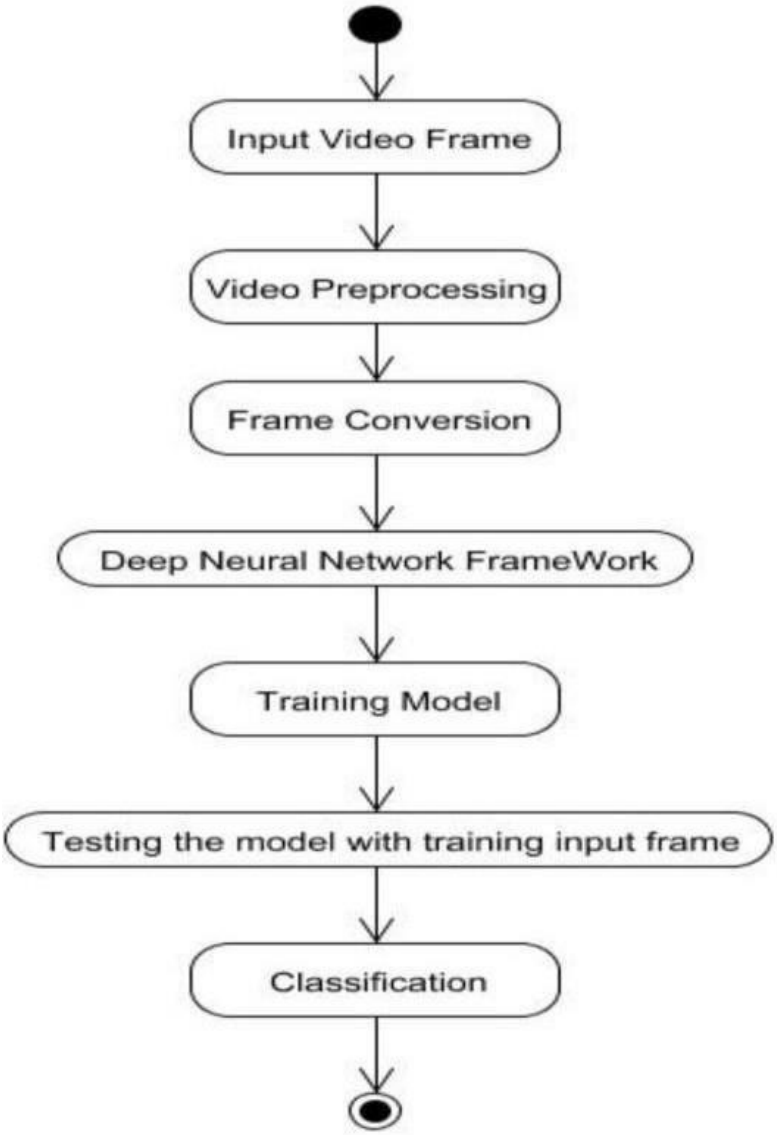


Fig 6.6 Activity Diagram

6.8 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams sometimes called event diagrams, event sceneries, and timing diagrams.

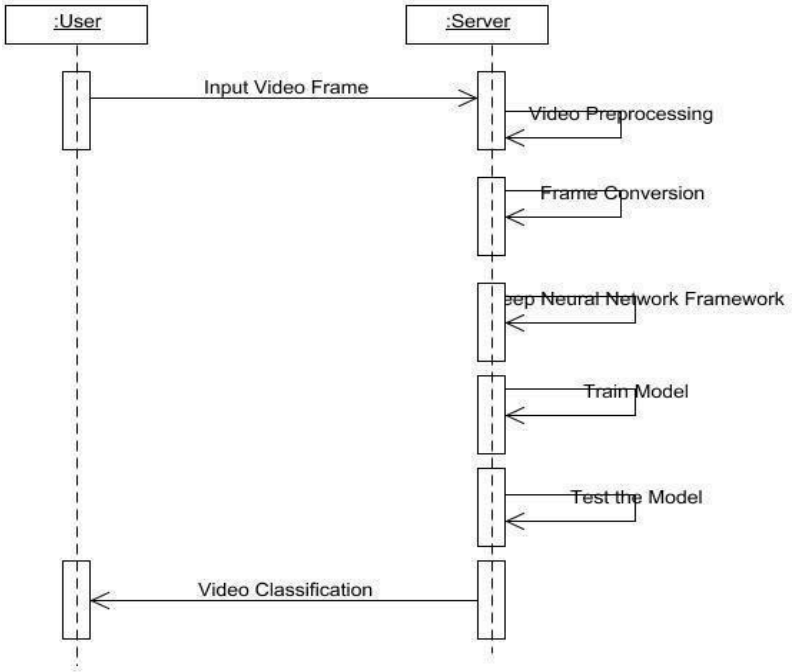


Fig 6.7 Sequence Diagram

CHAPTER 7

PROPOSED SYSTEM

7.1 MODULES

- Video Streaming
- Deep learning algorithm
- Classification

7.2 MODULE EXPLANATION:

VIDEO STREAMING

The input video will be captured by using the webcam. The input video is converted into number of frames. Video sequences present great variability due to huge scale changes, viewpoint variation and camera motion which pose great challenges for both video representations and classification.

DEEP LEARNING ALGORITHM

The convolution neural networks (CNN) have been studied in the video domain for a large variety of classification tasks. The captured video is feed into the convolution neural network and it is highly desired to have an end-to-end learning framework that can establish effective video representations while simultaneously conducting efficient video classification.

CLASSIFICATION

Then the maximum weight values are taken from the feature extraction frames by using the Convolution neural network. According to video labels the entire network is trained for action recognition. Finally the action will be detected in the videos and then label (action name) is identified. After that the detected value passed to the user through firebase via android notification.

CHAPTER 8

RESULTS AND DISCUSSION

8.1 RESULTS

The outcome of the project is the successful development of a sophisticated Human Activity Recognition system through Ensemble Learning of Multiple Convolutional Neural Networks. This system integrates ensemble learning techniques with convolutional neural networks to accurately identify and classify human activities based on input sensor data. Through extensive experimentation and validation, the system demonstrates robustness and effectiveness in recognizing various activities with high accuracy.

The Human Activity Recognition system incorporates multiple convolutional neural networks (CNNs) trained on diverse datasets and fused through ensemble learning strategies. The ensemble approach enhances the model's performance by leveraging the strengths of individual CNNs and mitigating their weaknesses. As a result, the system achieves superior accuracy in activity recognition tasks, providing reliable insights into human behavior.

By utilizing ensemble learning, the system not only improves classification accuracy but also enhances resilience to noisy or incomplete sensor data. This robustness is crucial for real-world deployment, where environmental factors and sensor limitations may affect data quality. Additionally, the system's architecture allows for scalability and adaptability, making it suitable for deployment in various domains requiring human activity recognition, such as healthcare monitoring, sports analytics, and smart environments.

8.1 DISCUSSION

Human Activity Recognition (HAR) through Ensemble Learning of Multiple Convolutional Neural Networks represents a significant advancement in the field of activity recognition. By combining the predictive power of multiple CNN models, the system achieves high accuracy and reliability in classifying human activities based on sensor data.

Key points for discussion include:

Ensemble learning enhances model performance: By aggregating predictions from multiple CNNs, the system mitigates individual model biases and improves overall classification accuracy. This approach ensures robustness and generalization capability across diverse activity types and environmental conditions.

Real-world applications: The HAR system has wide-ranging applications in various industries, including healthcare, fitness tracking, security surveillance, and human-computer interaction. Its ability to accurately recognize and interpret human activities opens up opportunities for enhancing user experiences and improving operational efficiency.

