DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

(HUMAN ACTIVITY DETECTION USING DEEP LEARNING)

By

David Antony Selvaraj A – ENG18CS0002
Adarsh SM – ENG18CS0016
Gagan GR – ENG18CS0102
Karthik B – ENG18CS0126
Ramesh Roshan M – ENG18CS0224

Under the supervision of

Mr. CVSN REDDY
Associate Professor, CSE Department

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **"HUMAN ACTIVITY DETECTION USING DEEP LEARNING"** is carried out by **A David Antony Selvaraj (ENG18CS0002), Adarsh SM (ENG18CS0016), Gagan GR (ENG18CS0102), Karthik B (ENG18CS0126), Ramesh Roshan (ENG18CS0224),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Prof CVSN Reddy	**Dr. Girisha G S**	**Dr. A Srinivas**
Associate Professor Dept. of CSE, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner	**Signature of Examiner**
1.	
2.	

DECLARATION

We, **David Antony Selvaraj A (ENG18CS0002), Adarsh SM (ENG18CS0016), Gagan GR (ENG18CS0102), Karthik B (ENG18CS0126), Ramesh Roshan M (ENG18CS0224),** are student's of the seventh semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled **"HUMAN ACTIVITY DETECTION USING DEEP LEARNING"** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Student **Signature**

Name1: David Antony Selvaraj A
USN: ENG18CS0002
Name2: Adarsh SM
USN: ENG18CS0016
Name3: Gagan GR
USN: ENG18CS0102
Name4: Karthik B
USN: ENG18CS0126
Name5: Ramesh Roshan
USN: ENG18CS0224

Place: Bangalore
Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Mr. CVSN Reddy**, **Associate Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinators Dr. Meenakshi Malhotra and Dr. Bharanidharan N, and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

Page

LIST OF ABBREVIATIONS vi
LIST OF FIGURES ………... vii
LIST OF TABLES ………….. viii
ABSTRACT ix

CHAPTER 1 INTRODUCTION………... 1
1.1. PURPOSE... 2
1.1.1 INTENDED AUDIENCE…………………………………………….. 2
1.2. SCOPE.. 3
CHAPTER 2 PROBLEM DEFINITION ……….. 4-5
CHAPTER 3 LITERATURE SURVEY.. 6-8
CHAPTER 4 PROJECT DESCRIPTION.. 9
4.1. PROPOSED DESIGN 10-11
4.2. ASSUMPTIONS AND DEPENDENCIES... 12
CHAPTER 5 REQUIREMENTS 13
5.1. FUNCTIONAL REQUIREMENTS... 14
5.2. NON-FUNCTIONAL REQUIREMENTS... 15
5.2.1 PERFORMANCE REQUIREMENTS 15
5.2.2 SOFTWARE QUALITY ATTRIBUTES 15
5.3. SOFTWARE REQUIREMENTS 15
5.4 HARDWARE REQUIREMENTS 15
CHAPTER 6 METHODOLOGY.. 16-19
CHAPTER 7 EXPERIMENTATION.. 20-23
CHAPTER 8 TESTING AND RESULTS …... 24-29
CHAPTER 9 CONCLUSION AND FUTURE SCOPE…………………………... 31
9.1. CONCLUSION ………………………………………………………… 31
9.2. FUTURE SCOPE……………………………………………………….. 31
REFERENCES... 32
APPENDIX A 33
OPENCV……………………………………………………………………………. 33
APPENDIX B 33
UCF50……………………………………………………………………………….. 33
APPENDIX C 33
CNN……………………………………………………..…………………….. 33
Funding and Published Paper details 34-36

LIST OF ABBREVIATIONS

HAD	Human Activity Detection
HAR	Human Activity Recognition
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
DL	Deep Learning
GPU	Graphics Processing Unit
Open CV	Open Source Computer Vision
ANN	Artificial Neural Network

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
4.1(a)	Workflow Diagram of video classification method	10
4.1(b)	Classification on videos into several image under CNN	10
4.1(c)	Workflow Diagram of Human Activity Detection using Deep Learning	11
8	Data Visualizations with labels	25
8.1	Before Frame Resizing	25
8.2	After Frame Resizing	25
8.3	Total Accuracy VS Total Validation Accuracy	26
8.4	Total Loss VS Total Validation Loss	26
	Result Screenshots	27-29

LIST OF TABLES

Table No.	Description of the Table	Page No.
8	Comparison between ConvLSTM and LRCN	26
8.1	Evaluation of Results	29

ABSTRACT

In this modern world, Human activity Recognition and Detection are gaining importance where it is and will be used in many upcoming innovations, not only in security and surveillance but also in understanding the behavioral patterns of humans.

