HUMAN ACTIVITY RECOGNITION

BY

USING CNN & LSTM

A Major Project Progress Report

Submitted in partial fulfillment for the degree of

Bachelor of Technology

BY

B. SRIDEVI	N180194
M. JYOTHI BHAVANI	N180177
N. N V E ANOOHYA	N180226
S. GEETHA	N180228
B. NISHA	N180898

Under the Guidance of

MR. CH SRINIVAS

Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge & Technologies

(Nuzvid, Eluru, Andhra Pradesh – 521202)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge & Technologies

(Nuzvid, Eluru, Andhra Pradesh – 521202)

CERTIFICATE OF COMPLETION

This is to certify that the work entitled **"HUMAN ACTIVITY RECOGNITION BY USING CNN & LSTM"** is the work of **B. SRIDEVI (ID: N180194), M. JYOTHI BHAVANI (ID: N180177), N. N V E ANOOHYA (ID: N180226), S. GEETHA (ID: N180228), B. NISHA (ID: N180898)** has been carried out under the guidance of **Mr. Ch Srinivas (Assistant Professor, Department of CSE)**, Computer Science and Engineering, Rajiv Gandhi University of Knowledge & Technologies - Nuzvid, Andhra Pradesh. The project is approved for submission requirements of Major Project in the final year in Computer Science and Engineering from Rajiv Gandhi University of Knowledge & Technologies.

Mr. Ch Srinivas
Assistant professor,
Department of CSE,
RGUKT - Nuzvid.

Dr. D. V. Nagarjana Devi
Assistant Professor,
Head of the Department,
Department of CSE,
RGUKT - Nuzvid.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge & Technologies

(Nuzvid, Eluru, Andhra Pradesh – 521202)

CERTIFICATE OF EXAMINATION

This is to certify that the work entitled **"HUMAN ACTIVITY RECOGNITION BY USING CNN & LSTM"** is the work of **B. SRIDEVI (ID: N180194), M. JYOTHI BHAVANI (ID: N180177), N. N V E ANOOHYA (ID: N180226), S. GEETHA (ID: N180228), B. NISHA (ID:N180898)** and here by accord our approval of it as a study carried out and presented in a manner required for its acceptance in the final year of Bachelor of Technology for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn, as recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

Mr. Ch Srinivas
Assistant Professor,
Department of CSE,
RGUKT-Nuzvid.

Project Examiner,
RGUKT-Nuzvid.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge & Technologies

(Nuzvid, Eluru, Andhra Pradesh – 521202)

CERTIFICATE OF DECLARATION

We **B. SRIDEVI (ID: N180194), M. JYOTHIBHAVANI (ID: N180177), N. N V E ANOOHYA (ID: N180226), S. GEETHA (ID: N180228), B. NISHA (ID: N180898)** declare that the project report entitled **"HUMAN ACTIVITY RECOGNITION BY USING CNN & LSTM"** done by us under the guidance of Mr. Ch Srinivas, Assistant Professor, Department of CSE is submitted for the fulfillment of a major project during the academic session July 2023-March 2023 at RGUKT-Nuzvid.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Date: 30-03-2024

Place: Nuzvid

B. SRIDEVI	(N180194)
M. JYOTHI BHAVANI	(N180177)
N. N V E ANOOHYA	(N180226)
S. GEETHA	(N180228)
B. NISHA	(N180898)

ACKNOWLEDGEMENT

We would like to express our profound gratitude and deep regards to our guide Mr. Srinivas Ch for his exemplary guidance, monitoring and constant encouragement to us throughout the B-Tech course. We shall always cherish the time spent with him during the course of this work due to the invaluable knowledge gained in the field of reliability engineering.

We are extremely grateful for the confidence bestowed in us and entrusting our project entitled **"HUMAN ACTIVITY RECOGNITION BY USING CNN & LSTM ".**

We express our gratitude to **Dr. D. V. Nagarjana Devi (HoD, Department of CSE)** and other faculty members for being a source of inspiration and constant encouragement which helped us in completing the project successfully. Finally, yet importantly, we would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project. .

ABSTRACT

Human activity recognition is a crucial task in the fields of computer vision and machine learning, with applications ranging from healthcare to security and beyond. The proposed methodology combines the strengths of CNN for spatial feature extraction and LSTM for temporal modeling. Initially, a CNN is utilized to process frames from video data, extracting relevant spatial features that represent different aspects of human activity.

These features are then fed into an LSTM network, which captures the temporal dependencies and context across frames, enabling the model to understand the sequential nature of activities. To evaluate the effectiveness of the approach, a diverse UCF50 Action Recognition Dataset comprising various human activities is used for training and testing. The results demonstrate the capability of the CNN-LSTM architecture to accurately recognize and classify a wide range of human activities, including walking, running, sitting, taichi, horserace, swing, walking with dog and more. The combination of CNN and LSTM models provides a robust solution for human activity recognition.

Deep learning methods for recognizing facial expressions[6] This thesis used deep learning techniques to identify seven main emotions: anger, disgust, fear, happiness, sadness,surprise, and neutrality. The following architectures were used in one of his works: VGG-16 and ResNet 50. A performance of 31.8% was determined using the SVM multiclass classifier as baseline accuracy. With improved results and ensemble and transfer learning techniques, the published accuracy was 67.2% for ensemble learning techniques and 78.3% for encapsulated transfer learning. And raw depth video-based RNNs are used with two CNN-LSTM models for detecting human action recognition[7] with different architectures but the same long- term learning strategy. One uses a stateless generator and the other emphasizes the stateful capabilities of her RNN, but especially applies to the specific case of human action detection.With this approach, the proposed model achieved accuracies of 73.26% (CS) and 72.45% (CV) for stateless models and 80.43% (CS) and 79.91% (CV) for stateful models. By these references we improved this project accuracy to 85% with considerations of specified action of human.

