

A
PROJECT REPORT
ON
Human Action Recognition on
Common Benchmark

SUBMITTED TO
SHIVAJIUNIVERSITY, KOLHAPUR
IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF
DEGREE BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND
ENGINEERING

SUBMITTED BY:

MISS.	GOURI VIJAY SONAVANE	19UCS127
MR.	PRAKHAR VINOD UPADHYAY	19UCS133
MR.	TEJAS RAGHUNATH YELAVKAR	19UCS137
MR.	GOURAV SANJAY SHINDE	19UCS124
MR.	SUYASH SANJAY CHOUGULE	17UCS52007XX

UNDER THE GUIDANCE OF

Prof. V. V. Kheradkar



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DKTE SOCIETY'S TEXTILE AND ENGINEERING
INSTITUTE, ICHALKARANJI
2022-2023

D.K.T.E.SOCIETY'S
TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN AUTONOMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that, project work entitled

Human Action Recognition on Common Benchmark

is a bonafide record of project work carried out in this college by

MISS. GOURI VIJAY SONAVANE	19UCS127
MR. PRAKHAR VINOD UPADHYAY	19UCS133
MR. TEJAS RAGHUNATH YELAVKAR	19UCS137
MR. GOURAV SANJAY SHINDE	19UCS124
MR. SUYASH SANJAY CHOUGULE	17UCS52007XX

is in the partial fulfillment of award of degree Bachelor in Engineering in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2022-2023.

V.V. Kheradkar
(PROJECT GUIDE)

PROF.(DR.) D.V. KODAVADE
(HOD CSE DEPT.)

PROF.(DR.) MRS. L.S. ADMUTHE
(DIRECTOR)

EXAMINER: _____

DECLARATION

We hereby declare that, the project work report entitled “Human Action Recognition on Common Benchmark” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.E.(CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

Miss. Gouri Vijay Sonawane	19UCS127
Mr. Prakhar Vinod Upadhyay	19UCS133
Mr. Tejas Raghunath Yelavkar	19UCS137
Mr. Gourav Sanjay Shinde	19UCS124
Mr. Suyash Sanjay Shinde	17UCS52007XX

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to Prof. V.V.Kheradkar for his valuable guidance, support and encouragement in completion of this project report.

Also, we would like to take opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thank you,

Miss. Gouri Vijay Sonawane	19UCS127
Mr. Prakhar Vinod Upadhyay	19UCS133
Mr. Tejas Raghunath Yelavkar	19UCS137
Mr. Gourav Sanjay Shinde	19UCS124
Mr. Suyash Sanjay Shinde	17UCS52007XX

ABSTRACT

Human cognitive processing has become an important area of research in computer vision, with important applications in analysis, human computer interaction, and body control. Recognizing and understanding human actions from video sequences is a challenge because of the variability of imagery, movement, emotion, and competition. To solve this problem, many benchmarks have been developed as a valuable resource to evaluate and compare the results of recognition algorithms.

This report presents an analysis of human knowledge of similar patterns. We review many commonly used documents, including the UCF50, to provide a comprehensive overview of state-of-art systems and accreditation processes. This examines the evolution of technology over time and identify key trends in the field.

The analysis of this report includes analysis of various extraction properties, including artificial intelligence and deep learning features such as convolutional neural networks (CNN). This report also explores the effectiveness of combining RCN with LSTMs (Long Short Term Memory) and analyze its impact on cognitive performance

Through this analysis, the aim is to provide researchers and practitioners with an overview of the current state of human standard acceptance. By understanding the strengths and limitations of existing systems, the hope is to further develop the study of knowledge and contribute to the development of what is correct, available, powerful and useful for practical use.

INDEX

1. Introduction	01
1.1 Problem Definition	04
1.2 Aim and objective of the project	04
1.3 Scope and limitation of the project	05
1.4 Timeline of the project	06
1.5 Project Management Plan	07
1.6 Project Cost	08
2 Background study and literature overview	09
2.1 Literature overview	09
3 Requirement analysis	13
3.1 Requirement Gathering	13
3.2 Requirement Specification	15
3.3 Use case Diagram	16
3.4 Project Costing	17
4 System design	18
4.1 Architectural Design	18
4.2 User Interface Design	18
4.3 Algorithmic description of each module	19
4.4 System Diagram	20
4.4.1 Dataflow Diagram	20
4.4.2 Sequence Diagram	21
4.4.3 Activity Diagram	22
4.4.4 Component Diagram	23
5 Implementation	24
5.1 Environmental Setting for Running the Project	24
5.2 Detailed Description of Methods	25
5.3 Implementation Details	27
6 Integration and Testing	33
6.1 Description of the Integration Modules	33
6.2 Testing	35
7 Performance Analysis	38
8 Future Scope	40
9 Applications	41
10 Installation Guide and User Manual	42
11 Plagiarism Report	45
12 Ethics	50

1.INTRODUCTION

Computer vision is a branch of computer science that focuses on changing some of the complexities of human vision, enabling computers to recognize and process objects in images and videos like humans. Thanks to advances in intelligence and innovations in deep learning and neural networks, this field has prospered in recent years, outpacing humans in certain tasks related to search and recording.

In applications that relates to Action recognition, it is important that machines can comprehend and recognize the action performed and use the result for the judging the severity of the Action. This project offers an end-to-end neural network model that determines the action performed by the user to gain insights from it.

So, the first task is to acquire the dataset that matches with the Requirements mentioned in the SRS report. Then one can fine-tune the dataset so that it can incorporate other actions that the stakeholder requires. This load dataset is then partitioned into two parts one is for training and another one is for testing. The video that needs to be monitored to detect action from it needs to be segmented. Segmentation is the process of partitioning a video sequence into a disjoint set of consecutive frames. The incoming video is segmented further to extract key features from it. Using Feature Selection and Feature Extraction the critical features of the frame can be restored. In feature extraction the system transforms the arbitrary data into numerical data without losing the data, feature selection selects the relevant data and eliminates noisy data. The frames thus generated needs to be trained using the appropriate algorithm, so that the model developed achieves the overall Objective of the project. Object detection detects the action performed by the human using the LRCN algorithm. The algorithm aims to learn viewpoint invariant representation for action recognition and action detection.

The LRCN algorithm, also known as the LongTerm Recurrent Convolutional Network, is a deep learning algorithm that combines the power of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to process and understand data connectivity, especially in the context of video analysis.

Here's a high-level overview of how the LRCN algorithm works:

Convolutional Neural Network (CNN) layer: The LRCN first processes each frame of the video through a CNN designed to extract the agreed-upon features. CNN layers help capture spatial information and identify important objects, shapes, and patterns in each frame.

Advanced Modeling Using Recurrent Neural Networks (RNN): When a frame is encoded by the CNN, the extracted data is transmitted to the RNN layer. RNN takes into account the temporal relationship between frames and can capture long-term dependencies in videos. In LRCNs, LSTM (Long Short Term Memory) differs from the commonly used RNN in that it can work well for long periods of time.

Video classification or recording: The results of the RNN layer can be used for different tasks depending on the purpose. For video classification, the softmax method is usually added on top of the RNN layer to predict the class name or category of the video. This allows LRCN to recognize videos and classify them into different categories.

The LRCN algorithm can be trained end-to-end using backpropagation through time. During training, the model is presented with labeled video data, and the parameters of both the CNN and RNN layers are optimized to minimize the loss function.

Overall, the LRCN algorithm combines the strengths of CNNs in spatial feature extraction and RNNs in capturing temporal dynamics to perform video analysis tasks such as classification and captioning. It has shown promising results in various video-related applications, including action recognition, video summarization, and video captioning.

After the dataset is loaded into the system, the dataset passes through various rounds of training. Referred as **Epoch**. In the context of machine learning and deep learning, an epoch refers to a complete pass through the entire training dataset during the training phase of a model. It is an important concept related to the iteration and optimization process of training a model.

During the training phase, the dataset is divided into smaller batches, and each batch is fed to the model for forward propagation, followed by the computation of the loss and subsequent backpropagation to update the model's parameters. An epoch is completed when all batches in the training dataset have been processed once.

Here's a step-by-step overview of the training process within an epoch:

1. **Forward Propagation:** Each batch of training data is fed through the model, and the inputs are processed to generate predictions or output values.
2. **Loss Computation:** The model's predictions are compared to the corresponding ground truth labels in the batch, and a loss function is calculated. The loss function measures the dissimilarity between the predicted values and the actual values.
3. **Backpropagation:** The gradients of the loss with respect to the model's parameters are computed using the backpropagation algorithm. These gradients indicate the direction and magnitude of the parameter updates necessary to minimize the loss.
4. **Parameter Update:** The model's parameters (weights and biases) are adjusted based on the computed gradients. This update step is typically performed using an optimization algorithm such as stochastic gradient descent (SGD) or one of its variants.
5. **Repeat for all Batches:** Steps 1 to 4 are repeated for all batches in the training dataset, allowing the model to learn from different subsets of the data.

Once all the batches have been processed, one epoch is completed. The number of epochs determines how many times the model will iterate over the entire training dataset. Training for more epochs allows the model to further refine its parameters and improve its performance, as it gets exposed to the entire dataset multiple times.

The choice of the number of epochs depends on various factors, including the complexity of the problem, the size of the dataset, and the convergence behavior of the model. It is often determined through experimentation and monitoring the model's performance on a separate validation dataset. Early stopping techniques can also be employed to automatically stop training when the model's performance plateaus or starts to degrade, thus preventing overfitting.

Once the training phase of the model is completed, the number of frames generated can show different actions that is spread across various frames. Thus, it becomes important that the action which is repeated maximum number of times in the frames is prioritized. This is where a **Softmax** Function comes into the picture.

Action recognition is typically a multi-class classification problem, where a video can belong to one of several action classes. The Softmax function is well-suited for multi-class classification tasks as it transforms the model's output into a valid probability distribution. It ensures that the predicted probabilities for each class sum up to 1, making it easier to compare and rank different action categories.

1.1) Problem Definition

Human activity recognition (HAR) is the problem of identifying physical activities performed by humans based on their movements in a particular environment. Keeping a tab on the number of actions performed the humans can help us better prepare in case of any untoward situation.