The purpose of this project is to develop a model that would take the video input from different sources like recorded videos and detect actions that are being performed by the people which is commonly referred to as Human Activity Recognition or Human Activity Detection. To emphasize its importance the project aim is to build a model which can intelligently classify the video clips related to human activities. The project is making use of keras library where video classification is involved and implement the model by using CNN and LSTM algorithms. The expected result can be a wise application that could predict various human activities. The project has made use of the UCF50 - Action Recognition dataset to train the model.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

Action recognition has been a widely researched topic in Computer Vision for over a couple of decades. Its applications in real-time surveillance and security makes it more challenging and interesting. Various approaches have been taken to solve the problem of Action Recognition, however the majority of the current approaches failed to address the issue of a large number of action categories and highly unconstrained videos taken from dataset.

The topic of Human Activity Recognition (HAR) is a prominent research area topic in the field of Computer Vision and Image Processing. It has empowered state-of-the-art application in multiple sectors such as surveillance, digital entertainment, medical healthcare domains. It is interesting to observe and intriguing to predict such kind of movements.

1.1 PURPOSE

The purpose of human activity recognition is to recognize the physical tasks of a particular individual. A few of these physical tasks may be carried out by a particular individual through changes in the entire body via. a particular body part movement, some activities are performed like performing hand gestures. Some actions may be done wherein a human interacts with objects such as TaiChi, WalkingWithDog, Swing, HorseRace.

1.1.1 INTENDED AUDIENCE

Human activity Recognition and Detection are gaining importance where it is and will be used in many upcoming innovations, not only in security and surveillance but also in understanding the behavioral patterns of humans. This project helps healthcare systems to monitor, analyze and interpret patient activities; In surveillance scenarios to track and monitor individuals, detect suspicious activities and threats in schools, colleges, libraries, parks, metro stations. This project can also be implemented in academics wherein the web camera captures the faces of the students and updates the attendance automatically. Its usage can also be extended in the field of research.

1.2 SCOPE

The scope of the project is to build a model that is able to recognize different types of human activities performed in the video clips which is trained during the model training phase. The designed model helps to overcome the disadvantage of the previously built models that relied on the use of sensors. With the elimination of sensors, we eliminate the cost associated with respect to the sensors and the errors that can be caused due to the use of sensors. This Project helps the healthcare systems to analyze and interpret patient activities; In surveillance scenarios to track and monitor individuals, detect suspicious activities and threats in schools, colleges, libraries, parks, metro stations.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

It is a challenging task to classify video clips based on human activity using sensors and It is a fundamental problem in computer vision.

Our goal is to detect/recognize human activity based on the input video clips provided by the user without any use of the sensors.

Example:

Problem: Use of sensors to record human activity patterns is expensive and needs necessity of frequent recharging of batteries.

Solution: Include deep learning technology to classify human activity based on video sequences.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

Paper Title	Conference Name and Year	Technology Used	Results	Inference
1 .Recognizing 50 Human Action Categories of Web Videos - UCF50 dataset	Machine Vision and Application Journal(MVAP), Sept-2012.	Dataset collection (consists of short video clips)	50 categories of video clips consists of different human activities.	Dataset can be used to train the model
2. Data Preprocessing and Network Building in CNN	Article by Tanya Dayanand, Aug-2020.	Normalization and standardization of video frames.	A set of frames converted to a similar sequence of resolution	Preprocessing takes about 50–70% of your time in most deep learning projects, and knowing some useful tricks will help in our project.
3. A CNN+LSTM Approach to Human Activity Recognition (IEEE)	Machine Vision and Application Journal(MVAP), Sept-2012.	Deep bidirectional LSTM (DB-LSTM) network	There is high probability where RNN may act unusual due to short term memory hence LSTM provides an upper-hand.	Since Sequence of image frames are involved CNN-LSTM is inferred.

4. An CNN-LSTM based Model for Video Classification - ConvLSTM (IEEE)	International Conference on Electronics, Information, and Communication (ICEIC), 2020	Deep bidirectional LSTM (DB-LSTM) network	A class of models that is both spatially and temporally deep, and has the flexibility to be applied to a variety of vision tasks involving sequential inputs and outputs.	Model that classifies video clips based on sequence of frames.
5. Long-term Recurrent Convolutional Networks for Visual Recognition and Description.	Computer Vision and Pattern Recognition Journal May 2016	Combination of both CNN and RNN.	Long-term Recurrent Convolutional Network (LRCN), which combines CNN and RNN layers in a single model.	It is very helpful to predict the action being performed in the video resulting in a robust model.