TABLE OF CONTENTS

Chapter 1: INTRODUCTION 1

Chapter 2: LITERATURE REVIEW 3

Chapter 3: REQUIREMENT AND ANAYSIS 5

Hardware Components

Software Components

Chapter 4: LIBRARIES AND MODULES 6

Chapter 5: PROPOSED METHODOLOGY 10

CNN Architecture

LSTM Architecture

Chapter 6: IMPLEMENTATION 14

Chapter 7: RESULT

Chapter 8: CONCLUSION 15

Chapter 9: FUTURE SCOPE

LIST OF FIGURES

Figure 5.1: Working of kernel with single input image 7

Figure 5.2: CNN Architecture 8

Figure 5.3: Working of kernel with an RGB images 9

Figure 5.4: LSTM Architecture 9

Figure 5.5: LRCN model network 10

Figure 5.6: Flow of work 11

Figure 6.1: Import the required Libraries 13

Figure 6.2: Download and visualize the data with its labels 13

Figure 6.3: Function of frames extraction 14

Figure 6.4: Sample Frames 15

Figure 6.5: Split the data into Train and Test set 15

Figure 6.6: Built LRCN model 16

Figure 6.7: Compile & Train the model 17

Figure 6.8: Plot Total Accuracy VS Total Validation Accuracy 17

Figure 6.9: Perform single prediction on a YouTube video 18

Figure 6.10: UserInterface 18

CHAPTER 1

INTRODUCTION

Human Activity Recognition is the process of identifying, analyzing and interpreting what kind of actions and goals one or more agents or persons will be performing. The decisions will be taken based on their previous actions performed with their behaviour. A typical human may perform the major activities such as walking, running, sitting, standing, laying, walking-upstairs, walking-downstairs, swing, tai chi, horse race, walking with dog etc. In his or her day today routine. Identifying and analyzing various human activities would be bringing out some smart solutions associated with childcare and eldercare fields if HAR could be integrated with IoT technologies and means. For example, suppose a situation where a child is kept at a day care center, parents have been gone for work and they need to check what their child is actually doing right now or are they safe at this moment, this HAR could be used as a measure to predict Human behaviour. Also, in case of elderly people monitoring guardians or caretakers could use this technology for making better safe environment for them by avoiding some actions elders tend to do. Thus, ranging from personal fitness, Gaming, security fields, health care industry and even more. HAR could bring attractive solutions for real life human problems.

CNN is great when we work with image data and need to solve something like image classification. LSTM is great when we work with sequences of data, but when we combine both of them together, we can do things like video classification and solve problems like action recognition.

In this project we used CNN and LSTM models. Here we are using UCF50-Action Recognition Dataset, consists of realistic videos taken from youtube. The entire process will be mentioned in further detail in this document. The video will go under test and train, then at the end it display the predicted action in the video. In future this technology can be used in healthcare and wellness monitoring.

CHAPTER 2
LITERATURE REVIEW

Alpoim, A. F. da Silva, and C. P. Santos, "Human Activity Recognition Systems: State of Art," [1] With the literature review been conducted, it was revealed that the Deep Learning Models have been widely used resulting better scales of accuracy and to serve the Human Activity Recognition process. In our approach, the implementations were carried out under four main areas. First, developed a basic CNN model for the activity recognition. Second, developed a LSTM network model for our dataset. As the third step, developed a CNN and LSTM hybrid model for the classification and the prediction of the six activities. Finally, proposed a ConvLSTM model which is a further extension of the CNN LSTM model to perform the convolutions of the CNN as part of the LSTM. Prior to developing the models and carried a comprehensive detailed exploratory data analysis of the WISDM dataset.

R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers,"[2] identifying a users's activity level and predicting their energy consumption monitoring user activity levels in order to promote health & fitness detecting a fall & movements after a fall. process for adressing the activity recognition tasks of data collecction, data preprocesing, data transformation, experiments, results activities like walking, jogging, ascending strairs, descending strairs.

B. Marinho, A. H. de Souza Junior, and P. P. Reboujas Filho, "A New Approach to Human Activity Recognition Using Machine Learning Techniques"[3] in Intelligent Systems Design and Applications. In this paper, they used the HCI-HAR dataset. The semi-supervised active transfer learning model proposed in this paper is a technique that uses existing semi-supervised learning, active learning, and transfer learning. Although it is not novel, no technology has previously applied it. Although the demand for data is increasing as research into deep learning continues expanding, labeling remains a challenging area as it is expensive. Labeling tasks are challenging, particularly in areas where humans need to collect data directly such as in HAR. To compensate for these shortcomings, the model used in the proposed work was able to guarantee 95.9% performance on the mHealth dataset while reducing the number of data labeling by 10% and the HCI-HAR dataset with 2.6% more accuracy and 80% less labeling than ATL. Therefore, the proposed semi-supervised active transfer learning is an effective way to reduce the cost of labeling tasks. The proposed research can be used in industries that require labeling tasks by administrators but need to

effectively build data such as the medical field and the field of HAR.

Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, LSTM Networks for Mobile Human Activity Recognition [4] Human Activity Recognition (HAR) has been a challenging problem yet it needs to be solved. . In this paper, we present various state-of-the-art methods and describe each of them by literature survey. Different datasets are used for each of the methods where in the data is collected by different means such as sensors, images, accelerometer, gyroscopes, etc. and the placement of these devices at various locations. The results obtained by each technique and the type of dataset are then compared. Machine learning techniques like decision trees, K-nearest neighbours, support vector machines, hidden markov models are reviewed for HAR and later the survey for deep neural network techniques like artificial neural networks, convolutional neural networks and recurrent neural networks is also presented.