1.2) Aim and Objective of the project

Aim:

Human Action Recognition on Common Benchmarks aims to enable the construction of an intelligent system that will recognize the action from video streams, with an intent to detect the action which is performed and then develop a system which can classify the actions into various categories based on its effects.

The video input will be segmented and these segmented frames will help in deciphering the action. The main aim of the project is to decipher the action that is performed by the human so that one can judge the intention behind the action.

Objectives:

- 1) To Fine-tune the UCF50 Dataset with different actions.
- 2) To resize and normalize the input video in order to make it suitable to be fed as an input
- 3) To train and develop the model using the algorithm that results in high accuracy and low loss function.
- 4) The model which is build needs to recognize the performed action and label the action onto the video frame accordingly.

1.3) Scope and Limitation of the Project

Scope:

a) Surveillance Cameras:-

Cameras installed in public areas such as banks, airports, hospitals, etc. This helps in activity detection of objects to monitor suspicious activities for real-time reactions like stealing etc.

For example, if the surveillance camera is present in the bank area then it monitors the people in the area. If any action of the person is found suspicious. Then it will alert the administrator.

b) Sports play analysis:-

Analyzing the play and deducing the action in the sport. The impact of action prediction in professional sports is gaining ground. Analyzing the next move based on gestures made by the opponent is a critical aspect in today's Game analysis and its importance is felt in every sport .

Limitation:

1. The video sample should be real-world related. Where only one positive sample is provided at a time. There should not be any forking path or more than one positive case.
2. Input video should be of 64*64 resolution.
3. Video link is needed.
4. Video must be of maximum 30 seconds.

1.4) Timeline of the Project

No. of Weeks	Work Done
Week 1	Define Project Topic, Scope, Objectives, and Requirements
Week 2	Conduct Research and Feasibility test of the Project
Week 3	Finalize Project plan and create detailed Project Schedule
Week 4	Initiate the process of preparing the SRS Report
Week 5	Initial Presentation of the Project
Week 6-7	Research through various Algorithms for developing the Model
Week 8-9	Finalize the Algorithm and Start the Development Phase
Week 10-11	Finalize the Dataset to be fed to train the model
Week 12	Start of with importing Important libraries and go forward with Training the model
Week 13	Second Presentation of showing code.
Week 14	Focus on Preparing key Diagrams that describe the process of the project
Week 15	Pre-Final/Internal Exam Presentation of the project
Week 16-18	Using the LRCN algorithm extract key features from the dataset and decide the actions that needs to be recognized
Week 19-21	Continuously Monitor the Accuracy and Loss function of the model and hit and trial with the epoch, keeping in mind the threat of overfitting
Week 22-25	Test the model with different videos that do not exceed the 30s range and note the performance of the model
Week 26-27	Prepare the Required Report with Holistic View of the Project and with required results.

1.5) Project Management Plan:

Project initiation:

Define the project objectives, scope, and deliverables.
Identify the stakeholders and their requirements.
Create a project charter and gain approval from the stakeholders.
Form a project team with the necessary skills and expertise.

Planning:

Develop a project management plan that outlines the project approach, timeline, budget, and resource allocation.
Conduct a risk assessment and develop a risk management plan.
Define the project requirements, including the AI algorithms, natural language processing, etc.
Create a detailed project schedule and task list.
Identify the technical and infrastructure requirements, including hardware, software, and data storage.
Establish a communication plan to ensure that stakeholders are kept informed of progress and changes.

Execution:

Finalize and develop the AI algorithms and natural language processing capabilities.
Develop the ML model using the dataset .
Use of Early Stopping function to avoid Overfitting.
Efficient use of number of epoch and controlling the batch size.

Monitoring and control:

Monitor progress against the project schedule and budget.
Manage risks and issues as they arise.
Conduct regular testing and quality assurance to ensure that the virtual assistant continues to meet the project requirements.
Report on project status and progress to stakeholders.
Manage changes to the project scope, timeline, and budget as necessary.

Project closure:

Obtain sign-off from stakeholders that the project objectives have been met.
Archive project documentation and data.
Conduct a post-project review to identify lessons learned and areas for improvement.
Release the virtual assistant to users, if applicable.

1.6) Project Cost:

COCOMO Model:

The COCOMO (Constructive Cost Model) is a regression model used to estimate the effort and development time for software projects. It is based on the size of the software product, measured in Kilo Lines of Code (KLOC). For an Embedded project with 1400 Lines of Code (LOC), we first convert LOC to KLOC by dividing by 1000:

$$\text{KLOC} = 1400 / 1000 = 1.4$$

Then we use the Basic COCOMO model equations for effort and development time:

$$\text{Effort} = a * (\text{KLOC})^b \quad \text{Development Time} = c * (\text{Effort})^d$$

For an Embedded project, the values of the constants a, b, c, and d are:

$$a = 3.6$$

$$b = 1.20$$

$$c = 2.5$$

$$d = 0.56$$

Substituting these values and the calculated KLOC value into the equations, we get: Effort = $3.6 * (1.4)^{1.20} = 5.29$ person-months Development Time = $2.5 * (5.29)^{0.56} = 6.35$ months So according to the Basic COCOMO model, our Embedded project with 1400 LOC would require an **estimated development time of 6.35 months**.

These are rough estimates and actual values may vary depending on various factors specific to our project.

2) Background study and literature overview:

2.1) Literature Overview:

1. "Two-Stream Convolutional Networks for Action Recognition in Videos" by Karen Simonyan and Andrew Zisserman (2014):

- This paper introduced the two-stream convolutional networks, consisting of spatial and temporal streams, for action recognition. The spatial stream utilizes frame-level appearance information, while the temporal stream captures motion information

2. "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition" by Limin Wang et al. (2016):

- This work proposed the Temporal Segment Networks (TSN) architecture, which samples multiple snippets from a video to model temporal dynamics effectively. TSN achieved state-of-the-art results on several benchmark datasets.

3. "I3D: Incremental 3D Network for Action Recognition" by Joao Carreira and Andrew Zisserman (2017):

- The I3D architecture extended 2D CNNs to 3D CNNs by inflating the 2D filters along the temporal dimension. It achieved state-of-the-art performance on various action recognition benchmarks, including Kinetics and UCF101.

4. "BERT for Action Recognition" by Heng Wang et al. (2020):

- This work adapted the Bidirectional Encoder Representations from Transformers (BERT) model for video action recognition. By leveraging pretraining and self-supervised learning, it achieved state-of-the-art results on various benchmarks.

5. "Prognosing Human Activity Using Actions Forecast and Structured Database" by Vibekananda Dutta and Teresa Zielinska (2020):

- This paper introduces a method for prognosing human activity by utilizing action forecasting and a structured database, enabling accurate predictions of ongoing activities based on completed action sequences and knowledge stored in the database.

6. “Long-Term Trajectory Prediction of the Human Hand and Duration Estimation of the Human Action” by Yujiao Cheng And Masayoshi Tomizuka (2021):

- This paper presents a method for long-term trajectory prediction of the human hand and duration estimation of human actions, enabling accurate anticipation of hand movements and action durations in human-robot collaborative tasks.

7. “Peeking into the Future: Predicting Future Person Activities and Locations in Videos” by Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander Hauptmann And Li Fei-Fei (2020):

- This paper presents a novel approach for predicting future person activities and locations in videos, enabling the ability to anticipate human behavior and movements. The proposed method utilizes spatio-temporal modeling and deep learning techniques to forecast future actions and spatial trajectories, demonstrating promising results in predicting future person activities and locations.

8. “Human Activity Prediction: Early Recognition of Ongoing Activities from Streaming Videos” by M.S. Ryoo (2012):

- This paper presents a method for early recognition of ongoing activities from streaming videos, enabling real-time prediction of human activity. The proposed approach leverages deep learning techniques and temporal modeling to accurately anticipate ongoing activities, demonstrating effective results in predicting human actions from streaming video data.

9. “SimAug: Learning Robust Representations from Simulation for Trajectory Prediction” by Junwei Liang, Lu Jiang and Alexander Hauptmann:

- This paper presents SimAug, a method for improving trajectory prediction by learning robust representations from simulation data, combining real and simulated data to enhance model generalization and accuracy.

10. “Human Activity Recognition and Prediction” by David Jardim, Luís Miguel Nunes, and Miguel Sales Dias:

- This paper focuses on human activity recognition and prediction, presenting a comprehensive approach for accurately recognizing and forecasting human activities based on machine learning techniques and sensor data.

11. “From Recognition to Prediction: Analysis of Human Action and Trajectory Prediction in Video” by Junwei Liang:

- This paper investigates the transition from recognition to prediction by analyzing human action and trajectory prediction in videos, aiming to improve the understanding and anticipation of human behavior and movements.

12. “Human activity recognition: A review” by Ong Chin Ann And Bee Theng Lau (2015):

- This paper provides a comprehensive review of human activity recognition techniques, highlighting the advancements and challenges in the field, and offering insights into the various methodologies and applications in human activity recognition.

13. “Human Action Recognition and Prediction: A Survey” by Yu Kong And Yun Fu”:

- This paper presents a survey on human action recognition and prediction, covering various methodologies and techniques used in the field, providing an overview of the advancements and challenges in the area of recognizing and forecasting human actions.

14. “Long Term and Key Intentions for Trajectory Prediction” by Harshayu Girase , Haiming Gang, Srikanth Malla, Jiachen Li ,Akira Kanehara, Karttikeya Mangalam And Chiho Choi (2021) :

- This paper addresses long-term trajectory prediction by incorporating key intentions, providing insights into human behavior for accurate and reliable forecasting of future trajectories.

15. “Cross-Domain Human Action Recognition” by Wei Bian, Dacheng Tao And Yong Rui:

- This paper focuses on cross-domain human action recognition, exploring techniques and approaches to recognize human actions across different domains, enabling improved understanding and analysis of human behavior across diverse datasets.

16. “Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting” by Yan Bin Ng And Basura Fernando:

- This paper introduces a new approach to weakly supervised action forecasting, utilizing attention mechanisms to forecast future action sequences, enabling accurate predictions of future actions with limited supervision.