Challenges and future directions: Despite its success, the HAR system may face challenges such as data privacy concerns, scalability issues, and domain adaptation requirements when deployed in real-world scenarios. Future research directions may focus on addressing these challenges, exploring novel ensemble learning techniques, and integrating additional sensor modalities for comprehensive activity recognition. In summary, the development of a Human Activity Recognition system through Ensemble Learning of Multiple Convolutional Neural Networks represents a significant milestone in the advancement of activity recognition technology. By harnessing the collective intelligence of multiple CNN models, the system offers accurate, reliable, and scalable solutions for understanding human behavior in various contexts.

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

We presented three CNN based models as well as their ensembles for dataset. It was found that the performance of the ensemble model is better than that of individual models. One of the ensemble model performed better than the methods in the literature. In the dataset we used, we see a class imbalance such that we have 38% samples for walking class but hardly 5% for sitting and standing. In future, the results might be improved even more, if we can remove the class imbalance from dataset.

9.2 FUTURE WORK

Moreover, currently an ensemble of average of the three models is created but for further exploration, a future direction can be performing weighted ensemble learning such that the best performing model has the most effect in the ensemble. Furthermore, we can explore the area of ensemble learning for a hybrid model, that is, ensemble learning of CNN and RNN models.

REFERENCES

[1] T. Pl¨otz, N. Y. Hammerla, and P. L. Olivier, “Feature learning for activity recognition in ubiquitous computing,” in 22nd International Joint Conference on Artificial Intelligence, 2011.

[2] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, “LSTM networks for mobile human activity recognition,” in International Conference on Artificial Intelligence: Technologies and Applications (ICAITA 2016), 2016.

[3] L. K. Hansen and P. Salamon, “Neural network ensembles,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no.10, pp. 993–1001, 1990.

[4] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishna swamy,“Deep convolutional neural networks on multichannel time series for human activity recognition,” in 24th International Joint Conference on Artificial Intelligence, 2015.

[5] M. Panwar, S. R. Dyuthi, K. C. Prakash, D. Biswas, A. Acharyya
K. Maharatna, A. Gautam, and G. R. Naik, “CNN based approach for activity recognition using a wrist-worn accelerometer,” in 2017 39th Annual International Conference of the IEEE Engineering in Medicineand Biology Society (EMBC). IEEE, 2017, pp. 2438–2441.

[6] S. W. Pienaar and R. Malekian, “Human activity recognition using LSTM-RNN deep neural network architecture,” in 2019 IEEE 2nd Wireless Africa Conference (WAC). IEEE, 2019, pp. 15.

[7] J.Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor based activity recognition: A survey,” Pattern Recognition Letters, vol.119, pp. 3–11, 2019.

[8] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, “A robusthuman activity recognition system using smartphone sensors and deep learning,” Future Generation Computer Systems, vol. 81, pp. 307–313, 2018.

[9] J. Sun, Y. Fu, S. Li, J. He, C. Xu, and L. Tan, "Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors," Journal of Sensors, vol. 2018, pp.1–10, 09 2018.

[10] K. Xia, J. Huang, and H. Wang, "LSTM-CNN Architecture for Human Activity Recognition," IEEE Access, vol. 8, pp. 56 855–56 866, 2020.

[11] D. Ravi, C. Wong, B. Lo, and G. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-powerdevices," in 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), 2016, pp. 71–76.

[12] M.Wo´zniak, M. Gra˜na, and E. Corchado, "A survey of multiple classifiersystems as hybrid systems," Information Fusion, vol. 16, pp. 3–17, 2014.

[13] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition usingcellphone accelerometers," ACM SigKDD Explorations Newsletter,
vol. 12, no. 2, pp. 74–82, 2011.

[14] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller,"Striving for simplicity: The all convolutional net," arXiv preprint arXiv:1412.6806, 2014.

[15] M. Lin, Q. Chen, and S. Yan, "Network in Network," arXiv preprintarXiv:1312.4400, 2013.

[16] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphonesensors using deep learning neural networks," Expert Systems
with Applications, vol. 59, pp. 235 – 244, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416302056

APPENDICES

AI SDG GOALS

Human activity recognition (HAR) through ensemble learning of multiple convolutional neural networks (CNNs) can contribute to several Sustainable Development Goals (SDGs). Here's how:

Goal 3: Good Health and Well-being: Human recognition systems can be used in healthcare for patient identification, monitoring, and personalized treatment. For example, facial recognition can help in patient identification in hospitals, ensuring accurate medical records and treatment.

Goal 4: Quality Education: Human recognition technology can be integrated into educational tools and systems to personalize learning experiences for students. For instance, gesture recognition can enable interactive learning applications, making education more engaging and effective.

Goal 5: Gender Equality: Ensuring fairness and inclusivity in human recognition systems is essential to promote gender equality. Developers need to be aware of potential biases in their algorithms and datasets to prevent discrimination based on gender or other characteristics.

Goal 8: Decent Work and Economic Growth: Developing human recognition technologies creates opportunities for employment and economic growth, particularly in the technology sector. It also improves productivity and efficiency in various industries by automating tasks such as attendance tracking, security, and customer service.

Goal 9: Industry, Innovation, and Infrastructure: Human recognition technology falls under the broader category of artificial intelligence and contributes to innovation in the field of computer vision and pattern recognition. It also drives advancements in infrastructure, such as surveillance systems and smart cities.

Goal 16: Peace, Justice, and Strong Institutions: Ethical development and deployment of human recognition systems are crucial for ensuring privacy, security, and human rights. Regulations and policies need to be in place to govern the use of these technologies and prevent misuse or abuse, thus contributing to peace, justice, and strong institutions.

Overall, the application of ensemble learning of multiple CNNs for human activity recognition aligns with multiple SDGs by leveraging technological innovations to address societal challenges and promote sustainable development.

A2 SOURCE CODE

VIDEO CLASSIFACTION

```
# python video_classification.py --model resnet-34_kinetics.onnx --classes

action_recognition_kinetics.txt --input example_activities.mp4

# import the necessary packages

import numpy as np

import argparse

import imutils

import sys

import cv2

import datetime

import time

#firebase = firebase.FirebaseApplication('https://human-activity-45558-default-rtdb.firebaseio.com/',

None)

ap = argparse.ArgumentParser()

ap.add_argument("-m", "--model", required=True,

        help="path to trained human activity recognition model")

ap.add_argument("-c", "--classes", required=True,

        help="path to class labels file")

ap.add_argument("-i", "--input", type=str, default="",

        help="optional path to video file")

args = vars(ap.parse_args())

CLASSES = open(args["classes"]).read().strip().split("\n")

SAMPLE_DURATION = 16

SAMPLE_SIZE = 112

print("[INFO] loading human activity recognition model...")
```

```
net = cv2.dnn.readNet(args["model"])
print("[INFO] accessing video stream...")
vs = cv2.VideoCapture(args["input"] if args["input"] else 0)
while True:
    frames = []
    for i in range(0, SAMPLE_DURATION):
        # read a frame from the video stream
        (grabbed, frame) = vs.read()
        if not grabbed:
            print("[INFO] no frame read from stream - exiting")
            sys.exit(0)
        frame = imutils.resize(frame, width=400)
        frames.append(frame)
    blob = cv2.dnn.blobFromImages(frames, 1.0,
        (SAMPLE_SIZE, SAMPLE_SIZE), (114.7748, 107.7354, 99.4750),
        swapRB=True, crop=True)
    blob = np.transpose(blob, (1, 0, 2, 3))
    blob = np.expand_dims(blob, axis=0)
    net.setInput(blob)
    outputs = net.forward()
    label = CLASSES[np.argmax(outputs)]
    lbl=str(CLASSES[np.argmax(outputs)])
    print(str(CLASSES[np.argmax(outputs)]))
    for frame in frames:
        # draw the predicted activity on the frame
        cv2.rectangle(frame, (0, 0), (300, 40), (0, 0, 0), -1)
```

```
cv2.putText(frame, label, (10, 25), cv2.FONT_HERSHEY_SIMPLEX,
        0.8, (255, 255, 255), 2)
datetime1=datetime.datetime.now()


Status=[str(CLASSES[np.argmax(outputs)])]
data1={"Status":[str(CLASSES[np.argmax(outputs)])]}
print(data1)
cv2.imshow("Activity Recognition", frame)
key = cv2.waitKey(1) & 0xFF
if the `q` key was pressed, break from the loop
if key == ord("q"):
        break
```