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

4.1 PROPOSED DESIGN

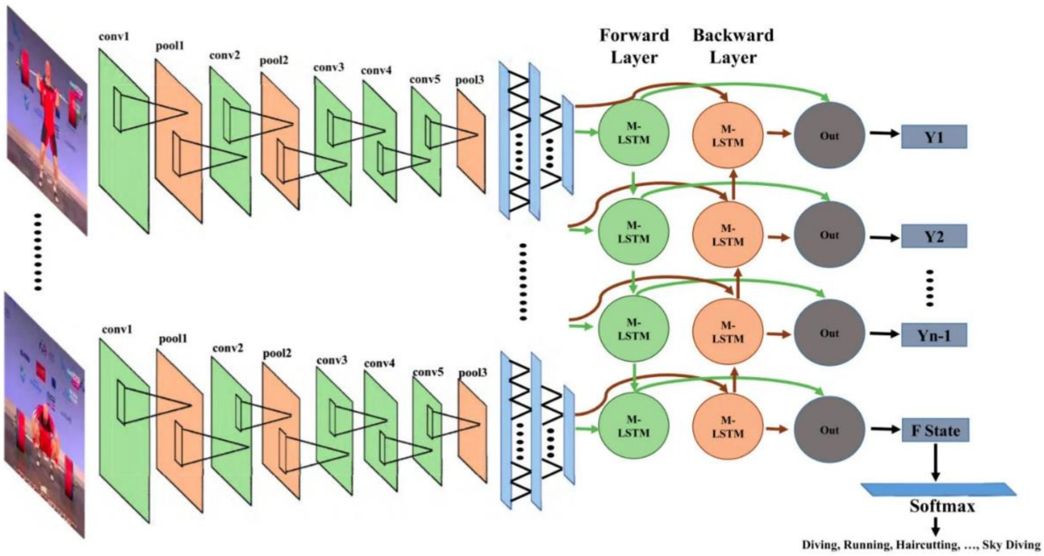

Figure 4.1(a) Workflow Diagram of video classification method

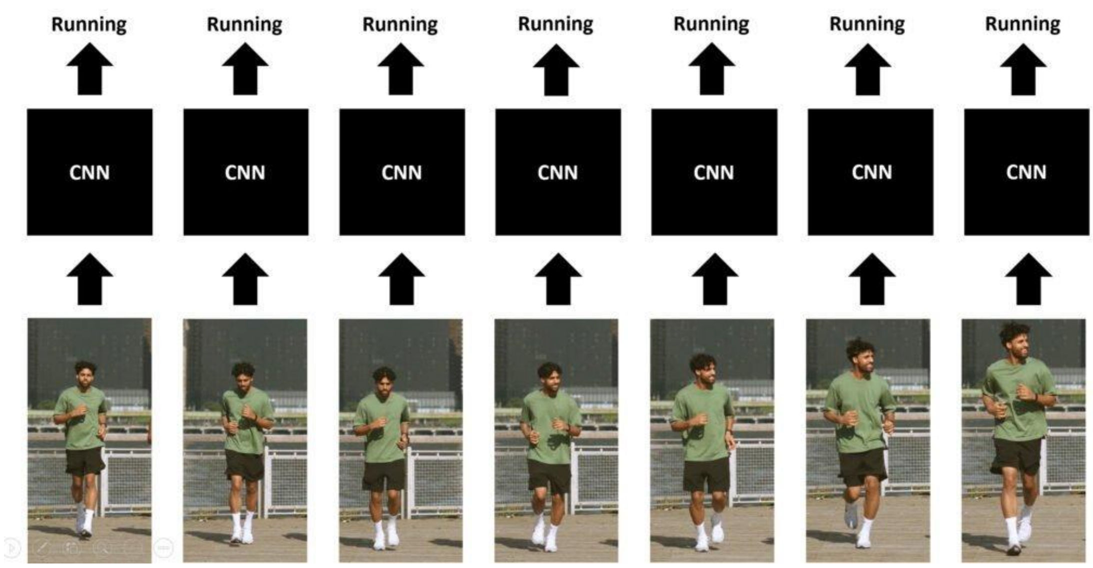

Fig 4.1(b) Classification of videos into several images under CNN

- Data collection from UCF50 - Action Recognition Data Set
- Visualize the Data with its Labels.
- Read and Pre-process the Dataset

- Create a Function for Dataset Creation for 10 classes which includes activities as follows:
  - WalkingWithDog
  - TaiChi
  - Swing
  - HorseRace
  - PushUps
  - Diving
  - Punch
  - Biking
  - Basketball
  - PlayingGuitar
- Extract frames from video clips.
- Resize & Normalize Frames.
- Split the Data into a Train and Test Set.
- Construct the ConvLSTM and LRCN Approach.
- Compile and Train the Model.
- Plot Model's Loss and Accuracy Curves.
- Save the model.
- Performing activity prediction on test video clips.