James Wong, Alexandru Savoiu: Recognizing Facial Expressions Using Deep Learning, page 2-5. 2020.Deep learning methods for recognizing facial expressions[6] This thesis used deep learning techniques to identify seven main emotions: anger, disgust, fear, happiness, sadness,surprise, and neutrality. The following architectures were used in one of his works: VGG-16 and ResNet 50. A performance of 31.8% was determined using the SVM multiclass classifier as baseline accuracy. With improved results and ensemble and transfer learning techniques, the published accuracy was 67.2% for ensemble learning techniques and 78.3% for encapsulated transfer learning.

David Fuentes-Jimenez, Cristina Losada-Gutierrez Adrian Sanchez-Caballero, Universidad de Alcala “Exploiting the ConvLSTM: Human Action Recognition using Raw Depth Video-Based Recurrent Neural Networks”, Pages 5-17.arXiv:2006.07744v1, 2020. In this paper raw depth video-based RNNs are used with two CNN-LSTM models for detecting human action recognition[7] with different architectures but the same long- term learning strategy. One uses a stateless generator and the other emphasizes the stateful capabilities of her RNN, but especially applies to the specific case of human action detection.With this approach, the proposed model achieved accuracies of 73.26% (CS) and 72.45% (CV) for stateless models and 80.43% (CS) and 79.91% (CV) for stateful models. By these references we improved this project accuracy to 85% with considerations of specified action of human.

.

CHAPTER 3

REQUIREMENTS AND ANALYSIS

Hardware Requirements

Laptop or Desktop

CPU: Intel core i3 or higher

RAM: 4GB or higher

Software Components

Python

Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning and AI, Python's simplicity allows developers to write reliable systems.

Jupyter Notebook

Jupyter Notebook is an open-source web application for interactive data analysis and scientific computing. It allows users to create and share documents containing live code, equations visualizations, and narrative text. Jupyter Notebook supports multiple programming languages, including Python, R, and Julia. It provides an intuitive and flexible environment for data exploration, prototyping, and collaboration.

Operating System – Windows

CHAPTER-4

LIBRARIES AND MODULES

Libraries

Os library

Python OS module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system. The OS comes under Python's standard utility module.

Open CV

Opencv is an open source library which is very useful for computer vision applications such as video analysis, CCTV footage analysis and image analysis. OpenCV is written by C++ and has more than 2,500 optimized algorithms. When we create applications for computer vision that we don’t want to build from scratch we can use this library to start focusing on real world problems.

Math module

We use the math module to perform various mathematical calculations, such as numeric, trigonometric, logarithmic, and exponential calculations. This tutorial will explore the common constants and functions implemented in the math module and how to use them.

Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

Collections

Although librarians are conscious of the need to preserve copies of materials for posterity, the principal role of library collections in practice is not archival but is the need to provide convenient physical access to the materials likely to be needed by the population to be served.

TensorFlow

TensorFlow is an open-source machine learning framework. it can be used to develop models for various tasks, including image recognition, video classification.

Pytube

Pytube is a lightweight library written in python. It has no third-party dependencies and aims to be highly reliable. Pytube also makes pipelining easy, allowing you to specify callback functions for different download events, such as on progress or on complete.

FRONT-END

HTML

HTML (Hypertext Markup Language) is the standard markup language for creating web pages. It uses tags to structure and present content on the web, defining elements like headings, paragraphs, link, images and more. HTML provides the foundation for web development and acts as a fundamental building block for creating and displaying content on the internet.

CSS

CSS is a stylesheet language that styles HTML elements by selecting them and applying visual properties, enabling developers to customize the appearance and layout of web pages.

JavaScript

JavaScript is a scripting language used for client-side and server-side web development, allowing for dynamic and interactive web content. It supports variables, functions, and objects, enabling the creation of complex applications and enhancing user experience. JavaScript is widely used in modern web development and has a vast ecosystem of libraries and frameworks to facilitate efficient development workflow.

BACK-END

Flask

Python's Flask micro web framework is well-liked and frequently used to create online apps. It offers a straightforward and adaptable method for developing Python-based web applications and APIs (Application Programming Interfaces).

CHAPTER-5

PROPOSED METHODOLOGY

Convolutional Neural Network (CNN)

Convolutional neural networks[8] refer to a sub-category of neural networks and have all the characteristics of neural networks. As image is a matrix of nxn dimension, an input image of nxn dimension and filter which can move along the image matrix calculate the convolution of the matrix to get output which provides a 3x3 matrix for the same and is called kernel or filter.

Input image is processed with the kernel. The kernel is sided over by 1 pixel which is also called a stride. And after every slide, compute the multiplication of part of matrix under the kernel and sum up the output which gives us integer which makes an element of the Convolved Feature matrix called a Feature Map. Convolved Feature matrix is also known as Feature Map.

The 3x3 matrix is a Kernel/Filter and the same sliding matrix is known as Filter or Kernel. In real world data the input image can be quite big and will contain lots of information most of which is not important for our deep learning or classification problem. To solve a problem at our hands Kernel or Filters are used.