LIVE VIDEO CLASSIFACTION

```
# USAGE
# python video_classification.py --model resnet-34_kinetics.onnx --classes
action_recognition_kinetics.txt --input example_activities.mp4
# import the necessary packages
import numpy as np
import argparse
import imutils
import sys
import cv2
import datetime
import time
# construct the argument parser and parse the arguments
#firebase = firebase.FirebaseApplication('https://video-classification-96483-default-
rtdb.firebaseio.com/', None)
ap = argparse.ArgumentParser()
ap.add_argument("-m", "--model", required=True,
	help="path to trained human activity recognition model")
ap.add_argument("-c", "--classes", required=True,
	help="path to class labels file")
ap.add_argument("-i", "--input", type=str, default="",
	help="optional path to video file")
args = vars(ap.parse_args())
CLASSES = open(args["classes"]).read().strip().split("\n")
```

```
SAMPLE_DURATION = 16

SAMPLE_SIZE = 112

print("[INFO] loading human activity recognition model...")

net = cv2.dnn.readNet(args["model"])

print("[INFO] accessing video stream...")

vs = cv2.VideoCapture(0)

while True:

        frames = []

        for i in range(0, SAMPLE_DURATION):

                # read a frame from the video stream

                (grabbed, frame) = vs.read()

                if not grabbed:

                        print("[INFO] no frame read from stream - exiting")

                        sys.exit(0)

                # otherwise, the frame was read so resize it and add it to

                # our frames list

                frame = imutils.resize(frame, width=400)

                frames.append(frame)

        # now that our frames array is filled we can construct our blob

        blob = cv2.dnn.blobFromImages(frames, 1.0,

                (SAMPLE_SIZE, SAMPLE_SIZE), (114.7748, 107.7354, 99.4750),

                swapRB=True, crop=True)

        blob = np.transpose(blob, (1, 0, 2, 3))

        blob = np.expand_dims(blob, axis=0)
```

```
# pass the blob through the network to obtain our human activity
# recognition predictions
net.setInput(blob)
outputs = net.forward()
label = CLASSES[np.argmax(outputs)]
lbl=str(CLASSES[np.argmax(outputs)])
print(str(CLASSES[np.argmax(outputs)]))
for frame in frames:
        # draw the predicted activity on the frame
        cv2.rectangle(frame, (0, 0), (300, 40), (0, 0, 0), -1)
        cv2.putText(frame, label, (10, 25), cv2.FONT_HERSHEY_SIMPLEX,
                0.8, (255, 255, 255), 2)
        datetime1=datetime.datetime.now()
        Status=[str(CLASSES[np.argmax(outputs)])]
        data1={"Status":[str(CLASSES[np.argmax(outputs)])]}
        print(data1)
        a=1
        #if a==1:
          #firebase.put('', 'Location 1',data1)
        # display the frame to our screen
        cv2.imshow("Activity Recognition", frame)
        key = cv2.waitKey(1) & 0xFF
        # if the `q` key was pressed, break from the loop
        if key == ord("q"): Sbreak
```