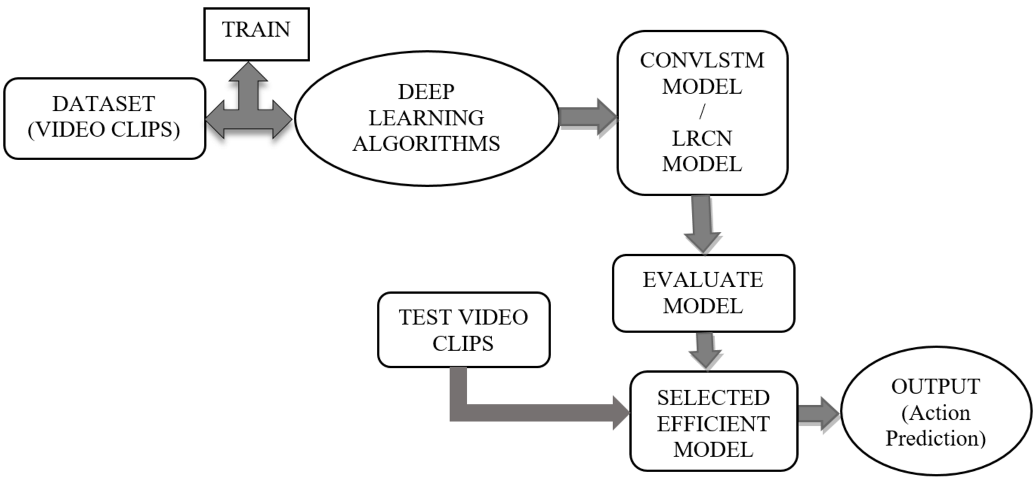

Fig 4.1(c) Workflow Diagram of Human Activity Detection Using Deep Learning

4.2 ASSUMPTIONS AND DEPENDENCIES

We have assumed that the trained model should have greater accuracy to avoid misclassification. The model selection is totally depending upon model performance and prediction score.

Another assumption is that the application is used on a computer with enough performance ability, and the use of an up-to-date internet browser. We assumed deep learning algorithm in order to solve this problem.

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

F1 - Collecting video clips from UCF50 dataset.

F2 - Visualize the dataset with labels.

F3 - Pre-process the Dataset.

- Create a Function for Dataset Creation for 10 classes which includes activities as follows:
    - WalkingWithDog
    - TaiChi
    - Swing
    - HorseRace
    - PushUps
    - Diving
    - Punch
    - Biking
    - Basketball
    - PlayingGuitar

F4 - Create a Function to Extract, Resize & Normalize Frames

F5 - Split the Data into Train and Test Set

F6 - Implement the ConvLSTM, LRCN Approach.

- Construct the model.
- Check Model's Structure.
- Compile & Train the Model.
- Plot Model's Loss and Accuracy curves.

F7 - Evaluating the trained model based on the loss function and saving the model.

F8 - Perform action recognition on test video.

5.2 NON-FUNCTIONAL REQUIREMENTS

5.2.1 Performance Requirements

The classification models built need to have a good accuracy in classifying the video clips on activities recognized.

The application needs to be able to respond to user inputs in a short span of time accurately and classify the video clips intelligently.

To build a model which can classify video clips of multiple classes requires huge amount of computation and GPU for achieving greater accuracy.

5.2.2 Software Quality Attributes

The built application must provide easy availability, correctness, flexibility, and usability to the user to achieve specific goals with effectiveness, efficiency, and satisfaction in a specified context of use.

5.3 SOFTWARE REQUIREMENTS

- Jupyter notebook
- Google Colab
- Python
- Scikit Learn
- TensorFlow
- Keras
- Matplotlib
- OpenCV

5.4 HARDWARE REQUIREMENTS

- Processor: Intel® Core™ i5-6200U cpu@2.30Ghz
- System type: 64-bit Operating system, x64-based processor
- Primary Storage: 8GB RAM
- Secondary Storage: 1TB
- GPU: NVidia GTX 1050

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

In this project, we will be using the ConvLSTM and LRCN Approach for the implemention.