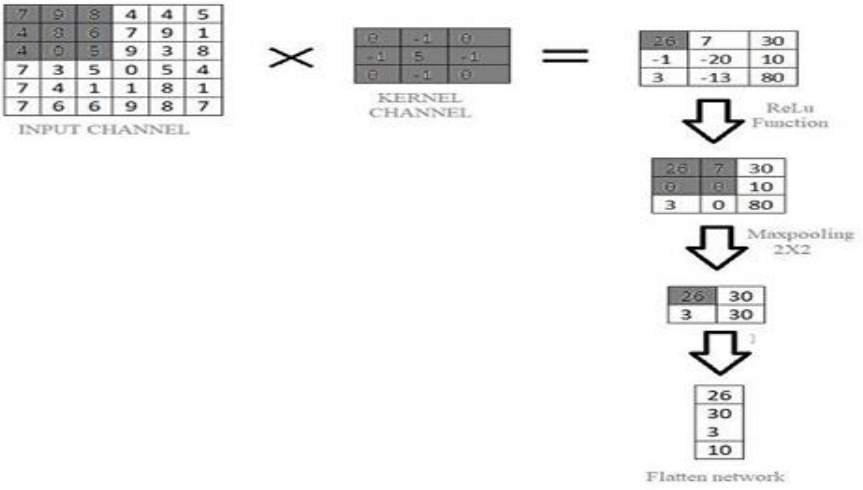

Figure 5.1 Working of kernel with single input image

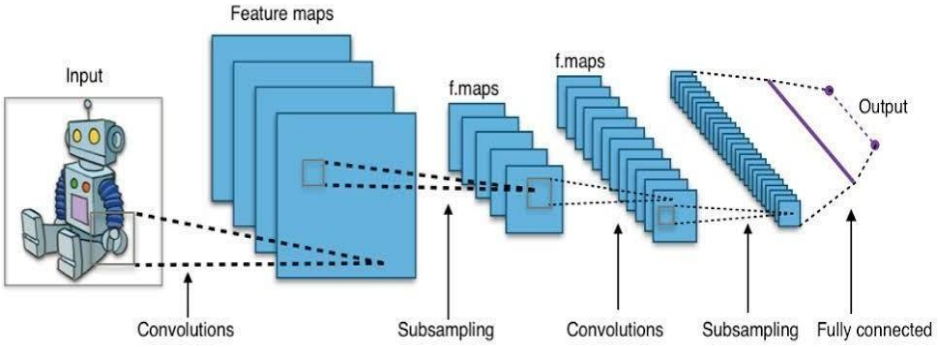

Figure 5.2 CNN Architecture

Feature Maps

The output obtained after repeatedly applying convolution over the sub-regions of an entire image is called Feature Maps. Previous layer is an input for initial pass. While doing convolution with help of nxn kernel and with a stride of 1, one pixel is moved at time and perform a convolution over a particular sub region of the input image, while doing that neurons get activated and the output is generated which contains the information about the features like edge, curve this information is then collected into a feature map.

Let's take an example of a 32x32 image, and using a receptive field of 5 x 5 over the whole image with stride equal to 1 will end up of having 28 x 28 sized feature map or distinct activation totaling to 784. This number is the times a neuron is fired off with the data of features.

An objective when involved in convolution operation is to extract features of high quality from an image frame. You can always add multiple transformation layers when creating a neural network. The first layer of convolution rounds off the gradients while the second limits the edges. Layers There is no magic number for how many layers to add, as the layers depend on the complexity of the image. Note that using the 3x3 filter effect on the original image will apply the effect to the dynamic 3x3 element. As such, images are usually overlapped at both end values to maintain their actual size.

3x3 is the size of kernel which is used to do convolution over the input image, it is called 3x3 convolution. Benefits of using a 3 x 3 Convolution is that it is very much effective to detect the local feature in the image. Being of smaller size the computation cost is also relatively less then the other let's say 5 x 5 convolution. When 3x3 convolution is applied over a N x N image, as aresult output is feature map of (N-2) x (N-2).

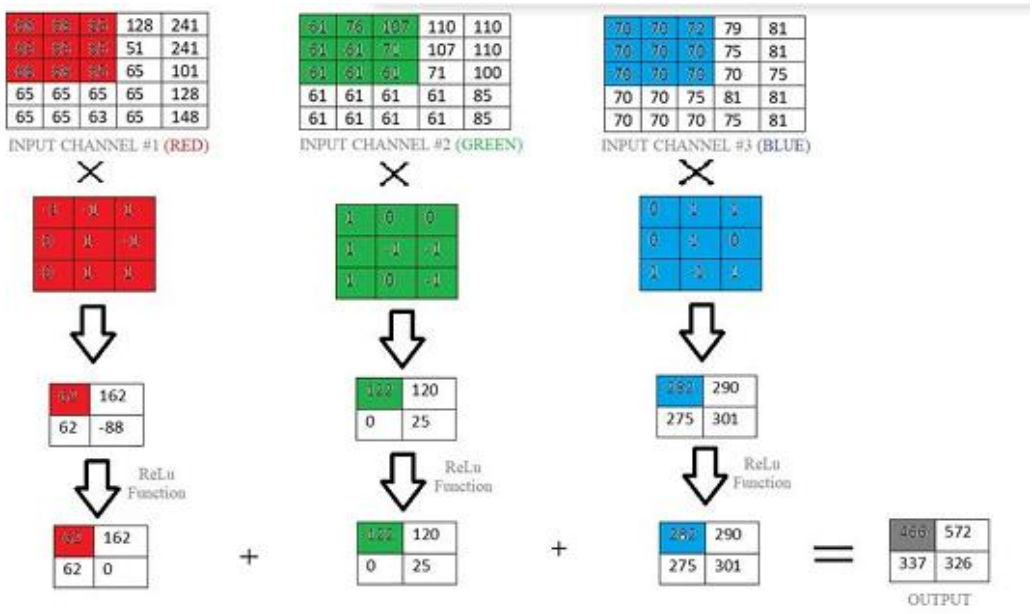

Figure 5.3 Working of kernel with an RGB images

Long Short-Term Memory

Long Short-term Memory Networks; a special form of RNN, which is known to adapt long-term dependencies. They have been delivered by means of Hochreiter & Schmidhuber (1997), and had been refined and popularized within the next paintings.They work very well on a variety of major problems, and are now widely used. LSTMs[9] are specifically designed to avoid the problem of long-term dependence. All emerging neural networks have a kind of repetitive series of neural network modules. LSTMs interacts with all the layers involved within a network with the repeating module constructed with distinction.