A3 SCREENSHOTS

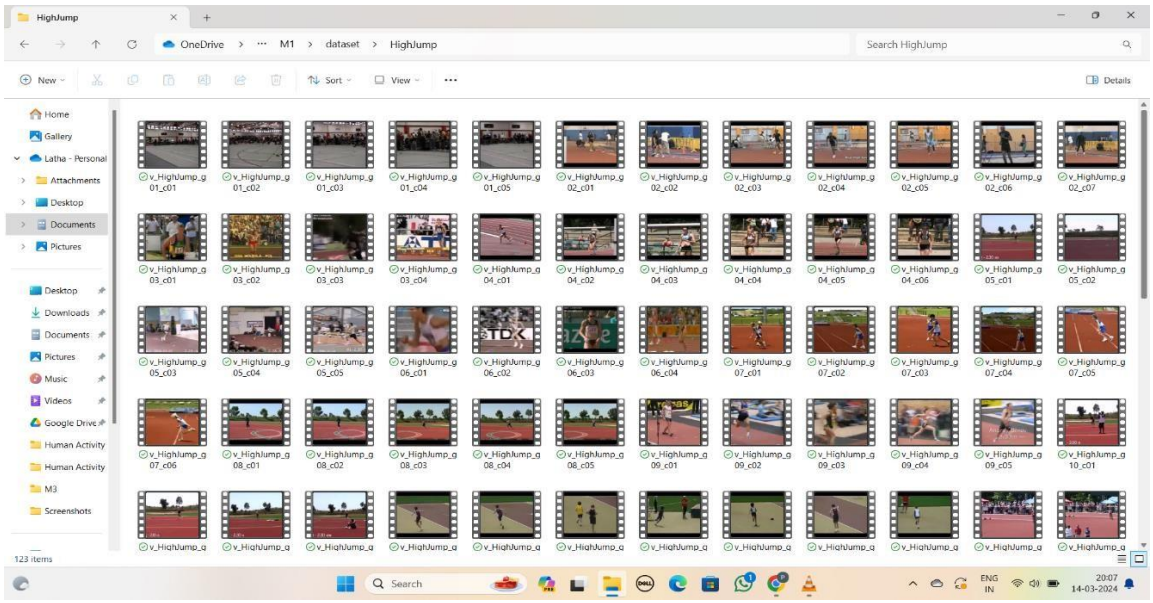


Figure A Video inputs

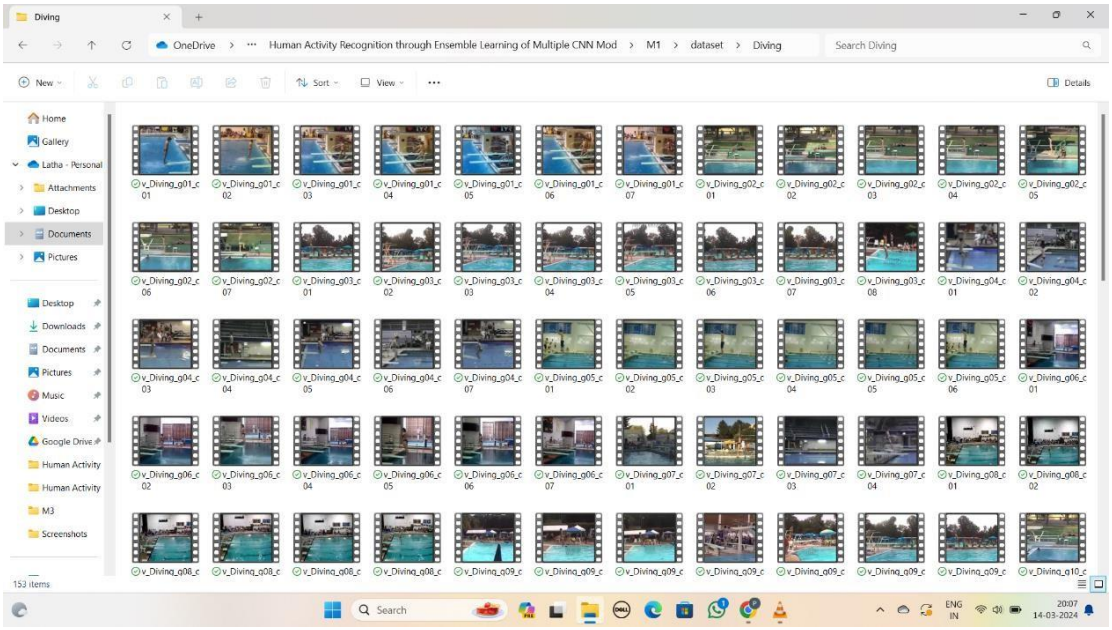


Figure B Video inputs

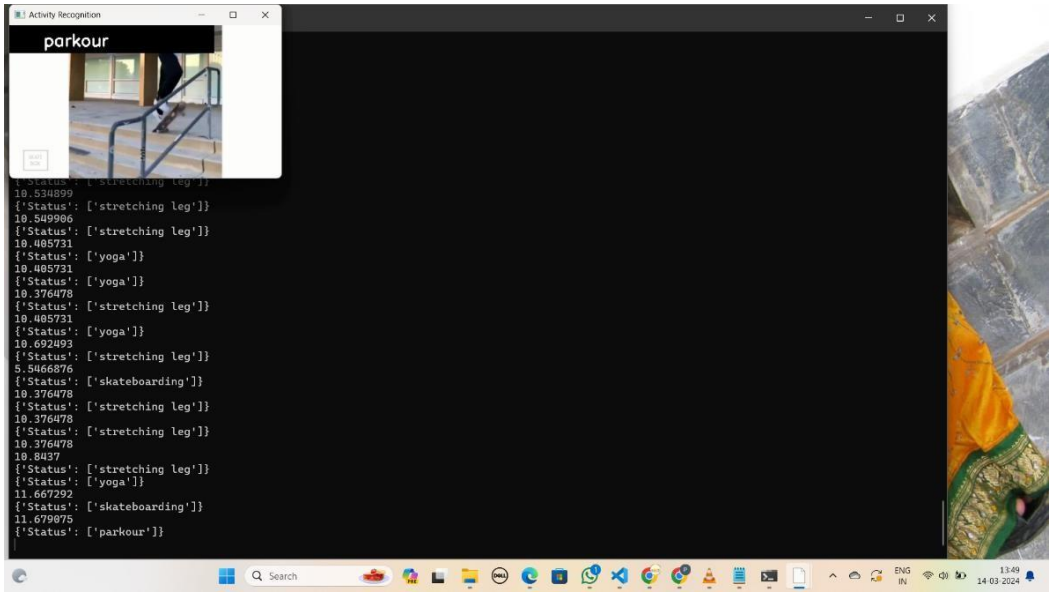


Figure C video Recognition

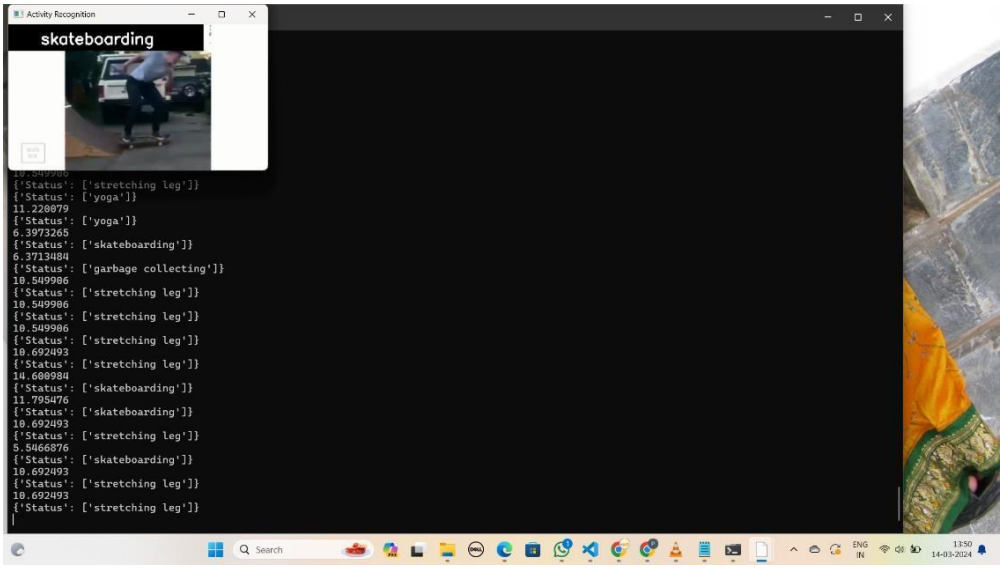


Figure D video Recognition

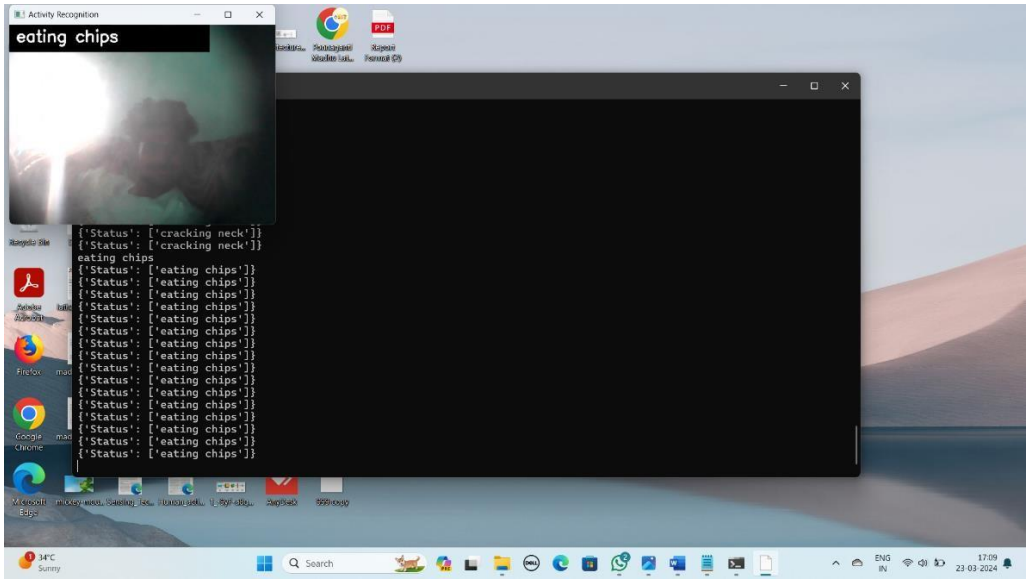


Figure E Live Video Recognition

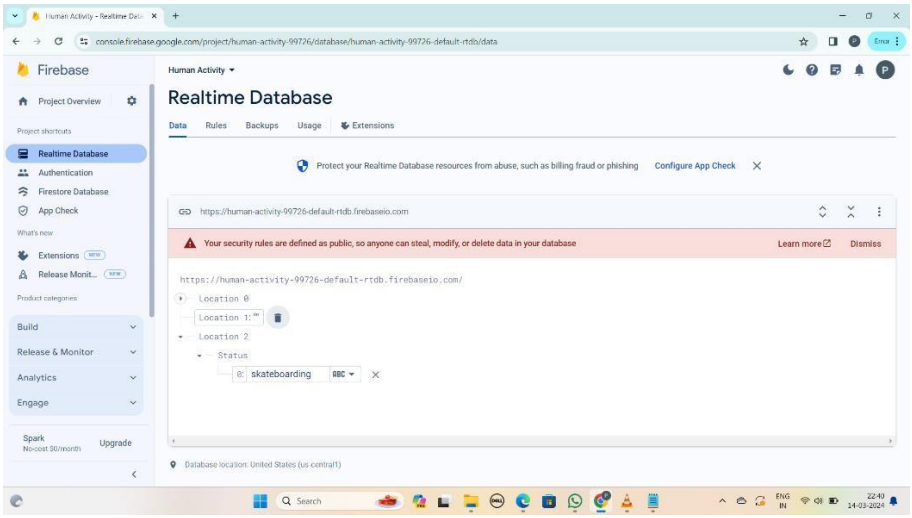


Figure F Output in Firebase

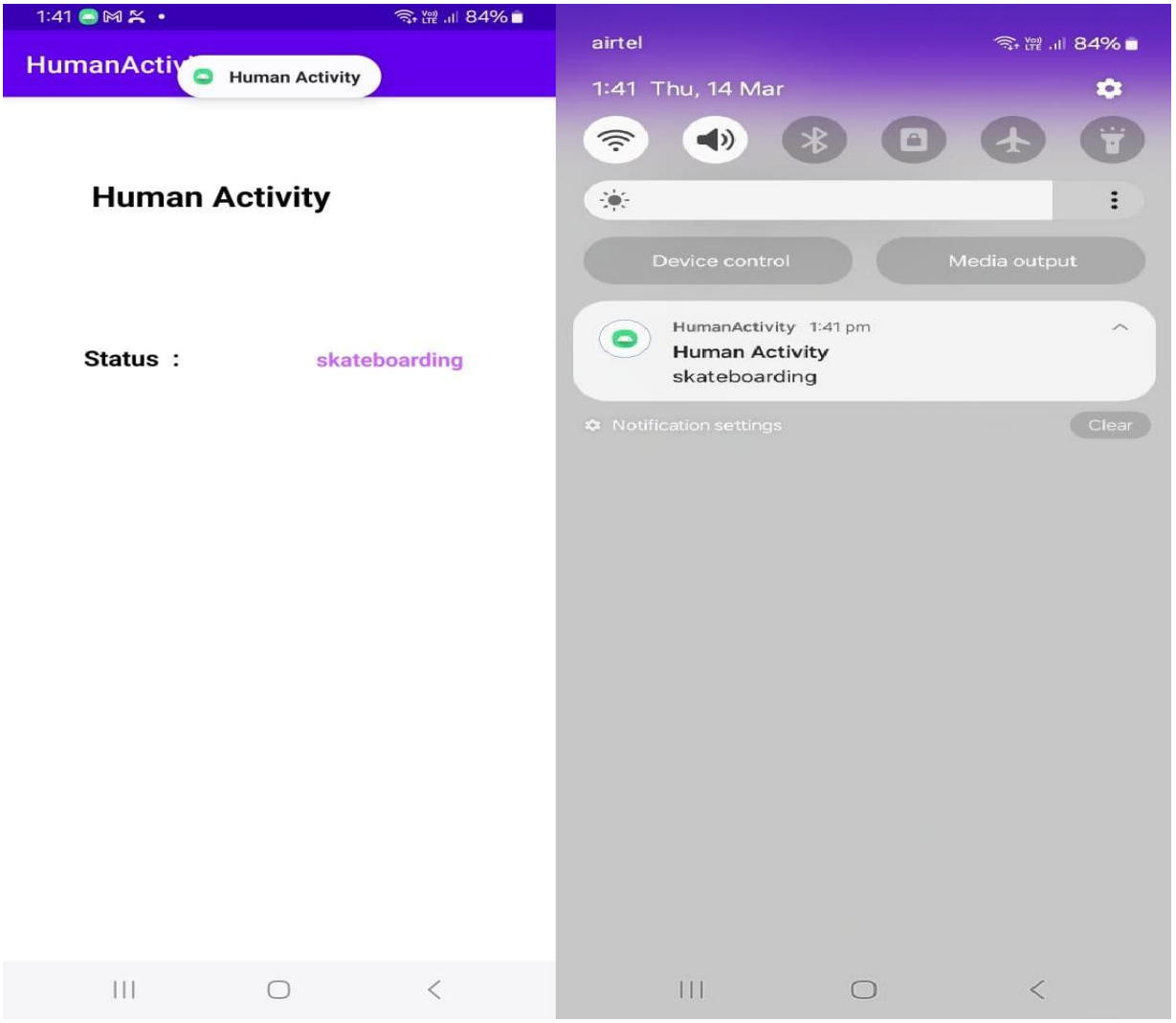


Figure G Output tracking in Mobile Application

PLAGIARISM REPORT

Detection of Physical Activity Using Ensemble Learning of Various CNN Models

Preena Jacitnth Shalom
Assistant professor, *Dept of CSE*
Panimalar Engineering College
Chennai, India
spjshalom@gmail.com

Ponnaganti Tharak Srinivas Manoj
Dept of CSE
Panimalar Engineering College
Chennai, India
tharaksrinivas5@gmail.com

Gurram Lokesh
Dept of CSE
Panimalar Engineering College
Chennai, India
lokeshgurram163014@gmail.com

Kamburi Aarif Ahmed
Dept of CSE
Panimalar Engineering College
Chennai, India
kaarifahmed29@gmail.com

***Abstract*—Comprehending what people do from videos is a challenging challenge in robot vision. The main role of intelligent video systems is to instinctively recognize and store the activities that humans are performing in the video stream. Most individual-centered challenges, such as those involving medical and one-on-one support, require inferring a variety of human actions, from basic to complicated. Thus, a clear classification of technologies for detecting is necessary for the methodical development of platforms that recognize behavior by humans. In the present research, we offer an innovative method that uses ensemble learning of several convolutional neural network (CNN) models to recognize human activities. Using the publicly accessible dataset, three distinct CNN algorithms are trained, and several ensembles of the models are produced. A success rate of 94% is obtained by the ensemble of the first two algorithms, which is higher than the techniques found in the existing literature.**

***Keywords*—Convolution Neural Network, Video Classification, Deep Learning, ResNet-34, Action Recognition.**

I. Introduction