Step 1 - Downloading and Visualizing the Data with its Labels

In the first step, we are going to download and visualize the data along with labels using the UCF50 – Action Recognition Dataset, which consists of various action categories such as swimming, boxing, Tai Chi, horse riding, etc. There are 25 groups of videos per action category with an average of 133 videos per action category and 199 frames per video.

We have selected 10 classes from UCF50 dataset which includes activities as follows:

- WalkingWithDog
- TaiChi
- Swing
- HorseRace
- PushUps
- Diving
- Punch
- Biking
- Basketball
- PlayingGuitar

Step 2 – Preprocessing the Dataset

Here, we will perform preprocessing on the dataset. Firstly, the video files will be read from the dataset. Then the frames of the videos will be resized to a fixed width and height in order to reduce the computations and normalize the data to range [0-1]. This is done by dividing the pixel values with 255, resulting in faster convergence during the model training.

Step 3 – Splitting the Data into Train and Test Set

For splitting the data into train and test set, we have the required features such as NumPy array containing all extracted frames of the videos and a one-hot-encoded-labels which is also a NumPy array containing all class labels in one-hot-encoded-label format. Now, the data is split into train and test sets. Also, the dataset is shuffled to avoid any bias or get splits representing the overall distribution of the data.

Step 4 – Implemented the ConvLSTM and LRCN Approach

In this step, we have implement the first approach by using a combination of ConvLSTM cells. A ConvLSTM cell is a variant of a LSTM network containing convolutions operations in the network. It is an LSTM with convolution embedded in the architecture that is capable of identifying spatial features of the data while keeping into account the temporal relation.

This approach is effective in video classification for capturing the spatial relation in the individual frames and temporal relation across different frames.

This step performs the following tasks:

- Constructing the Model

  To construct the model, we have used Keras ConvLSTM2D recurrent layers. The ConvLSTM2D layer also takes in the number of filters and kernel size required for applying the convolutional operations. The output of the layers is flattened in the end and is fed to the Dense layer with softmax activation which outputs the probability of each action category.

  We have used MaxPooling3D layers to reduce the dimensions of the frames and avoid unnecessary computations and Dropout layers to prevent overfitting the model on the data. The architecture is a simple one and has a small number of trainable parameters. This is because we are only dealing with a small subset of the dataset which does not require a large-scale model.
- Compiling and Training the Model

  Next, we have added an early stopping callback to prevent overfitting and start the training after compiling the model.
- Plotting Model's Loss and Accuracy Curves

  Now we have created a function plot_metric() to visualize the training and validation metrics. We already have separate metrics from our training and validation steps so now we just have to visualize them.

Step 5 -Test the Best Performing Model on YouTube videos

From the results, it seems that the LRCN model performed significantly well for a small number of classes. So in this step, we will put the LRCN model to test on some youtube videos.

- Create a Function to Download YouTube Videos
- We will create a function download_youtube_videos() to download the YouTube videos first using pafy library. The library only requires a URL to a video to download it along with its associated metadata like the title of the video.
- Download a Test Video
  Now we will utilize the function download_youtube_videos() created above to download a youtube video on which the LRCN model will be tested.
- Create a Function to Perform Action Recognition on Videos
  Next, we will create a function predict_on_video() that will simply read a video frame by frame from the path passed in as an argument and will perform action recognition on video and save the results.
- Perform Action Recognition on the Test Video
  Now we will utilize the function predict_on_video() created above to perform action recognition on the test video we had downloaded using the function download_youtube_videos() and display the output video with the predicted action overlayed on it.
- Create a Function to Perform a Single Prediction on Videos
  Now let's create a function that will perform a single prediction for the complete videos. We will extract evenly distributed N (SEQUENCE_LENGTH) frames from the entire video and pass them to the LRCN model. This approach is really useful when you are working with videos containing only one activity as it saves unnecessary computations and time in that scenario.
- Perform Single Prediction on a Test Video
  Now we will utilize the function predict_single_action() created above to perform a single prediction on a complete youtube test video that we will download using the function download_youtube_videos(), we had created above.

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

ConvLSTM Model Architecture:

In this step, we have implemented the first approach by using a combination of ConvLSTM cells. A ConvLSTM cell is a variant of an LSTM network that contains convolutions operations in the network. It is an LSTM with convolution embedded in the architecture, which makes it capable of identifying spatial features of the data while keeping into account the temporal relation.

For video classification, this approach effectively captures the spatial relation in the individual frames and the temporal relation across the different frames. As a result of this convolution structure, the ConvLSTM is capable of taking in 3-dimensional input (width, height, num_of_channels) whereas a simple LSTM only takes in 1-dimensional input hence an LSTM is incompatible for modeling Spatio-temporal data on its own.