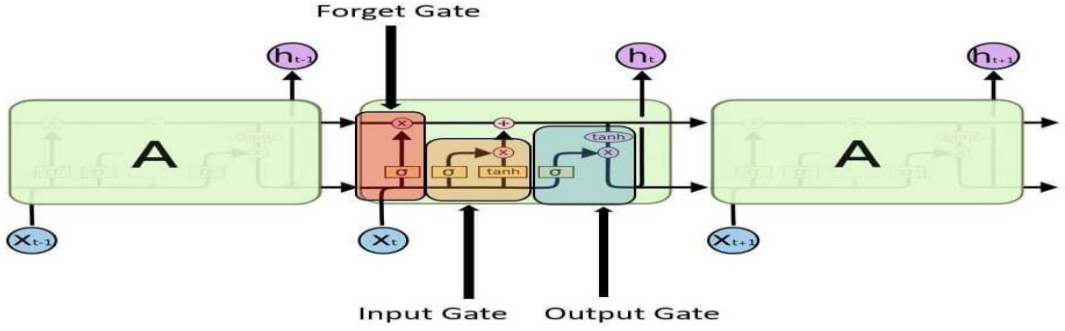

Figure 5.4 LSTM Architecture

In the diagram above, each line indicating a whole vector, i.e., from the output to the input of the other. The pink circles dictate independent operation the vector addition, while the yellow boxes are for aid of the layers of the neural networks. Line integration means merging, while the fork line displays its copied content and copies to different locations.

- Decide what data to discard within the network. This choice turned into made by using a sigmoid layer called the "gate oblivion layer". Draw a number between 0 and 1 by looking at ht - 1 and xt for each number in the Ct-1 cell state. output 1 means "stop doing this", output 0 means "get rid of it completely".
- The subsequent step is to determine what new information to keep within the cell environment. There are two components to this. First, a sigmoid layer referred to as the "input gate layer" determines the price to check. The responsibility of tanh layer is to deliver the produced vector of new candidate values C~t this subsequent step will combine the two into one international update.
- Now update the status of the old cell Ct-1 to the new status of Ct. The previous step determined what to do. you really just have to do it.
- We repeat old situations with ft and forget what we wanted to forget before. Then add ∗ C ~ t.
- Finally, you need to decide what to release. This output is based on the cell status but is a filtered version. First, we use a sigmoid layer that determines which parts of the cell structure are freed. To get the desired state of the cell set it to tanh and then let the sigmoid gate to multiply and extract the desired part.

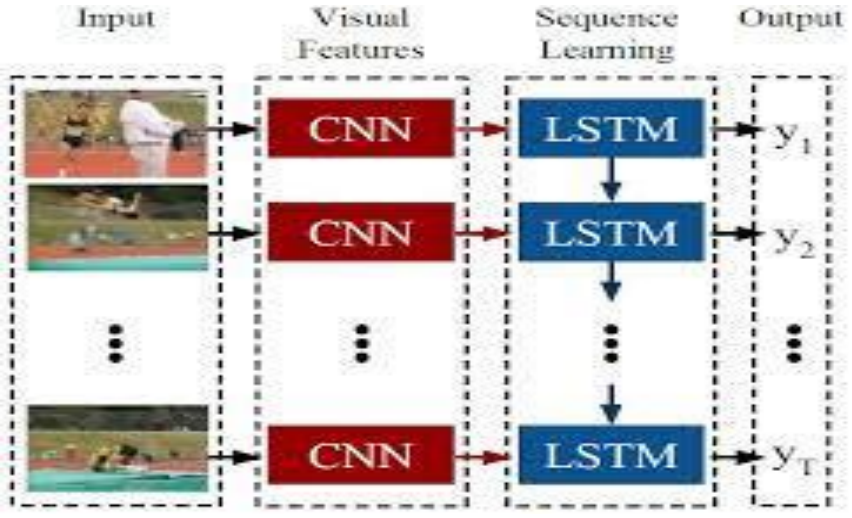

Figure 5.5 LRCN model network

Flow of work

The input divided into frames. Each frame undergo image classification using CNN we extract the feature of the image .then by using LSTM we predict the action in the video.

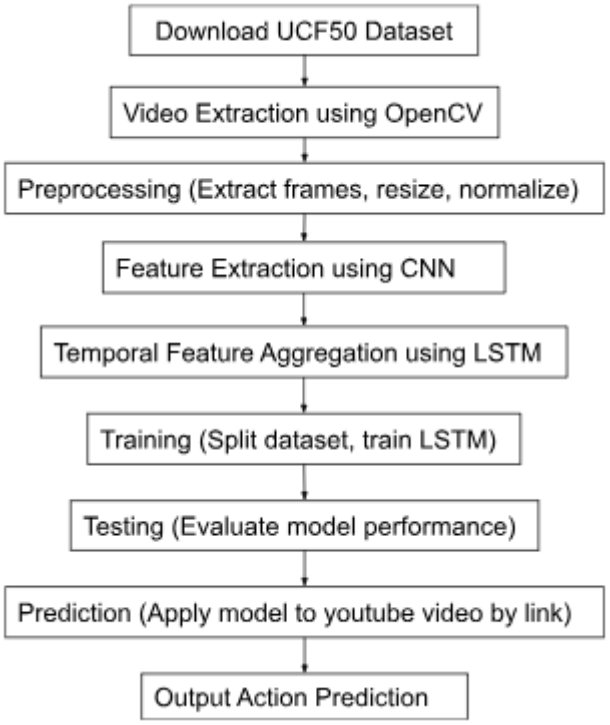

Figure 5.6 Flow of work

Download UCF50 Dataset: Download and visualize the data and labels. It is an Activity Data Recognition Set, which includes real-time videos taken from YouTube that separates this data set from many other data recognition systems that are not real and played by players. Dataset contains:

50 Action Classes with a total video of 6671

25 Video Groups per Activity Category

133 Medium Videos per Activity Category

199 Average number of frames per video

320 Frame Range for Each Video

240 Average Frame Length Each video

Medium frames 26 per second per video

Video Extraction: Extracting the video content from the videos of dataset. The extraction ensures that the system can work with the video data for further analysis. Successful extraction ensures that the subsequent preprocessing steps can be applied to the video frames.