The rising popularity associated with the Human Activity Recognition system is attributed to the rise in the quantity of video surveillance equipment. Identifying a substance's activities and goals through a sequence of examinations based on the object's behavior and its surroundings is the aim of movement assessment. The primary body regions' rotational postures and kinematics are the basis for our human movement recognition research in this publication. Although we provide a brief explanation of the physique part monitoring technique and other linked concerns, the main outcome of this study is a unique approach to individual behavior identification significantly outperforms existing approaches. formerly, feature extraction from the gathered information points was done in a task-dependent, bayesian manner by hand. Using attributes that are carefully chosen and built based on the program has proven essential to the feature acquisition process. From the unprocessed warnings, data points like the average and variance as well as modified coding metrics like Fourier transforms were taken out and used for classification. The drawback of these techniques is that they can only be used on the particular categorization jobs for which they were created. Another issue with a customized selection of features is that details derived from the unprocessed signal may be lost.

Installing sensors for the outdoors in the house is necessary and can be expensive. Vision technologies are seen as invasive equipment since they employ photographic evidence for classification. The alternative remedy is connected clothing which draws in researchers mostly because of how widely used it is. Wearable technology particularly health monitors and smartwatches is largely utilized for identification because of its many incorporated or constructed sensors, which include accelerometers, gyroscopes, and alignment sensors and are reasonably priced. Modern smartphones are also employed as an alternative to wearable technology for activity identification since they are more affordable, discrete, and have more integrated sensors than wearable technology. Additionally, they are mostly used for applications that run in real-time. Under the feature-based techniques approach, either local or worldwide characteristics, or both, are retrieved to help with the calculation of numbers for activity identification, whereas, in the prototype-based approach, a human model is constructed for action categorization.