To construct the model, we will use Keras ConvLSTM2D recurrent layers. The ConvLSTM2D layer also takes in the number of filters and kernel size required for applying the convolutional operations. The output of the layers is flattened in the end and is fed to the Dense layer with softmax activation which outputs the probability of each action category.

We will also use MaxPooling3D layers to reduce the dimensions of the frames and avoid unnecessary computations and Dropout layers to prevent overfitting the model on the data. The architecture is a simple one and has a small number of trainable parameters. This is because we are only dealing with a small subset of the dataset which does not require a large-scale model.

```
def create_convlstm_model():
# We will use a Sequential model for model construction
model = Sequential()
model.add(ConvLSTM2D(filters = 4, kernel_size = (3, 3), activation =
'tanh',data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True,
input_shape = (SEQUENCE_LENGTH,IMAGE_HEIGHT, IMAGE_WIDTH, 3))
model.add(MaxPooling3D(pool_size=(1,2,2),padding='same',data_format='channels_
last'))
```

```
model.add(TimeDistributed(Dropout(0.2)))
model.add(ConvLSTM2D(filters = 8, kernel_size = (3, 3), activation = 'tanh',
data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True))
model.add(MaxPooling3D(pool_size=(1, 2, 2), padding='same',
data_format='channels_last'))
model.add(TimeDistributed(Dropout(0.2)))
model.add(ConvLSTM2D(filters = 14, kernel_size = (3, 3), activation = 'tanh',
data_format = "channels_last",recurrent_dropout=0.2, return_sequences=True))
model.add(MaxPooling3D(pool_size=(1,2,2),padding='same',
data_format='channels_last'))
model.add(TimeDistributed(Dropout(0.2)))
model.add(ConvLSTM2D(filters = 16, kernel_size = (3, 3), activation = 'tanh',
data_format = "channels_last",recurrent_dropout=0.2, return_sequences=Tr
model.add(MaxPooling3D(pool_size=(1,2,2),padding='same',
data_format='channels_last'))
model.add(Flatten())
model.add(Dense(len(CLASSES_LIST), activation = "softmax"))
model.summary()
return model
```

LRCN Model Architecture:

In this step, we will implement the LRCN Approach by combining Convolution and LSTM layers in a single model. Another similar approach can be to use a CNN model and LSTM model trained separately. The CNN model can be used to extract spatial features from the frames in the video, and for this purpose, a pre-trained model can be used, that can be fine-tuned for the problem. And the LSTM model can then use the features extracted by CNN, to predict the action being performed in the video.

But here, we will implement another approach known as the Long-term Recurrent Convolutional Network (LRCN), which combines CNN and LSTM layers in a single model. The Convolutional layers are used for spatial feature extraction from the frames, and the extracted spatial features are fed to LSTM layer(s) at each time-steps for temporal sequence modeling. This way the network learns spatiotemporal features directly in an end-to-end training, resulting in a robust model.

We will also use a TimeDistributed wrapper layer, which allows applying the same layer to every frame of the video independently. So it makes a layer (around which it is wrapped) capable of taking input of shape (no_of_frames, width, height, num_of_channels) if originally the layer's input shape was (width, height, num_of_channels) which is very beneficial as it allows to input the whole video into the model in a single shot.

To implement our LRCN architecture, we will use time-distributed Conv2D layers which will be followed by MaxPooling2D and Dropout layers. The feature extracted from the Conv2D layers will be then flattened using the Flatten layer and will be fed to a LSTM layer. The Dense layer with softmax activation will then use the output from the LSTM layer to predict the action being performed.

```
def create_LRCN_model():
model = Sequential()
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'))
model.summary()
return model
```

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

We have visualized the data with its labels by randomly picking videos from selected category and visualized the first frame of the selected videos.