Preprocessing: Preprocessing begins with the extraction of individual frames from the video content. Resizing the frames to a standardized size ensures consistency in input data for subsequent processing. Normalizing pixel values adjusts the intensity values of the frames to a common scale, aiding in feature extraction.

Feature Extraction using CNN: Feature extraction employs a Convolutional Neural Network (CNN) to capture spatial features from individual frames. The CNN processes each frame to extract relevant visual features, such as edges, shapes, and textures. Features extracted by the CNN represent the visual characteristics of the actions present in the video.

Temporal Feature Aggregation using LSTM: Temporal feature aggregation employs a Long Short-Term Memory (LSTM) network to capture temporal dependencies across frames. The LSTM models sequential information, enabling the understanding of action sequences over time. By considering the order and timing of frame features, the LSTM enhances the model's capability to recognize dynamic actions.

Training: The dataset is typically split into training, validation, and testing sets to assess the model's performance. During training, the LSTM network is fed with sequences of features extracted by the CNN. The model learns to recognize patterns in the input data and generalize to unseen examples.

Testing: Testing evaluates the performance of the trained model using a separate testing dataset. The model's ability to generalize to unseen data is assessed by measuring various performance metrics.

Prediction: Prediction involves applying the trained model to new, unseen youtube videos by giving link to infer action labels. The model processes the video frames and generates predictions for each frame or sequence of frames.

Output Action Prediction: The output action prediction step presents the predicted action labels to the user or downstream systems. The predicted labels provide information about the recognized actions in the input youtube video.

CHAPTER-6

IMPLEMENTION

```
import os
import cv2
import pafy
import math
import random
import numpy as np
import datetime as dt
import tensorflow as tf
from collections import deque
import matplotlib.pyplot as plt

from moviepy.editor import *
%matplotlib inline

from sklearn.model_selection import train_test_split

from tensorflow.keras.layers import *
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.utils import plot_model
```

Figure 6.1 Import the required Libraries

```
seed_constant = 27
np.random.seed(seed_constant)
random.seed(seed_constant)
tf.random.set_seed(seed_constant)
```

Done to reduce the degree of randomness in the data and to make the data more less constant in all the runs that we will be doing

```
# Discard the output of this cell.
%%capture

# Downlaod the UCF50 Dataset
!wget --no-check-certificate https://www.crcv.ucf.edu/data/UCF50.rar

#Extract the Dataset
!unrar x UCF50.rar
```

Figure 6.2 Download and visualize the data with its labels

Studying video files in the database and change the frame size of the videos to a specified width and length, in order to reduce the figures and make the data normal to distance [0-1] by dividing 255pixel values, enabling faster integration while training pre-specified network IMAGE_HEIGHT, IMAGE_WIDTH and SEQUENCE_LENGTH. These static conditions can be increased for best results, although increasing the sequence length only applies to a certain area, and

increasing the values will result in a more costly process for the computer.

```
def frames_extraction(video_path):
    '''
    This function will extract the required frames from a video after
resizing and normalizing them.
    Args:
        video_path: The path of the video in the disk, whose frames
are to be extracted.
    Returns:
        frames_list: A list containing the resized and normalized
frames of the video.
    '''

    # Declare a list to store video frames.
    frames_list = []

    # Read the Video File using the VideoCapture object.
    video_reader = cv2.VideoCapture(video_path)

    # Get the total number of frames in the video.
    video_frames_count =
int(video_reader.get(cv2.CAP_PROP_FRAME_COUNT))

    # Calculate the the interval after which frames will be added to
the list.
    skip_frames_window = max(int(video_frames_count/SEQUENCE_LENGTH),
1)

    # Iterate through the Video Frames.
    for frame_counter in range(SEQUENCE_LENGTH):

        # Set the current frame position of the video.
        video_reader.set(cv2.CAP_PROP_POS_FRAMES, frame_counter *
skip_frames_window)

        # Reading the frame from the video.
        success, frame = video_reader.read()

        # Check if Video frame is not successfully read then break the
loop
```

```
        resized_frame = cv2.resize(frame, (IMAGE_HEIGHT, IMAGE_WIDTH))

        # Normalize the resized frame by dividing it with 255 so that
 each pixel value then lies between 0 and 1
        normalized_frame = resized_frame / 255

        # Append the normalized frame into the frames list
        frames_list.append(normalized_frame)

    # Release the VideoCapture object.
    video_reader.release()

    # Return the frames list.
    return frames_list
```