Deep learning has rapidly evolved in the past few decades, achieving capabilities never before seen in a variety of fields, including the processing of natural language, visual object identification, and logical deduction. In contrast to conventional PR techniques, deep learning successfully trains a from beginning to-end artificial neural network to acquire considerably more sophisticated and significant characteristics while also significantly reducing the effort required to construct fresh attributes. Furthermore, it is more practical to carry out unsupervised and gradual learning using a network with a complicated architecture. Deep learning has thus been extensively studied in the literature and is the best strategy for HAR. A collection of multivariate indices, or rotational poses, are obtained from the test activity's video sequence during the recognition step. A vote vector is produced for each activity model in the information system for every video frame in the test activity sequence. The log-likelihood that the indexed posture is part of the activity modeling is represented temporally by this selection vector. Using sequential correlation, the votes for each test screen are integrated to get the total vote for each activity category.

The procedure has sped with previously unheard-of increases due to deep learning developments, particularly with the availability of more data and more processing power. Deep learning models can autonomously gather features, which

may be utilized for different categorization tasks, eliminating the requirement for the feature selection procedure to be dependent on specific tasks. Given the availability of deep learning, it makes sense to leverage this cutting-edge technology to its fullest potential for focused HAR adoption. Thus, the goal of this study is to provide a methodical procedure for more efficient visualization of features so that CNN ensemble learning can enhance activity identification. Convolutional artificial neural networks are implemented quite well in the HAR work that is currently underway. In reference, Deep CNN utilizes a unique unified layer to adjust automatic feature extraction from direct inputs, ultimately concatenating diagrams of features. Additionally, it can enhance the deep learning model's resilience and ability to generalize, a few benefits of deep learning over artificial intelligence techniques include lessening the reliance on specialized expertise during the component development approach. Reliable identification of characteristics changes over time. A reduction in the duration of image-based acceptance training. Excellent results even with poorly tagged information. Because of all these special characteristics, deep learning techniques outperform HAR-based solutions. Employing deep learning techniques, peer-reviewed papers on HAR were retrieved from electronic databases such as PubMed, PsycINFO, Web of Science, Scopus, and Google Scholar. This allowed for a summary of the different models that deep learning suggested for HAR in real-time and comparison datasets.

II. Literature Background

A. *Background*

The majority of the CCTV cameras in use today are used for surveillance and oversight. It was a conventional way of determining the kinds of work that were completed. Watching our security camera footage for a while is difficult. The current approach requires a lot of labor for tracking the works. Our suggested technique reduces the amount of manpower required. We suggest using a convolution neural network approach in this framework to recognize actions in videos. The internet-connected camera will be used to record the user's video. The photographic input is transformed to a certain amount of pixels. The specific portion of the frame is then identified using the CNN (Convolution Neural Network) technique. Then, using a convolution-based neural network, the greatest weightings are extracted from the feature extraction frames. Ultimately, the action will be detected in the videos, followed by its label (action name). After that, the output is sent to Firebase, which is and the user receives a notice on their Android gadget with the Realtime value. Utilizing the available data, we developed and evaluated the models we constructed as well as their groups. Training losses and accuracies for the three CNN models. Over the course of the fifty-epoch period, precision crossed and got close to more than eighty percent after a few epochs. When opposed to the training process, there was more variance seen in the validation accuracy, which ranged from 70 to 80%.

B. *Approaches and Related Work*

Researchers Yang, Nguyen, San, Li, and Krishnaswamy (2020) tackle the challenge of recognizing human activities (HAR) using sensor data. Traditionally, handcrafted features were crucial for accurate classification, but these methods might miss subtle movement details. This work proposes a new approach using deep convolutional neural networks (CNNs). CNNs automatically learn informative features directly from raw sensor data, eliminating the need for manual feature design. Supervised learning further refines these features, making them more distinctive for recognizing different activities. This two-pronged approach of superior feature extraction and enhanced classification leads to improved performance compared to existing HAR methods, as proven by experiments on benchmark datasets.

Jindong Wanga,b, Yiqiang Chena,b,__, Shuji Haoc, Xiaohui Penga,b, Lisha Hua,b Deep Learning for Sensor-based Activity Recognition. This research paper investigates the use of deep learning for sensor-based activity recognition. While traditional methods rely on manual feature extraction, deep learning allows for automatic feature extraction from sensor data, leading to improved performance. This survey explores recent advancements in this field, looking at the types of sensors used, the deep learning models applied, and the potential applications. The authors also analyze existing research and propose areas for future exploration in this field.

Schalk Wilhelm Pienaar1, Reza Malekian Human Activity Recognition Using LSTM-RNN Deep Neural Network Architecture. This paper explores using a Long Short-Term Memory (LSTM) neural network for Human Activity Recognition (HAR). Unlike traditional methods requiring manual feature engineering, LSTMs can automatically learn features from raw sensor data. This approach holds promise for various applications like fitness trackers and safety monitoring. The authors discuss a publicly available dataset (WISDM) containing sensor readings from various activities. They propose an LSTM architecture and report achieving over 94% accuracy and under 30% loss within the first 500 training iterations. This suggests the effectiveness of LSTMs for HAR tasks.

Michał Wo´zniaka,_, Manuel Gra˜nab, Emilio Corchadoc A Survey of Multiple Classifier Systems as Hybrid Systems. This research focuses on applying deep learning to recognize human activities using sensor data. Traditional methods rely on manually crafted features, but deep learning can automatically extract these features, leading to better performance. The authors explore recent advancements in this field, analyzing the types of sensors used, the deep learning models applied, and potential applications. They also provide insights into existing research and propose areas for future exploration in this domain.

Madhuri Panwar1, S. Ram Dyuthi1, K. Chandra Prakash1, Dwaipayan Biswas2, Amit Acharyya1, Koushik Maharatna3, Arvind Gautam1, Ganesh R. Naik4. CNN-Based Approach for Activity Recognition using a Wrist-Worn Accelerometer. This study proposes a Convolutional Neural Network (CNN) approach for recognizing human activities using wrist-worn accelerometers. Conventional methods rely on manual feature engineering, which can be challenging. This approach overcomes this hurdle by automatically extracting features through deep learning, reducing complexity. The authors designed a model to recognize three basic forearm movements using data from four subjects. They tested the model's effectiveness under various data pre-processing conditions and noise levels using three evaluation methods. The results demonstrate the proposed CNN method achieves an average recognition accuracy of 99.8%, significantly outperforming

traditional techniques like K-means clustering, linear discriminant analysis, and support vector machines.

III. Proposed Methodology

A. Artificial Neural Networks

Deep Learning's computational models draw some inspiration from the structure of the brain of a person. Artificial Neural Networks are the numerous training layers (ANN).

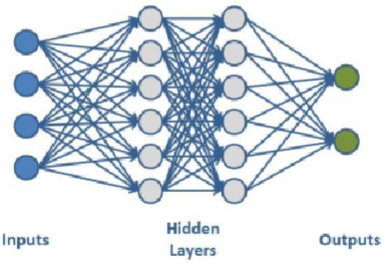


Fig. 1. ANN Process

Artificial neural networks (ANNs) are computing devices, which may be hardware or methods, that are based on the neuronal architecture of the human brain's cortex, but on a much smaller scale. It is a computing system composed of several basic, highly linked processing components that process data by responding dynamically to inputs from outside sources. There are layers of neurons in artificial neural networks. A computer unit called a neuron uses balanced input parameters to determine the value of a piece of data. Each input that the neuron accepts is given a unique weight. To create output, inputs are added together and run through a function that is not linear. Neurons in each layer pick up extra information, including the edges of objects in pictures or malignancies in people. You may employ many layers of neurons to find out more details about the parameters that were entered.

An artificial neural network (ANN) is a network of linked nodes that resembles the extensive network of layers of neurons in the human brain. An arrow shows a link from one neuron's output to another individual's input, and each circular node represents an artificial neuron. First layer inputs are passed through. The inputs are received by each neuron, each of which is assigned a unique value. Following that, an outcome is generated using these values. After that, the first layer's outputs are sent to the second layer for processing. This keeps on till the finished product is generated. It is assumed that the right output has already been determined. The outcome will be contrasted to the proper result each time data is sent through the network, and adjustments are made to their quantities until the network consistently produces an appropriate final output.