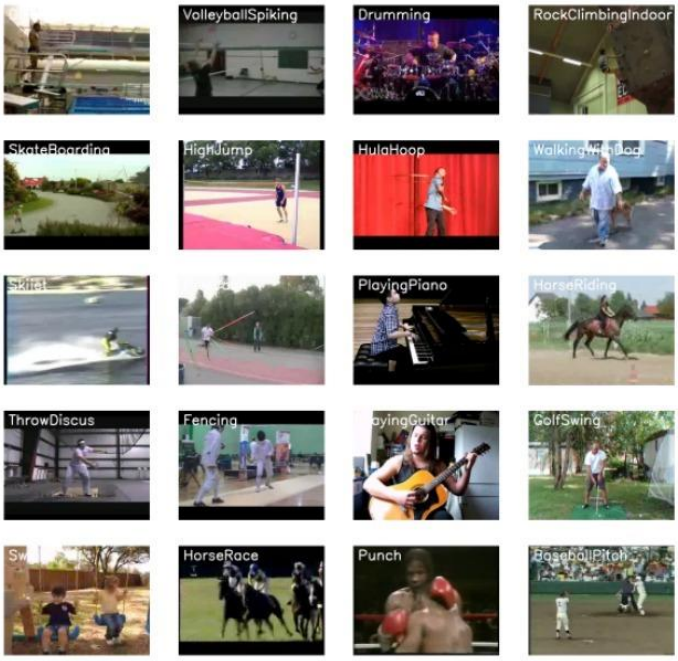

Fig 8. Data visualization with labels

We have successfully resized the frames of video to a fixed height and width (64 X 64) to reduce the computations and normalized the data which makes convergence faster while training the network.

Fig 8.1 BEFORE FRAME RESIZING (320 X 240) **Fig 8.2 AFTER FRAME RESIZING (64 X 64)**

Visualizing the training and validation metrics of the LRCN model.

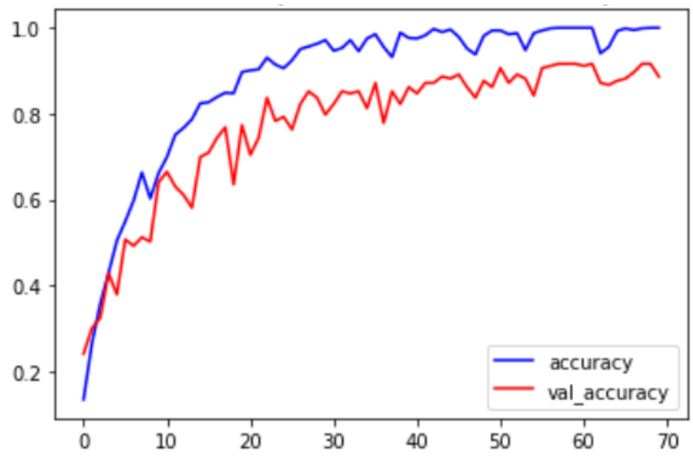

Fig 8.3 Total Accuracy VS Total Validation Accuracy

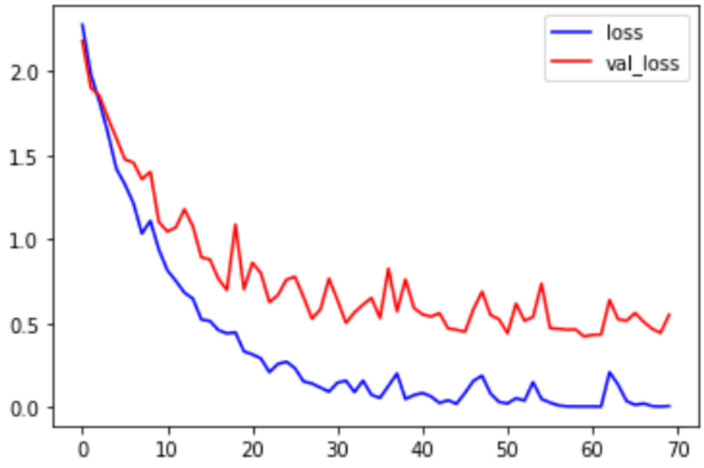

Fig 8.4 Total Loss VS Total Validation Loss

ConvLSTM Model	LRCN Model
Validation Accuracy: **0.6855**	Validation Accuracy: **0.8843**
Validation Loss: **1.0609**	Validation Loss: **0.4904**

Table 8. Comparison between ConvLSTM and LRCN

Result Screenshots

Case 1: An input test video classified as "WalkingWithDog".

Case 2: An input test video classified as "PlayingGuitar".

Case 3: An input test video classified as "Biking".

Case 4: An input test video classified as "TaiChi".

Case 5: An input test video classified as “Swing”.

Result Evaluation

Name of the Class	Accuracy
WalkingwithDog	0.998834
PlayingGuitar	0.999835
Biking	0.999183
Taichi	0.998823
Swing	0.999132

Table 8.1. Evaluation of Results

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

CHAPTER 9 CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION

Human Activity detection systems are a large field of research and development, currently with a focus on advanced deep learning algorithms, innovations in the field of hardware architecture, and on decreasing the costs of monitoring while increasing safety. We learnt a lot about this developing field of study while researching and developing this project. During the development of this project to detect human activities from video feeds we came across multiple approaches to help detect a set of activities. We had limited system hardware resources like GPU for training models and we had only a conservative amount of data for the given set of activities.