17. “Prediction of Human Activity by Discovering Temporal Sequence Patterns” by Kang Li E And Yun Fu:

- This paper presents a method for predicting human activity by discovering temporal sequence patterns, enabling accurate anticipation of future activities based on learned patterns and temporal dependencies in the data.

These papers represent a selection of influential works in the field of human action recognition on common benchmark datasets. They showcase various approaches, including two-stream networks, trajectory-based methods, 3D CNNs, attention mechanisms, and leveraging pretraining techniques, all aimed at improving action recognition performance.

3.Requirement Analysis

3.1) Requirement Gathering:

Gathering requirements for human recognition of a measure using the LRCN (LongTerm Recurrent Convolutional Network) algorithm will include the identification of specific needs, constraints, and goals. Here are some key steps and considerations for aggregation:

1. Define Scope: Clearly define the scope of the project, including any specifications that need to be implemented, the type of human behavior should be defined and a performance measure or criterion set.

2. Analyze data: Analyze data from training and assessment. Reference standards for validation include UCF50, among others. Decide whether to use publicly available information or to collect and document it yourself.

3. Data Preprocessing: Determine the necessary preliminary steps for data entry. This can include operations such as resizing/rescaling videos, removing frames, normalizing pixel values, and increasing datasets with techniques such as randomly cropping, flipping, or adding noise.

4. Selection Algorithm: Suggest using LRCN algorithm for authentication. LRCN combines a convolutional neural network (CNN) for spatial feature extraction and a recurrent neural network (RNN) to capture physical interactions in action sequences. Specify changes or modifications to the basic LRCN model based on your needs.

5. Performance Requirements: Specify the performance requirements for the recognized performance. This may include metrics such as accuracy, precision, recall, LossFunction, or special rules regarding the efficiency of time or the performance of the calculation.

6. Hardware and Software Requirements: Define the hardware and software infrastructure required to train and deliver an operational experience. Consider computing resources, memory requirements, and the specific software or libraries required, such as TensorFlow, PyTorch, or Keras.

7. User interface and integration: Decide whether the system will have a user interface to interact with the authentication results or whether it will be integrated into the existing platform or system. Specify special requirements for real-time processing, data flow, or offline processing.

8. Evaluation and Evaluation: Identify evaluation methods and evidence systems to measure performance. Determine how data will be divided into training, validation, and testing metho and specify the competitor or hash methods used.

9. Documentation and Maintenance: Plan for appropriate system documentation, including algorithms, data preprocessing steps, training process, and any soft ware code or script. Consider future maintenance and possible system updates.

By following these steps and considering these scenarios, you can write the necessary rules to create a human recognition algorithm using the LRCN algorithm on a shared data rate.

3.2) Requirement Specification

1.Functional Requirements:

- The system must be able to read the video as input
- The system must be able to extract each frame from the video input for processing
- The system must be able to move forward and change, or the desired size of the crop extracted from the input
- The system must be able to compare the frame with the learned weight.
- After comparison, the system should be able to classify the input order within each group with accuracy.

2.Nonfunctional Requirements:

a) Security

Do not allow third parties to modify content without permission.

b) Availability

- Self-study support should be available.
- The system should be smart enough to suggest appropriate steps as you continue to use the system.
- The system should be able to recognize the many tasks humans can do.
- There should be no limit to the types of input video streams the system can handle.

c)Reliability

- The system should be able to recover itself in a timely manner.
- The system must be able to handle all exceptions

3.3) Use Case Diagram

Admin –

Load Test data
Train the model
Test the Performance
Deploy Model
Action Recognition

End User –

Alert Generation

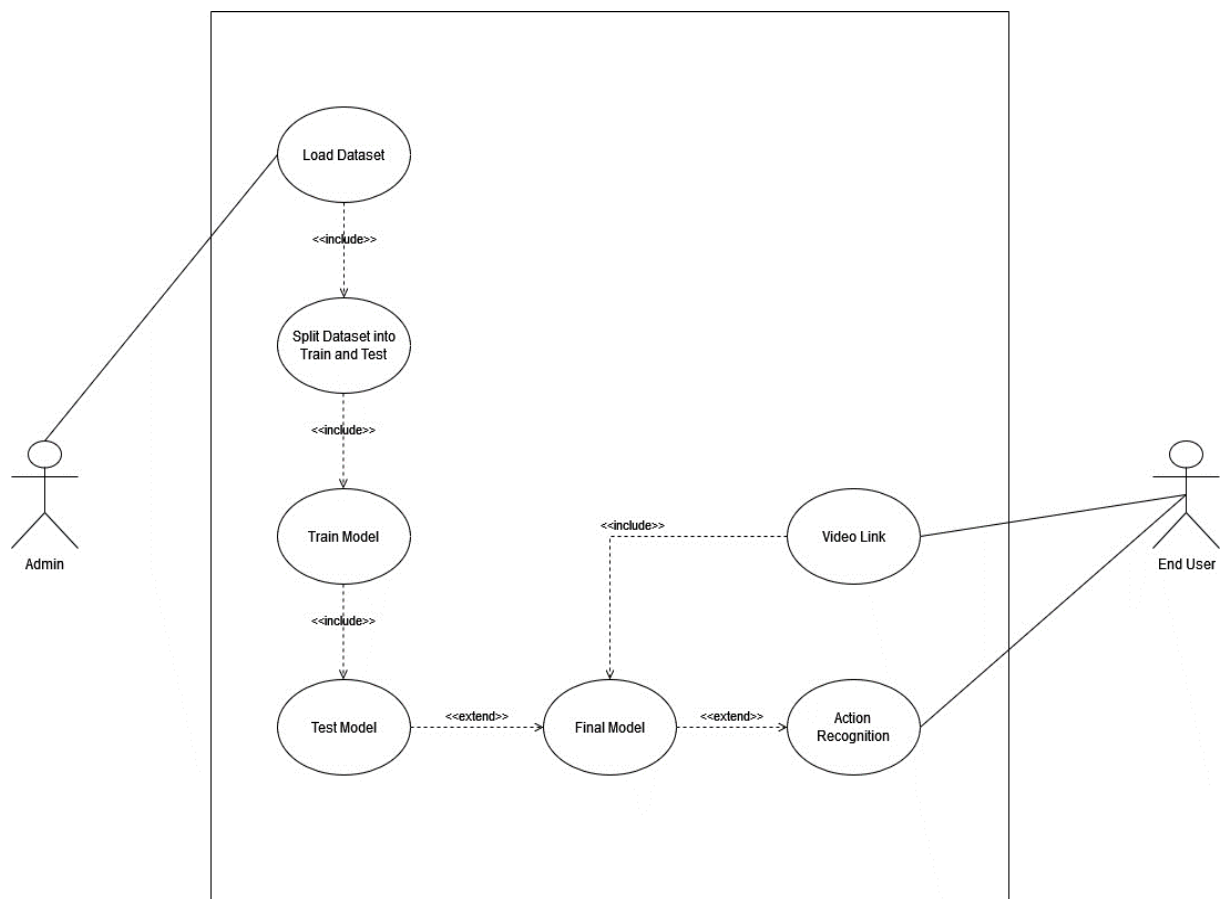


Figure 3.1 Use Case Diagram

3.4) Project Costing:

Hardware/Software	Cost
Computer system with i5 10 th generation or above/8GB or above RAM/128GB SSD	50000
NVDIA 1650 GPU	10000
Python IDE to run machine learning Modules	Open Source
Python 2.7; Tensorflow = 1.10.0	Open Source
Electricity	560
Internet	3000

4.System Design

4.1) Architectural Design

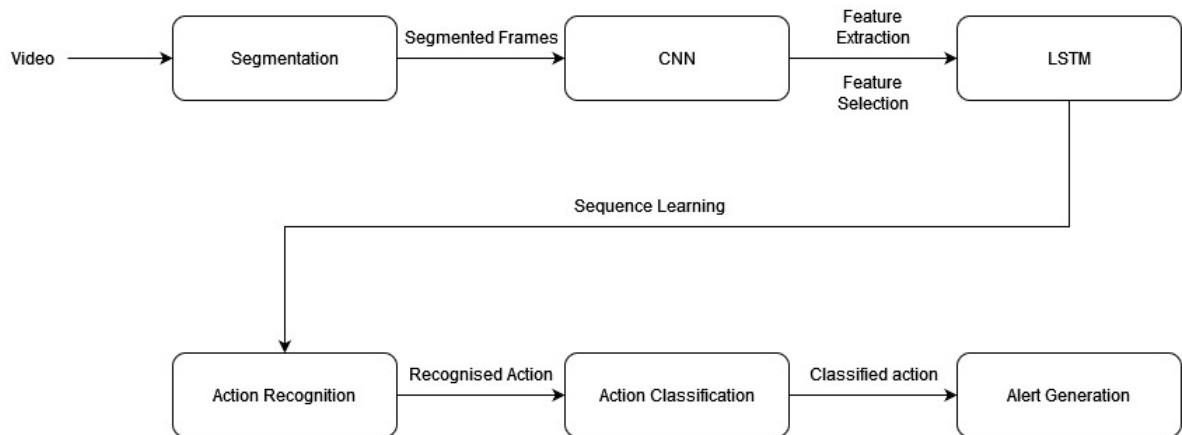


Figure 4.1 Architecture Diagram

4.2) Use Interface Design:

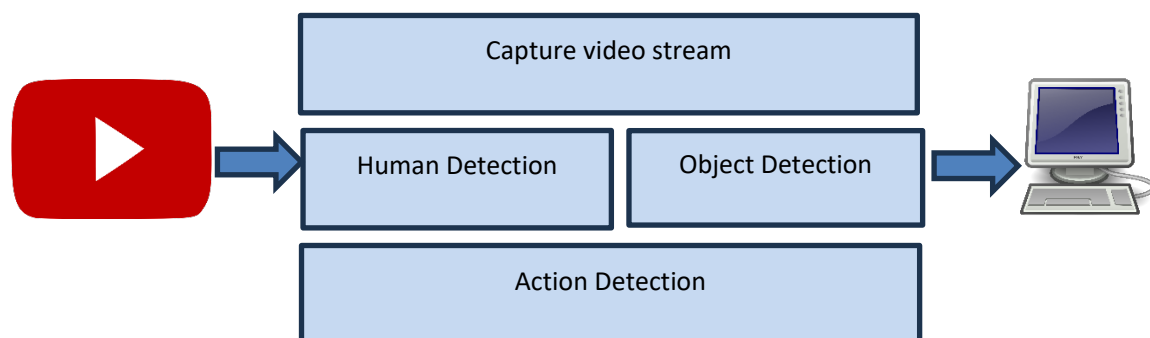


Figure 4.2 User Interface Design

4.3) Algorithmic Description of each Module:

Module 1: Human Detection

- a) Start
- b) Detect Video link
- c) Resize and Normalized Video
- d) Extract Frames from video
- e) Apply deep learning methodology LRCN algorithm
- f) Human Detected
- g) End

Module 2: Object Detection

- a) Start
- b) Detect video
- c) Compare video with dataset
- d) Classify object
- e) Object detected
- f) End

Module 3: Action Detection

- a) Start
- b) Detect video
- c) Action comparing with dataset
- d) Action classification and Detection
- e) End

4.4) System Diagram:

4.4.1) Data Flow Diagram:

a) DFD Level 0:

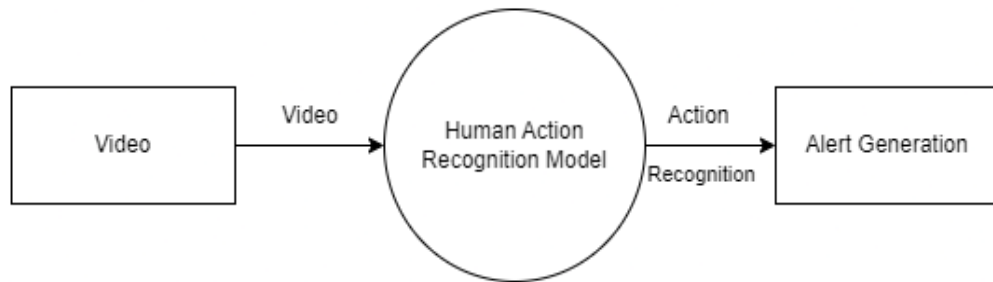


Figure 4.3 Data Flow Diagram Level 0

c) DFD Level 1:

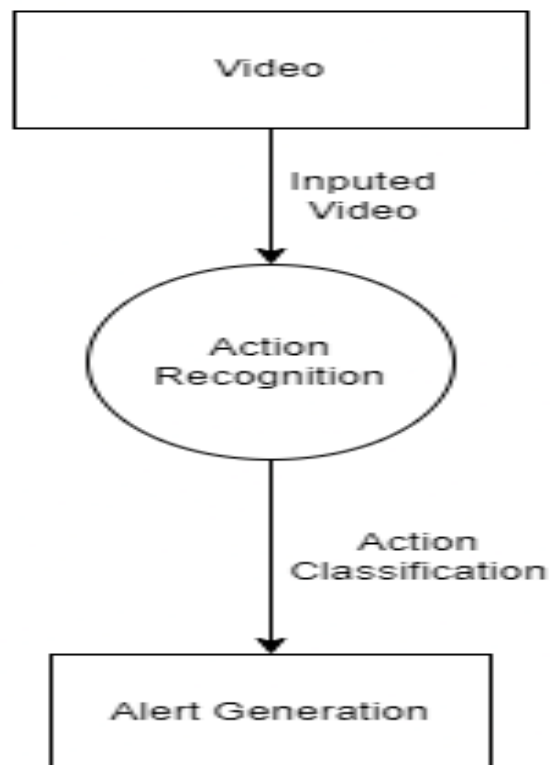


Figure 4.4 Data Flow Diagram Level 1

4.2.2) Sequence Diagram:

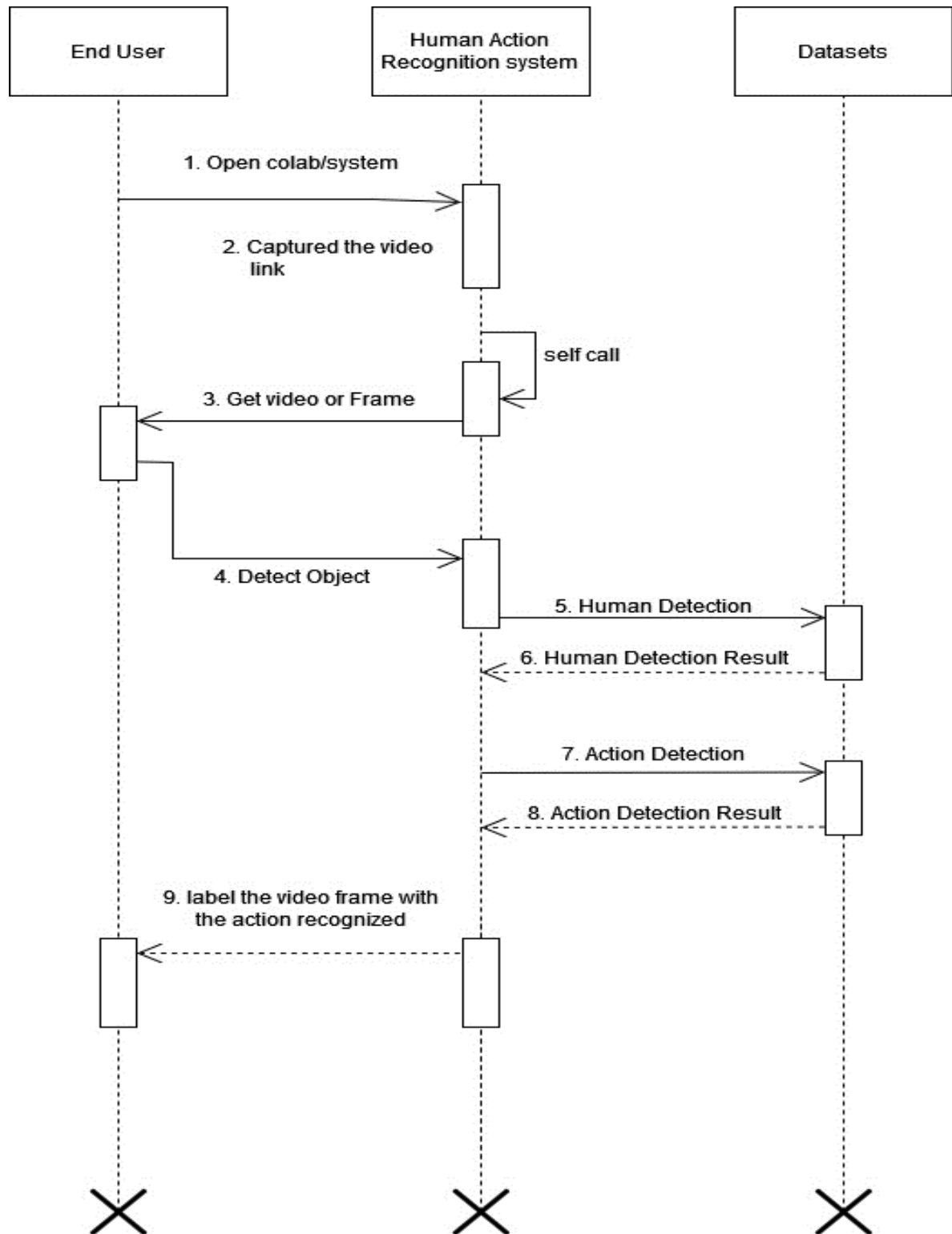


Figure 4.5 Sequence Diagram

4.4.3) Activity Diagram:

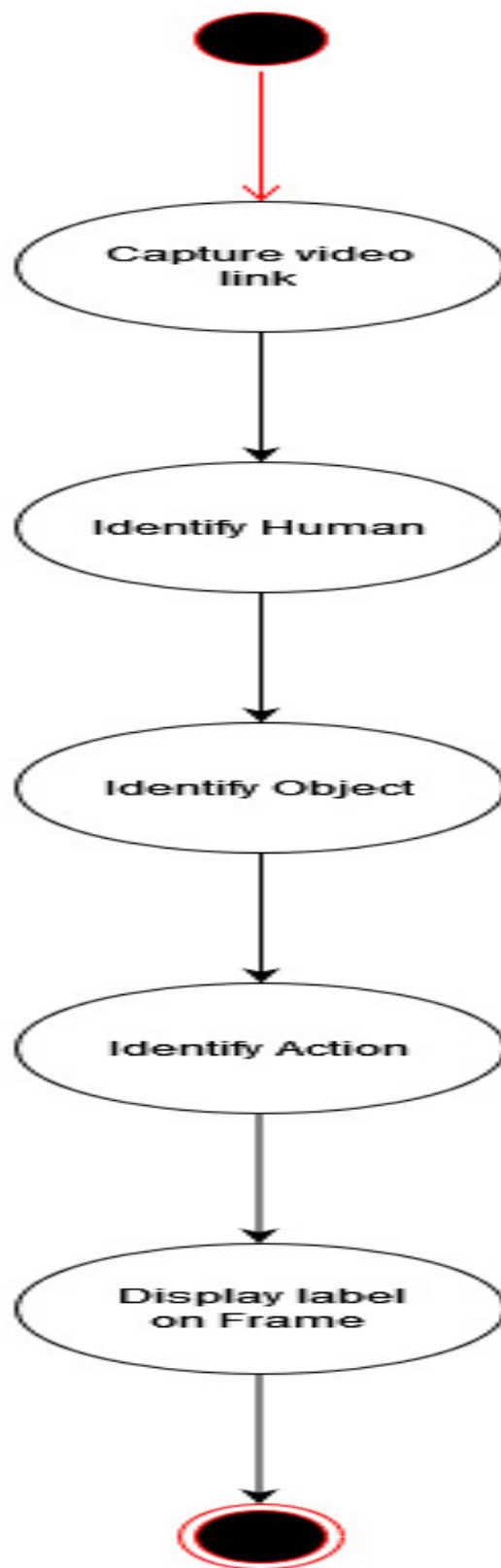


Figure 4.6 Component Diagram

4.4.4) Component Diagram:

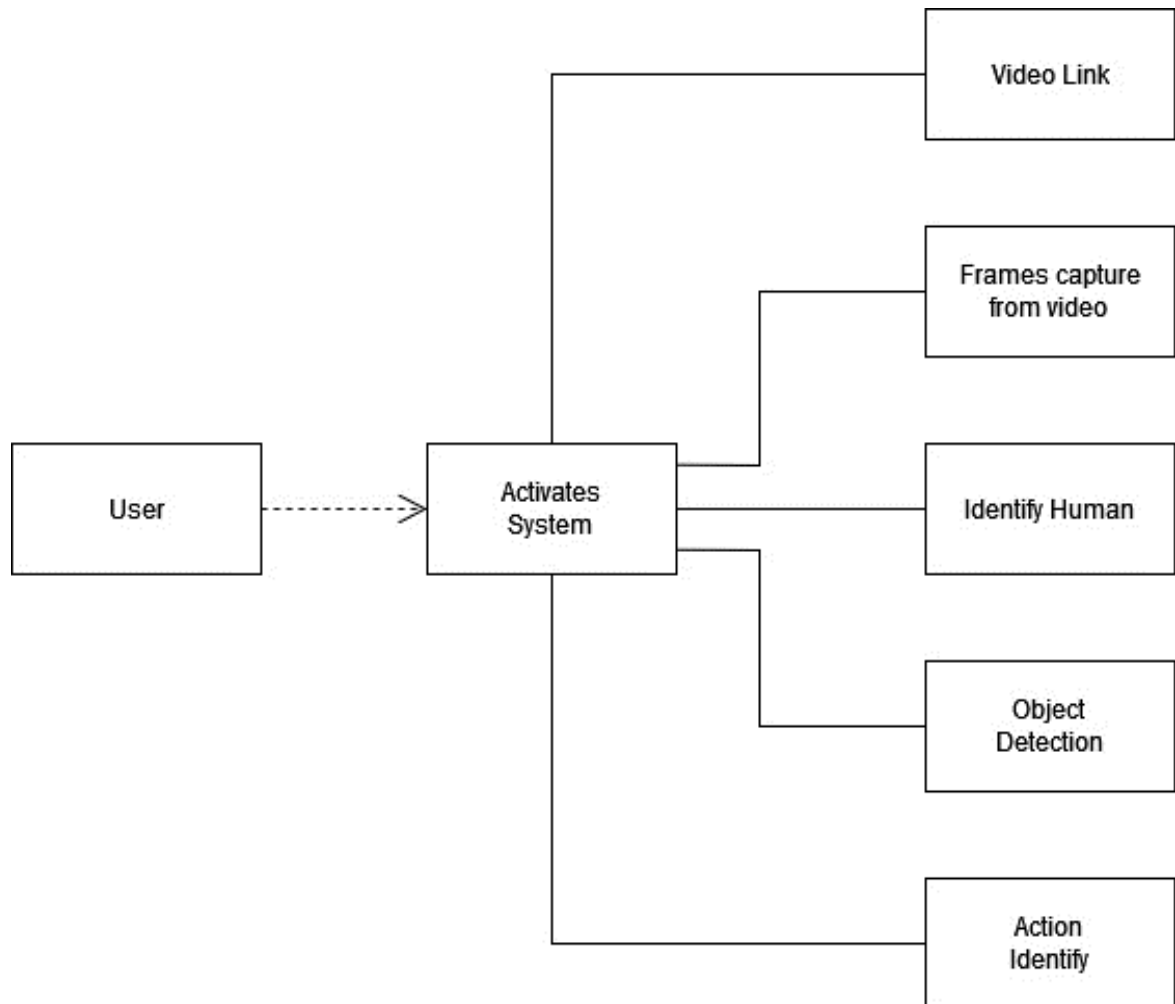


Figure 4.7 Component Diagram

5.Implementation:

5.1) Environmental Setup for Running the Project

Software required to run the system:

1. Software:

Google Colab	Linux
Python	Version 3.10
TensorFlow	Version 2.12.0

Hardware required to run the system:

2. Hardware:

Desktop/Laptop	1
RAM	16GB
Graphics Card	8GB

5.2) Detailed Description of Method:

Detailed implementation:

1. Dataset Description:

UCF-50 Dataset: <https://www.crcv.ucf.edu/data/UCF50.php>

The UCF50 dataset is a dataset containing 100 segments for each activity unit, with links representing 6 activities. Each episode has about 600 frames and the video is shot at 25fps. Kaggle database contains more than 100 videos extracted from movies and YouTube videos that can be used for education

UCF50 data set's 50 action categories collected from youtube are: Baseball Pitch, Basketball Shooting, Bench Press, Biking, Biking, Billiards Shot, Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Playing Piano, Pizza Tossing, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Playing Tabla, TaiChi, Tennis Swing, Trampoline Jumping, Playing Violin, Volleyball Spiking, Walking with a dog, and Yo Yo.

2. Data processing:

- a) Reading videos and text: using OpenCV library from class files and their files. Video class files are stored in NumPy arrays that are read from the class folder.
- b) Split into frames to create a sequence: Each video is read using the OpenCV library and only 30 frames of equal length are read to create a sequence of 30 frames.
- c) Resize: When we need to increase or decrease all pixels, we need to change the image. So, we resize all frames to width: 64px and height: 64px so that the picture frame looks the same.
- d) Normalization: Normalization helps the learning algorithm learn faster and capture the appropriate features of the image. So, we normalize the resized frame by dividing it by 255 so that each pixel value is between 0 and 1.

e) Stored in NumPy arrays: An array of 30 resized and normalized frames is stored in the NumPy array and put into the build.

3. Train Test Split Data:

75% of Data Used for Training

25% of Data Used for Testing

5.3) Implementation Details:

1) Import the Libraries:

Necessary libraries should be imported to build model.

Libraries like

```
import os  
import cv2  
import pafy  
import math  
import random  
import pandas as pd  
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 import keras  
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
```

So the above libraries gives following information like

OS library : In this module provides functions for interacting with the operating system, such as file operations, directory operations, and environment variables.

cv2 library : The cv2 module is from opencv library for computer vision task like to capture the video and process operation on it.

pafy library: It is used to extract the data and download Youtube video using URL'S.

math library: It provides mathematical functions and constants.

random library: It provides function for generating the random numbers.

numpy library: It is used for numerical operation. It provides array data structure and function.

pandas library: It is used for data manipulation and analysis.

datetime library: It supplies classes for working with dates and times.

matplotlib library: It is used for creating visualizations, such as plots and graphs.

moviepy library: It is used for video editing and processing.

tensorflow.keras library: It is used for building and training neural networks.

2) Finetune and load the dataset:

In this model we used UCF50 data set published by CRCV. This dataset contains 50 video actions.

And we have fine tune it by adding two more action which are Fire and StreetFighting.

After this we load the dataset and split it into train and test in order of 75% and 25% respectively.

Convert it into bgr to rgb.

```
rgb_frame = cv2.cvtColor(bgr_frame, cv2.COLOR_BGR2RGB)
```

```
import zipfile
```

```
zip_ref = zipfile.ZipFile('/content/drive/MyDrive/UCF52.zip', 'r')
```

```
zip_ref.extractall('/content/dataset')
```

```
zip_ref.close()
```

```
dataset_path = '/content/dataset'
```

3) Resizing and Normalizing Dataset:

In this the video which would be taken for training the model is preprocessed by resizing and normalizing it.

It is resized into 64X64 and divided by 255 to convert it into 0 and 1, to make the color intensity to black and white.

Also defining the classes list on which the model will be trained for.

IMAGE_HEIGHT , IMAGE_WIDTH = 64, 64

SEQUENCE_LENGTH = 40

DATASET_DIR = "/content/dataset/UCF50/UCF50"

CLASSES_LIST = ["Punch", "Fire", "Nunchucks", "TaiChi", "StreetFighting"]

4) Creating Model for Following modules:

We create the LRCN (Long-term Recurrent Convolutional Networks) model by combining the CNN (convolutional neural networks) and RNN (recurrent neural networks)

CNN:

Convolutional Neural Network (CNN): The CNN component is responsible for extracting spatial features from individual frames of a video.

It consists of multiple convolutional layers followed by pooling layers to capture visual patterns and hierarchical representations.

The CNN learns to recognize spatial patterns in each frame, such as edges, textures, and object features.

RNN:

The RNN component is used to capture temporal dependencies and sequential information across frames.

It takes the output features from the CNN and processes them sequentially through recurrent layers, such as LSTM (Long Short-Term Memory).

The RNN captures the temporal dynamics of the video by modeling the dependencies between consecutive frames and learning long-term dependencies.

By integrating both spatial and temporal information, LRCN can effectively recognize and classify actions or activities in videos.

By passing the perfect parameters to the model it will help to build a efficient model.

```
model = Sequential ()  
  
model.add(TimeDistributed(Conv2D(32, (3, 3), padding='same', activation =  
'relu'), input_shape = (SEQUENCE_LENGTH, IMAGE_HEIGHT,  
IMAGE_WIDTH, 3)))  
  
model.add(TimeDistributed(MaxPooling2D((4, 4))))  
  
model.add(Dense(len(CLASSES_LIST), activation = 'softmax'))  
  
model.summary()
```

Also use early stopping to make model perfectly fit and not overfitting nor underfitting.

```
early_stopping_callback = EarlyStopping(monitor = 'accuracy', patience = 50, mode  
= 'max', restore_best_weights = True)
```

In this model RMSProp optimizer is used. RMSprop (Root Mean Square Propagation) is an adaptive learning rate optimization algorithm that aims to handle different gradient magnitudes.

It uses a moving average of squared gradients to scale the learning rate for each parameter.