B. Convolutional Neural Networks

Convolutional neural networks (CNNs, or ConvNets) are multi-layer neural networks that are used to extract visual patterns from pixel photographs. CNN uses the term "convolution" to describe the mathematical function. You can combine two functions to demonstrate how one function can change the shape of another in a particular type of linear operation. To extract information from the image, two images that are represented as two matrices are simply multiplied to get the output. CNNs are similar to other neural networks, but their use of many convolutional layers makes things more complicated. CNN cannot function without convolutional layers. CNN artificial neural networks have shown to be quite successful in a variety of computer vision applications. It has influenced people's choices in several fields. A convolutional neural network automatically and adaptively learns the spatial hierarchies of the data through the use of backpropagation. It is made up of several layers, including convolution, pooling, and fully connected layers.

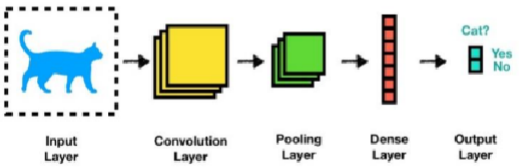


Fig. 2. CNN Architecture

Convolutional Layer: An essential part of the architecture of a convolutional neural network (CNN) is the convolutional layer. When applied to the input data, its main function is convolutional operations. Local patterns, edges, and textures are captured by the layer by applying a set of learnable filters to the input. This allows the network to extract meaningful features and capture spatial dependencies within the data. The convolutional layer slides the filters across the input using a receptive field, performing multiplications and summing the results to produce activation maps. These activation maps represent the response of the filters to different spatial locations in the input. The layer also incorporates biases and activation functions to introduce non-linearities, enhancing the network's modeling capabilities. Convolutional layers let CNNs efficiently receive and understand complex visual input for tasks like object and picture detection, which makes them crucial for feature extraction.

Pooling Layer: An essential part of the architecture of a convolutional neural network (CNN) is the pooling layer. Reducing the spatial dimensions of the input data while maintaining significant features is its primary function. This function helps to control the number of parameters and computational complexity of the network. The pooling layer works separately on each feature map. It separates the input into regions that don't overlap, such as squares or rectangles, and then compiles the data for each zone. Max pooling, which chooses the maximum value inside each region, is the most used pooling operation. The average value is computed using other pooling techniques, such as average pooling.

The pooling layer serves two main purposes. Firstly, it reduces the spatial dimensions of the feature maps, effectively down-sampling the data. This makes the network more efficient by reducing the computational burden and the risk of overfitting. Second, pooling makes the model more robust to small spatial translations and changes in the input data by focusing on the most prominent features. By applying pooling operations, the CNN can extract important features while discarding unnecessary information. This improves the

network's ability to generalize and recognize patterns in different regions of the input. The pooling layer is typically placed after one or more convolutional layers and is followed by additional layers to further process the data and make higher-level decisions.

Fully Connected Layer: The dense layer, or fully linked layer, is a crucial part of the design of a convolutional neural network (CNN). Its main functionality is to perform high-level reasoning and decision-making based on the extracted features from the previous layers. Unlike the convolutional and pooling layers, which operate on local spatial regions, the fully connected layer connects every neuron to every neuron in the preceding layer. It takes the flattened feature maps as input and applies a set of learnable weights and biases to generate an output vector. Each neuron in the fully connected layer determines the weighted total of the inputs and then passes it via an activation function.

The fully connected layer's purpose is to learn complex relationships and make predictions based on the extracted features. It captures global dependencies in the data, enabling the network to recognize higher-level patterns and make classifications. In most cases, a softmax activation function is applied to the fully connected layer's output in order to produce probability scores for various classes in a classification task.

To make final predictions or decisions based on the task at hand, the fully connected layer integrates and processes the learned features, which is a critical component of the CNN architecture.

Activation-Function: An essential part of the architecture of the mechanism of activation function is a convolutional brain network or CNN for short. Its primary function is to add non-linearities to the network so that it can learn intricate patterns and predict non-linearly. In CNNs, the activation function is applied to the output of each neuron in the network. It introduces non-linear transformations to the weighted sum of inputs, adding flexibility and expressive power to the model. The Rectified Linear Unit (ReLU), which sets negative values to zero, and the Sigmoid function, which squashes the output between 0 and 1, are common activation functions used in CNNs.

The activation function enables the network to capture non-linear relationships and make accurate predictions. By introducing non-linearities, CNNs can model complex data distributions and learn intricate patterns in the data. Additionally, the activation function aids in resolving the vanishing gradient issue and enhances the network's convergence during training. The activation function plays a crucial role in CNN architectures by introducing non-linearities to the network, allowing it to learn and represent complex patterns, and improving the overall performance of the model.

Dropout Layers: The dropout layer is a regularization technique commonly used in Convolutional Neural Network (CNN) architectures. Its main functionality is to reduce overfitting by preventing co-adaptation of neurons during training. The dropout layer randomly selects a fraction of neurons and sets their outputs to zero during each training iteration. This means that these neurons are "dropped out" and do not contribute to the forward or backward pass. The dropout layer compels the network to acquire more resilient and universal features that aren't reliant on particular neurons by doing this. The dropout layer helps prevent overfitting by introducing noise and reducing the complex interdependencies between neurons. It encourages the network to learn multiple independent representations of the input, which improves its ability to generalize to unseen data. During inference or testing, the dropout layer is usually turned off, and the full network is used for prediction. This allows the model to make more reliable and accurate predictions. In CNN designs, the dropout layer is a regularization approach that randomly deactivates neurons during training to assist prevent overfitting. By forcing the network to acquire more robust and broad features, it enhances its ability to generalize to novel and unknown data. Normalization Layer: The normalization layer, also known as the batch normalization layer, is a technique commonly used in Convolutional Neural Network (CNN) architectures. Its main functionality is to normalize the activations of the network's neurons, improving the training process and accelerating convergence. The normalization layer operates by normalizing the output of a previous layer, typically the output of a convolutional or fully connected layer. It subtracts the mean and divides by the standard deviation of the mini-batch, effectively centering and scaling the activations. The normalization layer helps with the internal covariate shift, which is a variation in the distribution of network activations during training. Normalizing the activations, it stabilizes the learning process, making the network less sensitive to changes in input distributions. This leads to faster convergence, improved gradient flow, and better overall training performance. Furthermore, the normalization layer acts as a regularizer, reducing the reliance on other regularization techniques like dropout. It helps prevent overfitting by controlling the magnitude of activations, ensuring they stay within a reasonable range. The

normalization layer is a technique in CNN architectures that normalizes the activations of the network's neurons. It improves the training process, accelerates convergence, and helps prevent overfitting by addressing the internal covariate shift and stabilizing the learning process.

IV. Results

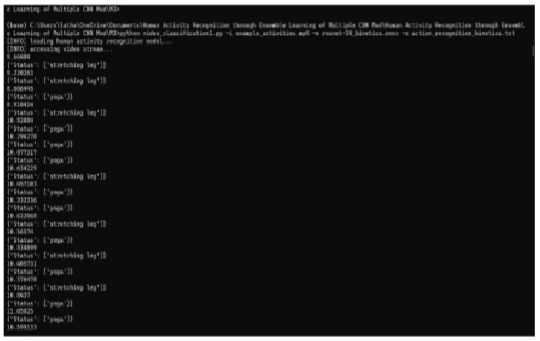


Fig. 3. Execution of CNN Model

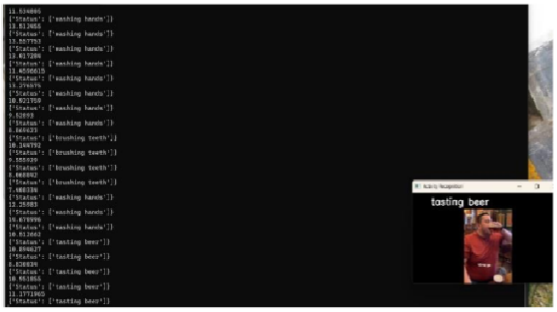


Fig. 4. Identification of Toasting Beer

Figure 4 represents the identification of human activity.