9.2 FUTURE SCOPE

In future we can do this implementation with greater processing resources and with greater amount of data to increase the accuracy further. We have proposed to add multiple new activities for detection to our currently developed model. As we all know that the accuracy of the models can never be 100% so the results that are generated will always have a possibility of providing improper output as is the case with all deep learning so if we can increase the data to improve the overall efficiency of detection of human activity we can do so in future models where a huge data set is used to train the developed models. Other future enhancements can include the use of IOT based smart devices that can perform pre-programmed activities based on actions performed and provide automated solutions to simple problems.

REFERENCES

[1] Understanding of a convolutional neural network (IEEE)
International Conference on Engineering and Technology (ICET), 2017.

[2] Recognizing 50 Human Action Categories of Web Videos Kishore K. Reddy, Mubarak Shah.

[3] A CNN-LSTM Approach to Human Activity Recognition (IEEE) International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2020.

[4] Long-term recurrent convolutional networks for visual recognition and description (IEEE), Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[5] Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting by Xingjian Shi (NIPS 2015)

[6] Long-term Recurrent Convolutional Networks for Visual Recognition and Description by Jeff Donahue (CVPR 2015)

[7] An CNN-LSTM based Model for Video Classification - ConvLSTM (IEEE) International Conference on Electronics, Information, and Communication (ICEIC), 2020

[8] Data Preprocessing and Network Building in CNN, International conference on Data preprocessing and Network building(ICDP),
Aug-2020.

APPENDIX A

OPENCV

OpenCV is the huge open-source library for the computer vision, by using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

APPENDIX B

UCF50

UCF50 is an action recognition data set with 50 action categories, consisting of realistic videos. Our data set is very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. For all the 50 categories, the videos are grouped into 25 groups, where each group consists of more than 4 action clips. The video clips in the same group may share some common features, such as the same person, similar background, similar viewpoint, and so on.

APPENDIX C

CNN

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network, we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

FUNDING AND PUBLISHED PAPER DETAILS

We have successfully published a paper entitled “Human Activity Detection Using Deep Learning” in International Journal of Scientific Research and Engineering Trends(IJSRET), in Volume 8, Issue 02, Mar-Apr-22, Page 1009-1013.

Below are the certificates that we have received from the International Journal of Scientific Research and Engineering Trends(IJSRET).

IJSRET

International Journal of Scientific Research and Engineering Trends (IJSRET)

This is to certify that
David Antony Selvaraj A
has published a paper entitled
“Human Activity Detection Using Deep Learning”
in International Journal of Scientific Research and Engineering Trends, in Volume 08, Issue 02, Mar-Apr-22, Page 1009-1013

Editor-in-Chief
IJSRET
www.ijsret.com
ISSN : 2395-566X


International Journal of Scientific
Research and Engineering Trends
(IJSRET)
This is to certify that
Adarsh SM
has published a paper entitled
"Human Activity Detection Using Deep Learning"
in International Journal of Scientific Research
and Engineering Trends, in Volume 08, Issue 02,
Mar-Apr-22, Page 1009-1013
Editor-in-Chief
IJSRET
www.ijsret.com
ISSN : 2395-566X


International Journal of Scientific
Research and Engineering Trends
(IJSRET)
This is to certify that
Gagan GR
has published a paper entitled
"Human Activity Detection Using Deep Learning"
in International Journal of Scientific Research
and Engineering Trends, in Volume 08, Issue 02,
Mar-Apr-22, Page 1009-1013
Editor-in-Chief
IJSRET
www.ijsret.com
ISSN : 2395-566X

International Journal of Scientific Research and Engineering Trends (IJSRET)

This is to certify that
Karthik B
has published a paper entitled
"Human Activity Detection Using Deep Learning"
in International Journal of Scientific Research and Engineering Trends, in Volume 08, Issue 02, Mar-Apr-22, Page 1009-1013

Editor-in-Chief
IJSRET
www.ijsret.com
ISSN : 2395-566X

International Journal of Scientific Research and Engineering Trends (IJSRET)

This is to certify that
Ramesh Roshan M
has published a paper entitled
"Human Activity Detection Using Deep Learning"
in International Journal of Scientific Research and Engineering Trends, in Volume 08, Issue 02, Mar-Apr-22, Page 1009-1013

Editor-in-Chief
IJSRET
www.ijsret.com
ISSN : 2395-566X