Figure 6.3 Function of frames extraction

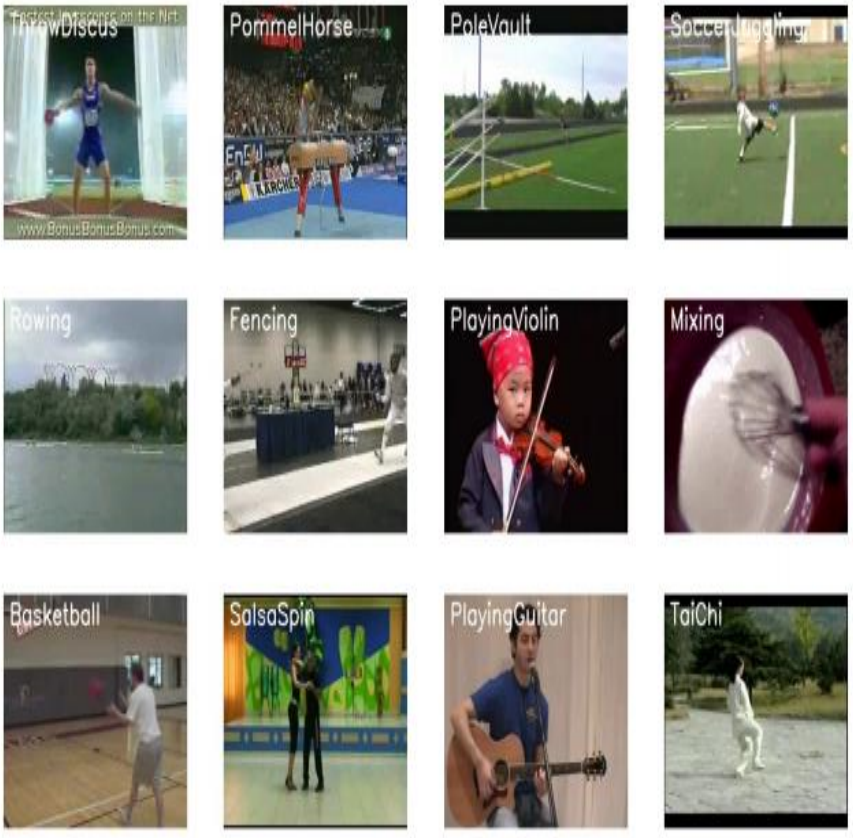

Figure 6.4 Sample Frames

The frames_extraction () function is defined as creating a list containing framed and standard frames for the video in which the channel is transmitted as an argument. The function will read the video file frame by frame, although not all frames are added to the list as we will only need an evenly distributed sequence length.

```
# Split the Data into Train ( 75% ) and Test Set ( 25% ).
features_train, features_test, labels_train, labels_test =
train_test_split(features, one_hot_encoded_labels,

test_size = 0.25, shuffle = True,

random_state = seed_constant)
```

Figure 6.5 Split the data into Train and Test set

```
def create_LRCN_model():
    '''
    This function will construct the required LRCN model.
    Returns:
        model: It is the required constructed LRCN model.
    '''

    # We will use a Sequential model for model construction.
    model = Sequential()

    # Define the Model Architecture.

########################################################################
################################################

    model.add(TimeDistributed(Conv2D(16, (3, 3),
padding='same',activation = 'relu'),
                              input_shape = (SEQUENCE_LENGTH,
IMAGE_HEIGHT, IMAGE_WIDTH, 3)))

    model.add(TimeDistributed(MaxPooling2D((4, 4))))
    model.add(TimeDistributed(Dropout(0.25)))

    model.add(TimeDistributed(Conv2D(32, (3, 3),
padding='same',activation = 'relu')))
    model.add(TimeDistributed(MaxPooling2D((4, 4))))
    model.add(TimeDistributed(Dropout(0.25)))

    model.add(TimeDistributed(Conv2D(64, (3, 3),
padding='same',activation = 'relu')))
    model.add(TimeDistributed(MaxPooling2D((2, 2))))
    model.add(TimeDistributed(Dropout(0.25)))

    model.add(TimeDistributed(Conv2D(64, (3, 3),
padding='same',activation = 'relu')))
    model.add(TimeDistributed(MaxPooling2D((2, 2))))
    #model.add(TimeDistributed(Dropout(0.25)))

    model.add(TimeDistributed(Flatten()))

    model.add(LSTM(32))

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

Figure 6.6 Built LRCN model

The LRCN Approach is used to combine the Convolution and LSTM layers into a unique model. Another similar approach would be to use the CNN model and the separately trained LSTM model. A CNN model can be used to extract local features from frames in a video, and a pre-trained model can be used for this purpose, which can be better prepared for the problem while the LSTM model can predict performed action on video. But here, we will use another method known as the Long-term Recurrent Convolutional Network (LRCN), which integrates CNN and LSTM layers into a single model. Convolutional layers are used to exclude a local element from frames, and extruded local features are given to the LSTM layer for each step of the sequence modeling time. In this way, the network learns the spatiotemporal features directly in the final training, resulting in a stronger model.

```
# Create an Instance of Early Stopping Callback.
early_stopping_callback = EarlyStopping(monitor = 'val_loss', patience
= 15, mode = 'min', restore_best_weights = True)

# Compile the model and specify loss function, optimizer and metrics
to the model.
LRCN_model.compile(loss = 'categorical_crossentropy', optimizer =
'Adam', metrics = ["accuracy"])

# Start training the model.
LRCN_model_training_history = LRCN_model.fit(x = features_train, y =
labels_train, epochs = 200, batch_size = 4 ,
                                                shuffle = True,
validation_split = 0.2, callbacks = [early_stopping_callback])
```

Figure 6.7 Compile & Train the model

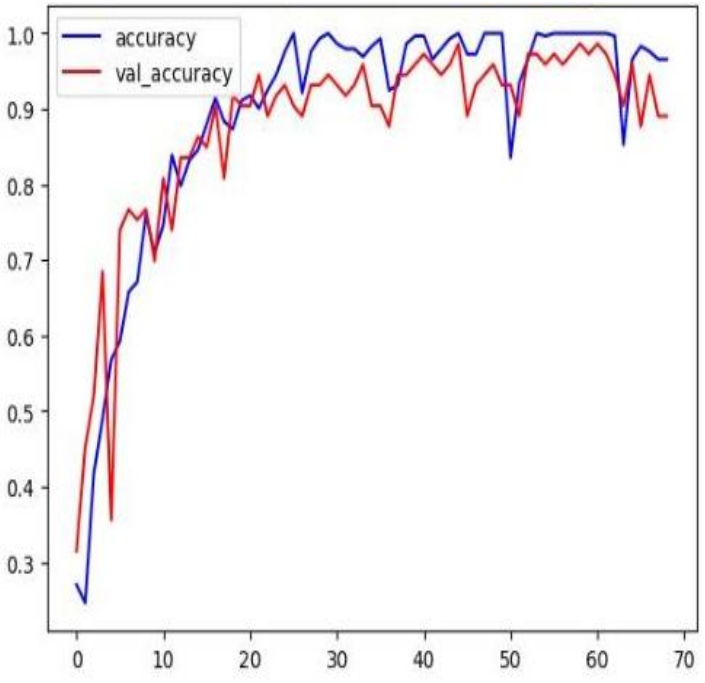

Figure 6.8 Plot Total Accuracy VS Total Validation Accuracy

CHAPTER -7

RESLUT