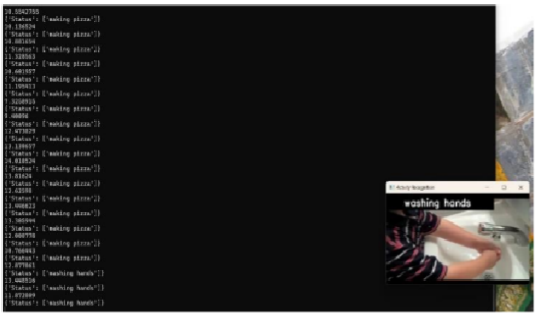


Fig. 5. Detecting the activity of washing hands

Figure 5 shows the physical activity of the human.

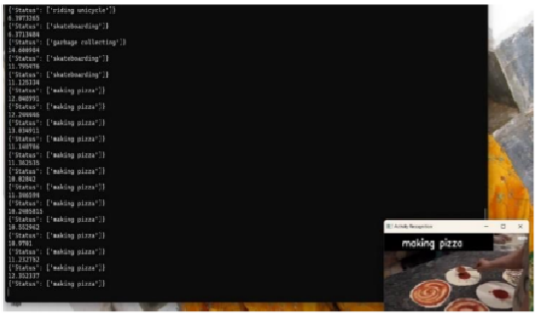


Fig. 6. Identifying of making pizza

Figure 6 Overview of the making of pizza.

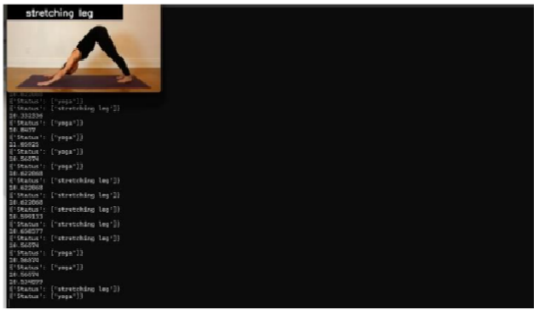


Fig. 7. Recognition of Stretching Leg

Figure 7 shows the outcome of the Stretching leg.

Fig. 8. Identification of Skateboarding

This picture shows the output of Skateboarding.

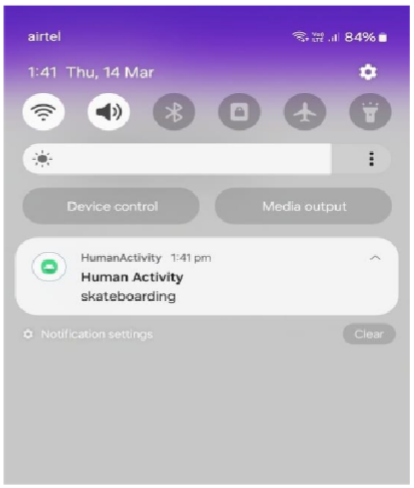


Fig. 9. Android Notification

This image is the notification of the physical activity in smart device.

V. Conclusion

We presented a novel approach in this research for automated gathering features for the job of human activity identification. The suggested approach creates a fresh deep architecture for the CNN to use while analyzing the wideband time series information. Convolution and pooling procedures are the primary methods used by this deep structure to capture the prominent patterns of the sensor outputs at various time scales. Every noteworthy pattern that has been found is methodically unified across several channels and then integrated into the various categories of human conduct. By autonomously learning high-level representation, deep learning outperforms standard methods for pattern recognition and decreases reliance on human-crafted extraction of features.

References

[1] T. Pl¨otz, N. Y. Hammerla, and P. L. Olivier, "Feature learning for activity recognition in ubiquitous computing," in 22nd International Joint Conference on Artificial Intelligence, 2011.

[2] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, "LSTM networks for mobile human activity recognition," in International Conference on Artificial Intelligence: Technologies and Applications (ICAITA 2016), 2016.

[3] L. K. Hansen and P. Salamon, "Neural network ensembles," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993–1001, 1990.

[4] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in 24th International Joint Conference on Artificial Intelligence, 2015.

[5] M. Panwar, S. R. Dyuthi, K. C. Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, and G. R. Naik, "CNN based approach for activity recognition using a wrist-worn accelerometer," in 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2017, pp. 2438–2441.

[6]] S. W. Pienaar and R. Malekian, "Human activity recognition using LSTM-RNN deep neural network architecture," in 2019 IEEE 2nd Wireless Africa Conference (WAC). IEEE, 2019, pp. 1–5.

[7] J.Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensorbased activity recognition: A survey," Pattern Recognition Letters, vol. 119, pp. 3–11, 2019.

[8] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," Future Generation Computer Systems, vol. 81, pp. 307–313, 2018.

RE-2022-221862-plag-report

ORIGINALITY REPORT

13%	9%	12%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijrpr.com Internet Source	1%
2	B. Vinay Kumar, Y. Ayyappa, Bandaru Kanaka Aparna, Bheemineni Ravi Kiran, Bandi Naga Gopala Krishna, Eda Kavya. "Building an Intelligent Brain Tumor System using Magnetic Resonance Imaging", 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2023 Publication	1%
3	www.mdpi.com Internet Source	1%
4	Narjis Zehra, Syed Hamza Azeem, Muhammad Farhan. "Human Activity Recognition Through Ensemble Learning of Multiple Convolutional Neural Networks", 2021 55th Annual Conference on Information Sciences and Systems (CISS), 2021 Publication	1%

5	Nishanthi G, Vimala S. "Bulk SMS Communication Portal", IOS Press, 2020 Publication	1%
6	Altork, Jawaher. "Machine Learning Techniques for JetMET Data Certification of the CMS Detector at CERN", Marmara Universitesi (Turkey), 2023 Publication	1%
7	webthesis.biblio.polito.it Internet Source	1%
8	Sathiyabhama Balasubramaniam, Yuvarajan Velmurugan, Dhayanithi Jaganathan, Seshathiri Dhanasekaran. "A Modified LeNet CNN for Breast Cancer Diagnosis in Ultrasound Images", Diagnostics, 2023 Publication	1%
9	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Student Paper	1%
10	Phyo P. San, Pravin Kakar, Xiao-Li Li, Shonali Krishnaswamy, Jian-Bo Yang, Minh N. Nguyen. "Deep Learning for Human Activity Recognition", Elsevier BV, 2017 Publication	1%
11	odr.chalmers.se Internet Source	1%

12	Mohammad Arshad, Dr. Mohammad Ali Hussain. "A Hybrid Model for Detecting DDoS Attacks in Wide Area Networks", International Journal of Recent Technology and Engineering (IJRTE), 2019 Publication	<1%
13	Reema Gera, Kalyan Ram Ambati, Pallavi Chakole, Naveen Cheggoju, Vipin Kamble, V. R. Satpute. "Classifying Human Activities using CNN and ConvLSTM in Video Sequences", 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS), 2023 Publication	<1%
14	vdoc.pub Internet Source	<1%
15	Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-garadi, Uzoma Rita Alo. "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges", Expert Systems with Applications, 2018 Publication	<1%
16	dblp.uni-trier.de Internet Source	<1%