```
video_url = 'https://youtu.be/fc3w827kwyA'
video_title = download_youtube_video(video_url, test_videos_directory)

# Get the path of the downloaded video
if video_title:
    input_video_file_path = os.path.join(test_videos_directory,
f"{video_title}.mp4")
    print(f"Downloaded video: {input_video_file_path}")
else:
    print("Failed to download the video.")

# Construct tihe nput youtube video path
input_video_file_path = f'{test_videos_directory}/{video_title}.mp4'

# Perform Single Prediction on the Test Video.
predict_single_action(input_video_file_path, SEQUENCE_LENGTH)
```

```
Downloaded video: test_videos/Comparison of Four Styles of Tai Chi.mp4
1/1 [==============================] - 0s 37ms/step
Action Predicted: WalkingWithDog
Confidence: 0.9748544692993164
```

Figure 6.9 Perform single prediction on a YouTube video

From the results, it seems that the LRCN model has performed very well in a small number of classes. so in this step, we will set up an LRCN model to test in other youtube videos. We will create a download function for_youtube_videos () to download YouTube videos first using the pafy library. The library only needs the URL in the video to download it and its related metadata as the title of the video. Next, we will create a predict_on_video () function that will automatically read the frame of the video frame in a path that has been transmitted as an argument and will perform action recognition on the video and save the results.

Figure 6.10 User Interface

CHAPTER-8

CONCLUSION

Proposed a CNN-LSTM approach to human activity recognition using UCF50-Human Activity Recognition dataset. This approach helps in frame extraction from a video robustly while taking advantage of both the CNN and LSTM models which works great with Spatial and temporal feature extraction respectively. We evaluated the model metric using total loss vs total validation loss and total accuracy vs total validation accuracy for each of these architectures of CNN-LSTM model. The LRCN model out performed with an accuracy of 80%. This model is applied with complex activities as dataset categories to tackle the recognition of the activity.

CHAPTER-9

FUTURE WORK

Future work, the model is to be trained to recognize action on multiple people performing different activities in the frame. To train this asset in the model the data used should be annotated for more than one person's activity and also provide the bounding box coordinates of the person along with the activity he or she is performing or a hacky way is to crop out each person and perform activity recognition separately on each person but this will be very expensive.

Long-Term Context Modeling: Enhancing LSTM models to capture long-term dependencies and contextual information over extended periods can lead to more accurate and context-aware activity recognition systems.

Edge Computing: Deploying lightweight LSTM and CNN models on edge devices can reduce latency and bandwidth requirements, enabling real-time inference and improving the scalability of activity recognition systems.

Human Pose Estimation: Combining LSTM and CNN with human pose estimation techniques can enable fine-grained activity recognition by analyzing body movements and gestures, opening up applications in gesture recognition and human-computer interaction.

Enhancing LSTM models to capture long-term dependencies and contextual information over extended periods can lead to more accurate and context-aware activity recognition systems.

REFERENCES

[1] L. Alpoim, A. F. da Silva, and C. P. Santos, "Human Activity Recognition Systems: State of Art," in 2019 IEEE 6th Portuguese Meeting on Bioengineering (ENBENG), Lisbon, Portugal, Feb. 2019, pp. 1-4, doi: 10.1109/ENBENG.2019.8692468.

[2] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," SIGKDD Explor. Newsl., vol. 12, no. 2, pp. 74-82, Mar. 2011, doi: 10.1145/1964897.1964918.1

[3] C. Jobanputra, J. Bavishi, and N. Doshi, "Human Activity Recognition: A Survey," Procedia Computer Science, vol. 155, pp. 698-703, 2019, doi: 10.1016/j.procs.2019.08.100.

[4] P. Kuppusamy and C. Harika, "Human Action Recognition using CNN and LSTM-RNN with Attention Model" International Journal od Innovative Technology and Exploring Engineering(IJITEE), vol.8, Issue 8, pp.1639-1643, 2019.

[5] L. B. Marinho, A. H. de Souza Junior, and P. P. Reboujas Filho, "A New Approach to Human Activity Recognition Using Machine Learning Techniques," in Intelligent Systems Design and Applications, vol. 557, A. M. Madureira, A. Abraham, D. Gamboa, and P. Novais, Eds. Cham: Springer International Publishing, 2017, pp. 529-538.

[6] James Wong, Alexandru Savoiu: Recognizing Facial Expressions Using Deep Learning, page 2-5. 2020.

[7] David Fuentes-Jimenez, Cristina Losada-Gutierrez Adrian Sanchez-Caballero, Universidad de Alcala “Exploiting the ConvLSTM: Human Action Recognition using Raw Depth Video-Based Recurrent Neural Networks”, Pages 5-17.arXiv:2006.07744v1, 2020.

[8] https://en.m.wikipedia.org/wiki/Convolutional_neural_network

[9] https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/