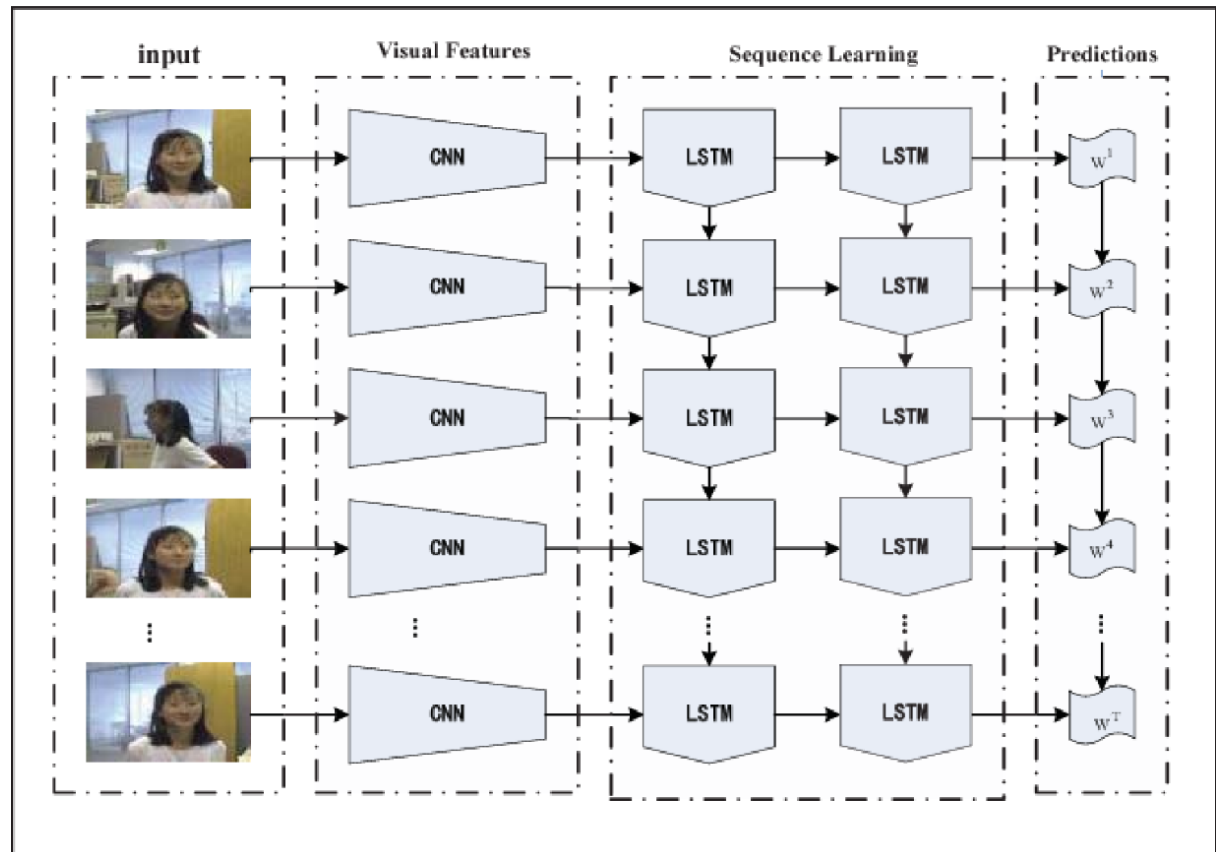
RMSprop is known for its ability to converge quickly and handle sparse gradients effectively.

It is often a good choice when dealing with non-stationary objectives or problems with varying magnitudes of gradients.

```
model.compile(loss = 'categorical_crossentropy', optimizer = 'RMSProp',  
metrics = ["accuracy"])
```

Model Creation:

A deep learning Network, LRCN is used in our proposed system from video surveillance



The main idea behind LRCNs is to use a combination of CNNs to learn the properties of images and LSTMs to convert sequences of images into lists, sentences, occurrences or whatever you want. So, as seen, the input is processed by the CNN whose output is fed into a set of sequential models.

LSTM networks are well suited for classification, processing, and forecasting based on real-time data, where significant events may have long-term tradeoffs between them. LSTMs were developed to solve the gradient extinction problem that can be encountered when training RNNs.

6. Integration and Testing:

6.1) Description of the Integration Modules

1) Module Name: - Human Detection

Module Input: Frames captured from video.

Description:

- a) This module processes the frames from the video for detecting a human object.
- b) It uses the LRCN algorithm and pretrained weights to identify human in video.
- c) The CNN extract the visual features and the LSTM store the value of the frames for longer time to include all the frames.

Output: Human Object detected from frame.

2) Module Name: Object Detection

Module Input: Things or items edge detected from the frames of videos.

Description:

- a) This function collects the frames and loads in the trained model.
- b) It then applies the deep learning method by passing through the hidden layers of model while different weights are applied on the frames.
- c) By comparing the things or items with frames or images of videos already available in the dataset on the basis of this comparison it classifies whether the object is Nunchuck, building, trees or boxing gloves.

Output: Detecting and classifying the object.

3) Module Name: Action Detection

Module Input: Along with human and object detected from the frames of videos.

Description:

- a) This module or function collects the frames from the video and loads it into the trained model. It then compares with the trained dataset and perform action on it.
- b) It checks for the similar action from the dataset and displays the result on the video frame by frame. Action like Fire, Punch, Taichi, Nunchuck and Streetfighting.

6.2) Testing:

Unit Testing:

1. Human Identification

Test Case No.	Test Case	Input	Expected Output	Actual Output	Status
1	Check whether the human is present in frame or not.	Object detection dataset i.e., Human.	Yes, Human is found in frame.	Yes, Human is found in frame.	Pass
2	Check whether the module can detect multiple persons in frame.	Object detection dataset i.e., Human.	Yes, Multiple humans are found in frame.	Yes, Multiple humans are found in frame.	Pass
3	Check whether the module can detect only human object.	Object detection dataset i.e., Human.	Yes, Module can detect human object present in frame.	Yes, Module can detect human object present in frame.	Pass

2. Object Detection

Test Case No.	Test Case	Input	Expected Output	Actual Output	Status
1	Detect things	Object detection dataset i.e., Things.	Yes, Object is found in frame.	Yes, Object is found in frame.	Pass
2	Classify object on the basis of thing detected.	Object detection dataset i.e., Things.	Objects classified such as Nunchucks, Tree, Buildings, Punching gloves.	Objects classified such as Nunchucks, Tree, Buildings, Punching gloves.	Pass

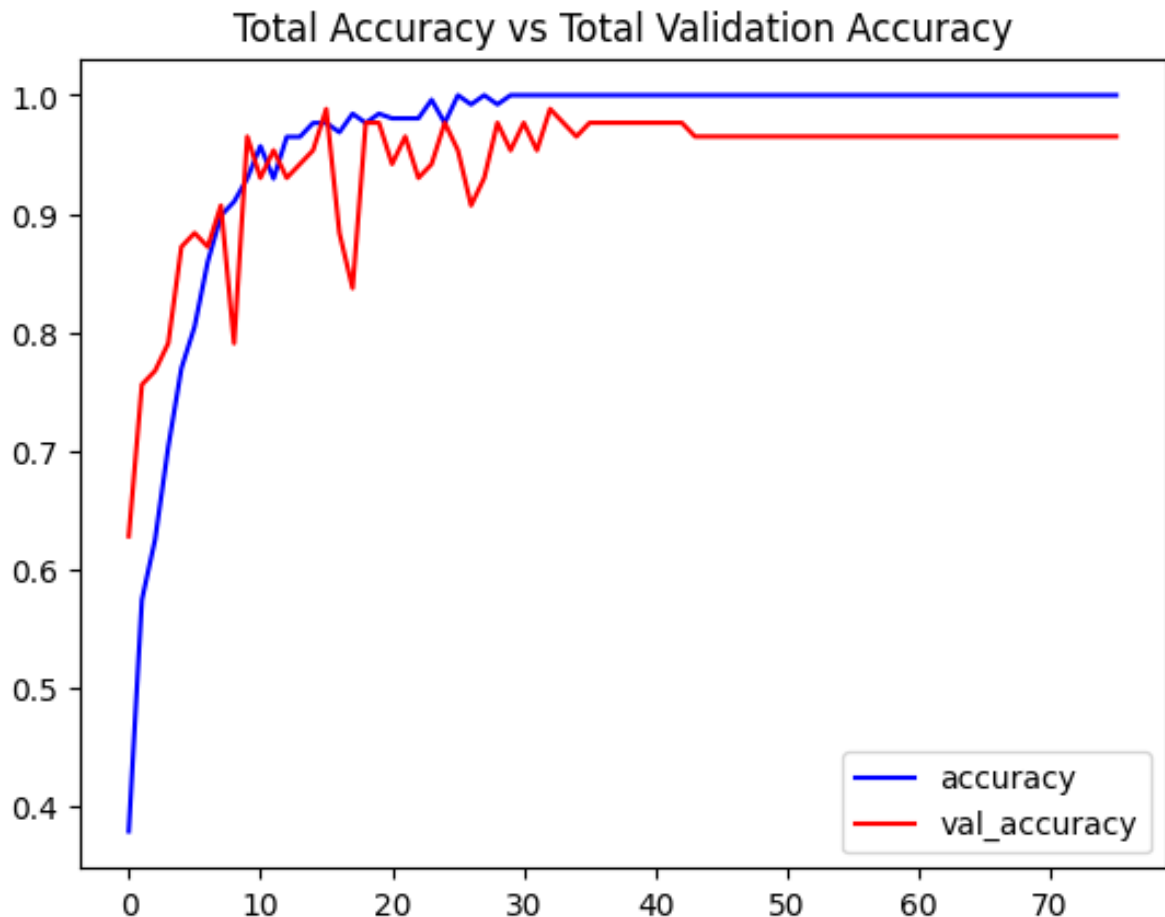
3.Action Detection

Test Case No.	Test Case	Input	Expected Output	Actual Output	Status
1	Check whether the action is identified.	Actions as per the dataset.	Yes, Action is properly identified	Yes, Action is properly identified	Pass

System Testing:

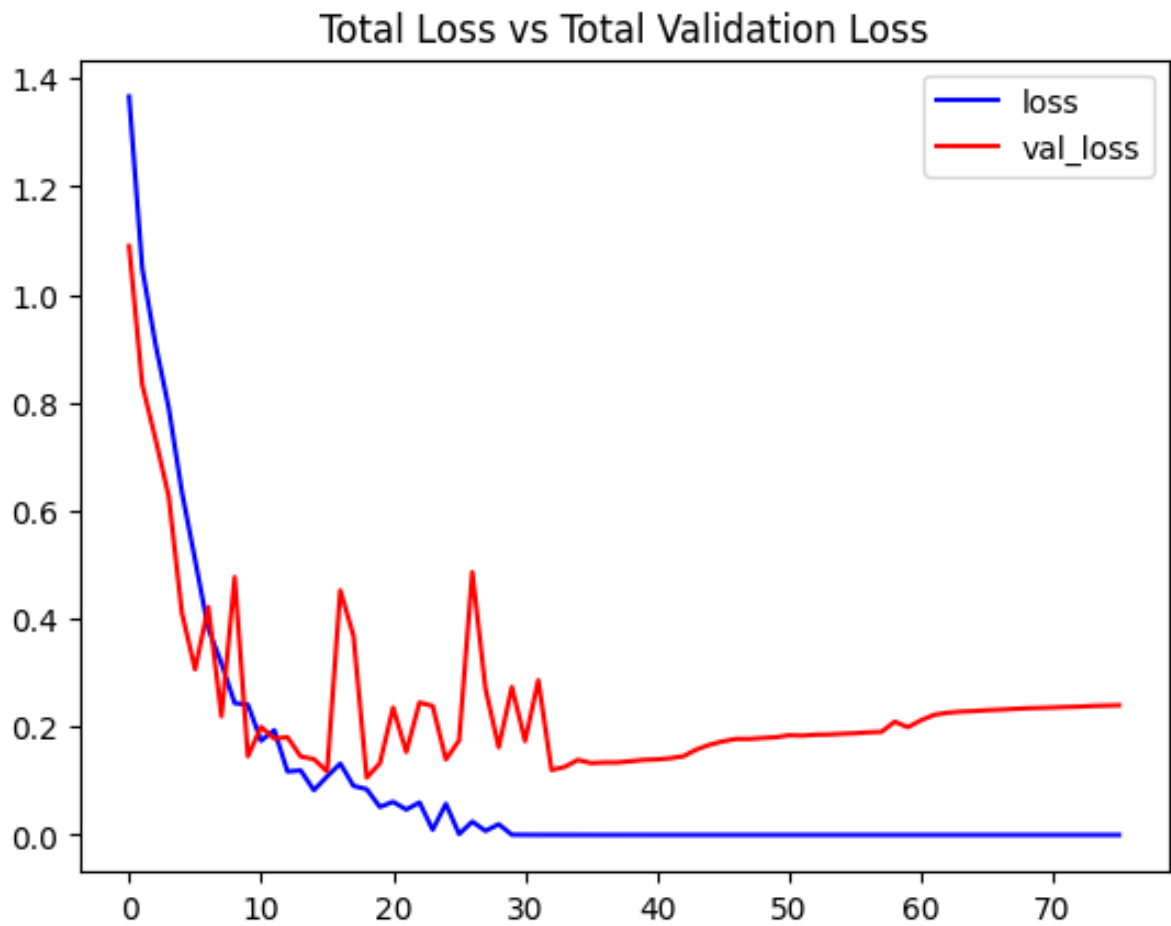
Test Case No.	Test Case	Input	Expected Output	Actual Output	Status
1	Check whether the object is detected in frame or not.	Object detection dataset.	Objects are detected.	Objects are detected.	Pass
2	Check whether the human is detected in frame or not.	Human detection dataset.	Human detection dataset.	Humans are detected.	Pass
3	Check whether the action is detected in frame or not.	Action detection dataset.	Action detection dataset.	Actions are detected.	Pass
4	Accuracy of actions detected in frame.	Action detection dataset.	Accuracy between 90% to 100%	Accuracy between 90% to 100%	Pass
5	Check whether the system is able to detect multiple actions in single frame.	Video containing multiple actions.	Single action is detected.	Multiple actions are detected.	Fail

7. Performance Analysis:



The above graph gives the description of Accuracy vs Validation Accuracy for LRCN Algorithm. It shows that the accuracy is stabilized after 25 epochs to give a 100% accuracy. The Validation accuracy nearly follows the curve of the total accuracy indicating that the model built isn't overfitted and the Loss Function is minimized which is one of the objectives of the project.

The LRCN(LSTM+RCN) Algorithm is much better than the ConvLSTM(CNN+LSTM) .



The above graph represents the steady decrease in losses as the epoch approaches till 30 And then stabilizes further. The validation losses also stabilize with increase in epoch.

8. Future Scope:

A more efficient software can understand and analyze long videos on a daily basis. Although many comprehensive review articles have been published on the general topic of HAR, the development of a range of topics, along with the multidisciplinary nature of HAR, encourage the need for topic review. In fact, most computer vision applications are specialized for HAR work, including human computer interaction, virtual reality, security, video surveillance, and homemonitoring.

This creates new and important challenges in the development cycle of HAR models. Here we present a unique insight into current work and research used to detect human movement. Comparing idiosyncratic human motion using similar models is difficult because of well-designed methods to represent similarities and the results depend on the image dataset used. This will be useful for the researcher's future research in this area. The system of generating an alert by classifying the action according to the intent and notifying the end user if the action recognized is dangerous can be an extended part of the project

9. Applications:

Advances in today's technology offer us new ways to improve the quality of life of the elderly and disabled. Housing services and operations use HAR technology and analytics to monitor residents and help keep them safe.

The smart home is an environment filled with sensors that improve the safety and well-being of residents while monitoring their health. Thus, such buildings with HAR systems contribute to the independence and quality of life of people who need physical and mental support. Basically, in a smart building, the data collected by the sensors is analyzed and the behavior of the inhabitants and their interaction with the environment are monitored.

10. Installation Guide and User Manual:

Installation:

- 1) Install Python's version 3.10
- 2) Install Anaconda Navigator or Visual Studio Code or use Google Colab directly.
- 3) Then install the libraries. Libraries to be installed pip:

TensorFlow version--> 2.12.0

pandas --> 1.5.3

opencv --> 4.7.0

pafy --> 0.5.5

NumPy --> 1.22.4

Matplotlib --> 3.7.1

moviepy --> 1.0.3

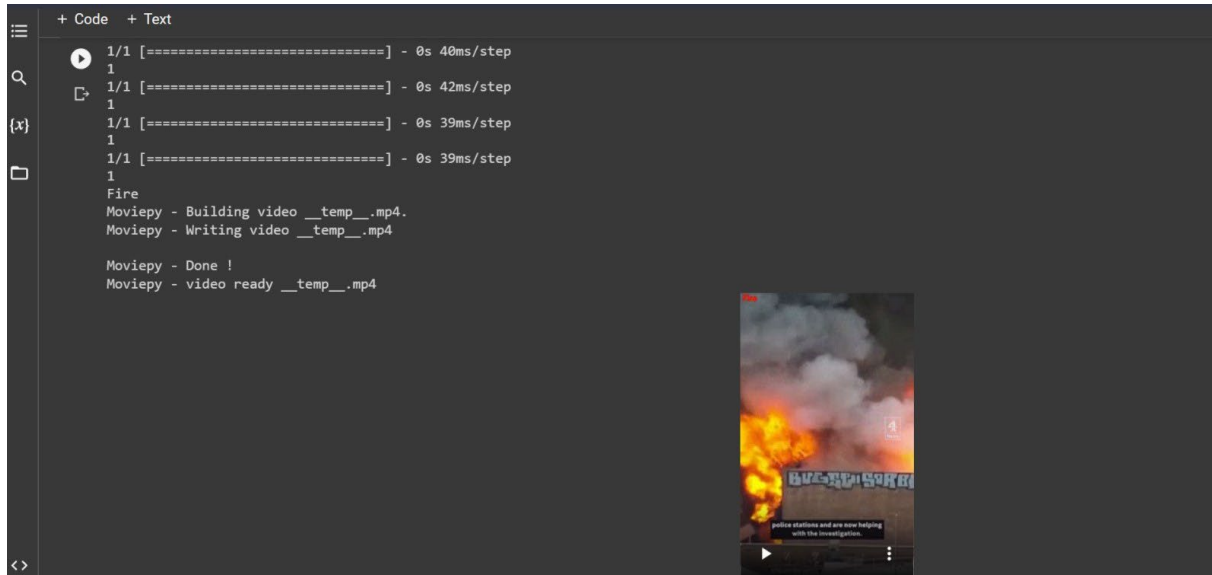
keras --> 2.12.0

imageio-->2.31.0

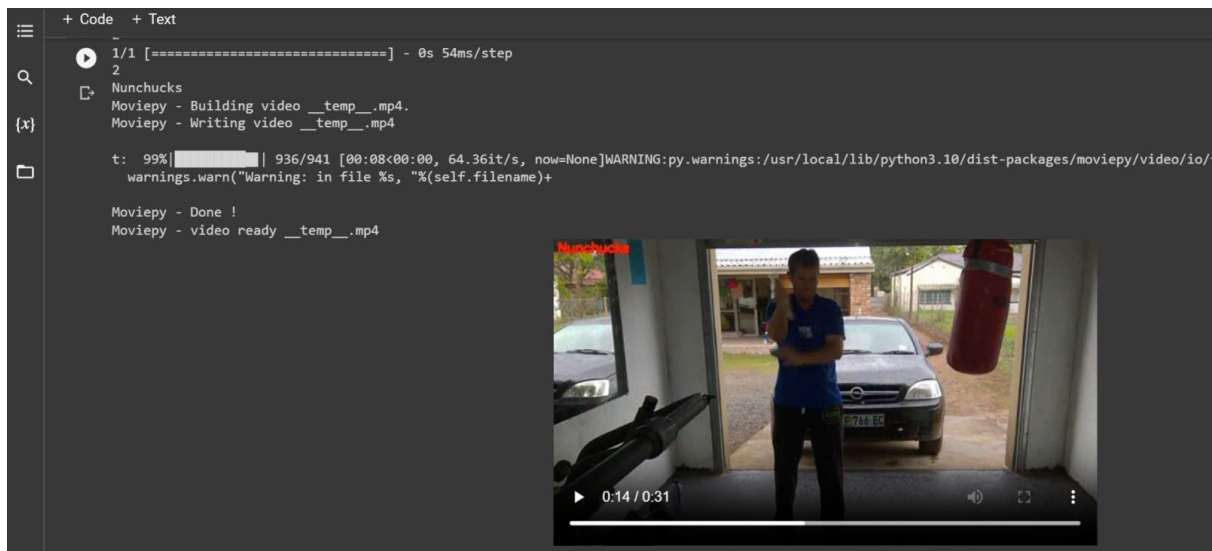
youtube-dl --> 2021.12.17

yt-dlp --> 2023.03.04

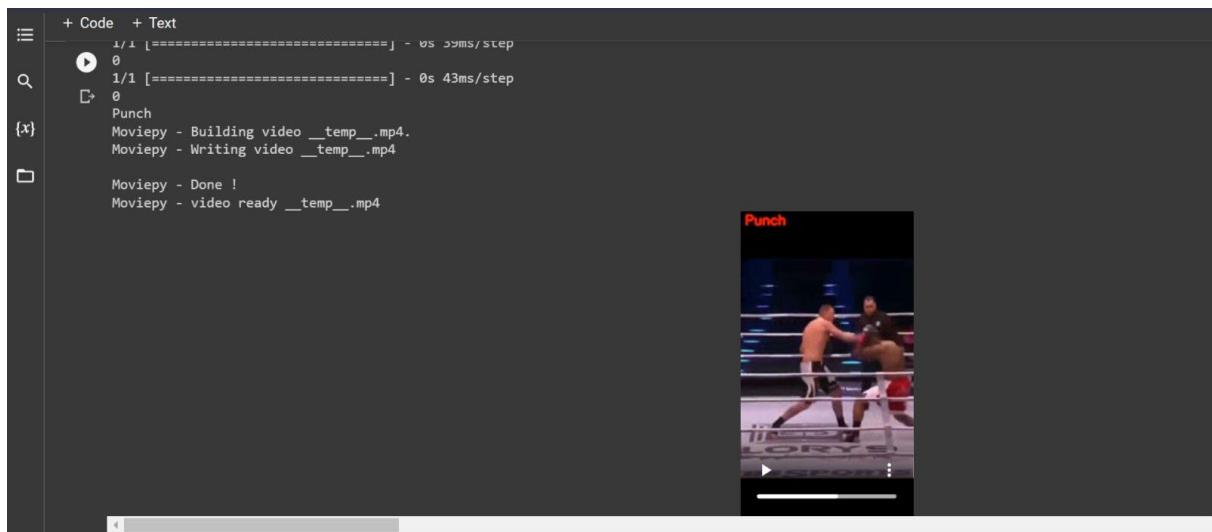
User Manual:



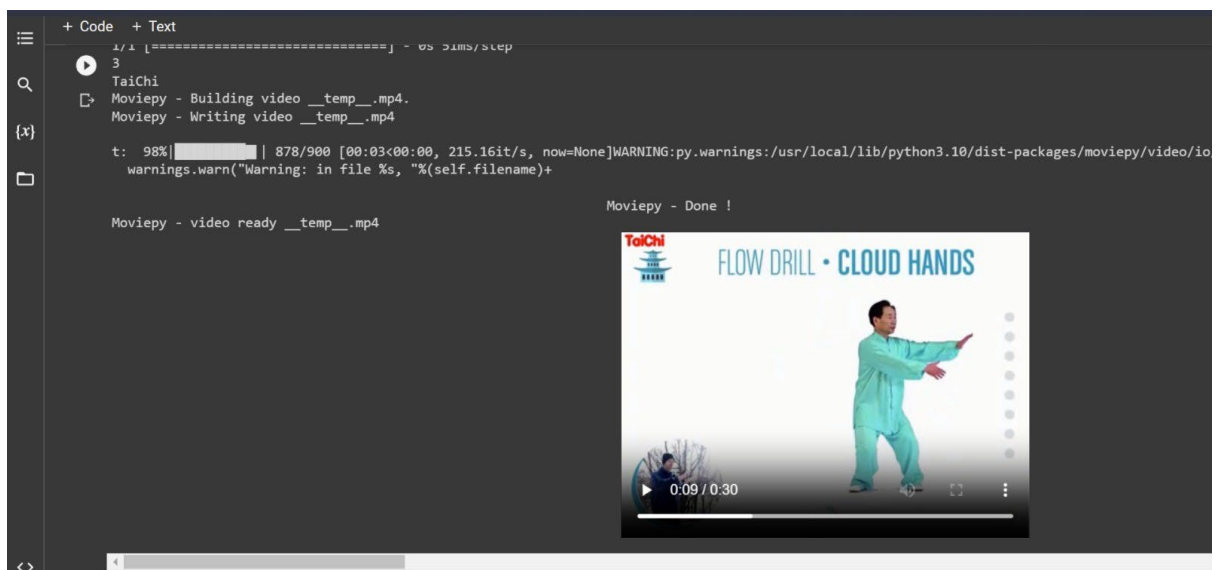
The above snap shows that the system detected the Fire from the Video Input



The above snap shows that the system detected the Action of Nunchucks with 99% accuracy.



The above snap shows that the system detected the action of Punching.

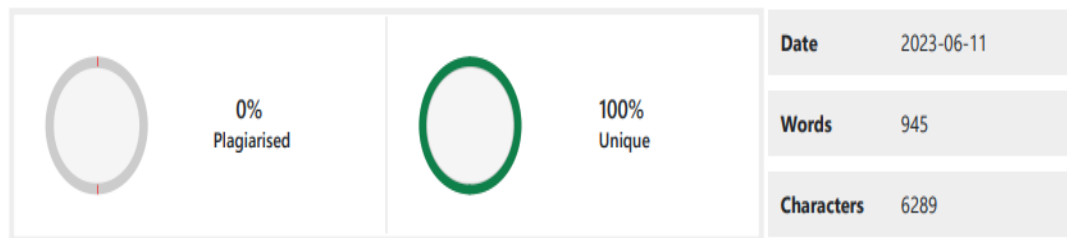


The above snap shows that the system detected the action of TaiChi with 98% accuracy.

11. Plagiarism Report:



PLAGIARISM SCAN REPORT



Content Checked For Plagiarism

1.INTRODUCTION

Computer vision is a branch of computer science that focuses on changing some of the complexities of human vision, enabling computers to recognize and process objects in images and videos like humans. Thanks to advances in intelligence and innovations in deep learning and neural networks, this field has prospered in recent years, outpacing humans in certain tasks related to search and recording.

In applications that relates to Action recognition, it is important that machines can comprehend and recognize the action performed and use the result for the judging the severity of the Action. This project offers an end-to-end neural network model that determines the action performed by the user to gain insights from it.


So, the first task is to acquire the dataset that matches with the Requirements mentioned in the SRS report. Then one can fine-tune the dataset so that it can incorporate other actions that the stakeholder requires. This load dataset is then partitioned into two parts one is for training and another one is for testing. The video that needs to be monitored to detect action from it needs to be segmented. Segmentation is the process of partitioning a video sequence into a disjoint set of consecutive frames. The incoming video is segmented further to extract key features from it. Using Feature Selection and Feature Extraction the critical features of the frame can be restored. In feature extraction the system transforms the

Scan Properties

Number of Words : **945**
Results Found : **0**

To or From To or From

Binary Translator PDF Converter



0% 100%

Plagiarism Unique

Make it Unique Start New Search

To check plagiarism in photos click here

Reverse Image Search

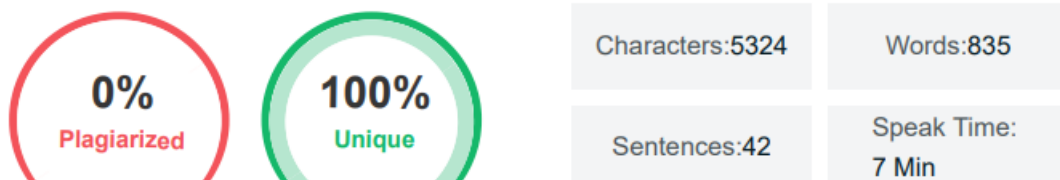
INTRODUCTION

Computer vision is a branch of computer science that focuses on changing some of the complexities of human vision, enabling computers to recognize and process objects in images and videos like humans. Thanks to advances in intelligence and innovations in deep learning and neural networks, this field has prospered in recent years, outpacing humans in certain tasks related to search and recording.

In applications that relates to Action recognition, it is important that machines can comprehend and recognize the action performed and use the result for the judging the severity of the Action. This project offers an end-to-end neural network model that determines the action performed by the user to gain insights from it.

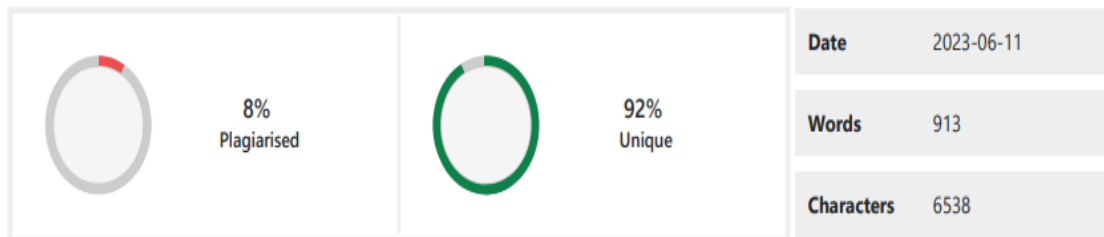
So, the first task is to acquire the dataset that matches with the Requirements mentioned in the SRS report. Then one can fine-tune the dataset so that it can incorporate other actions that the stakeholder

Plagiarism Scan Report





PLAGIARISM SCAN REPORT



Content Checked For Plagiarism

6.2) Testing:

Unit Testing:

1. Human Identification

Test Case No.	Test Case Input	Expected Output	Actual Output	Status
---------------	-----------------	-----------------	---------------	--------

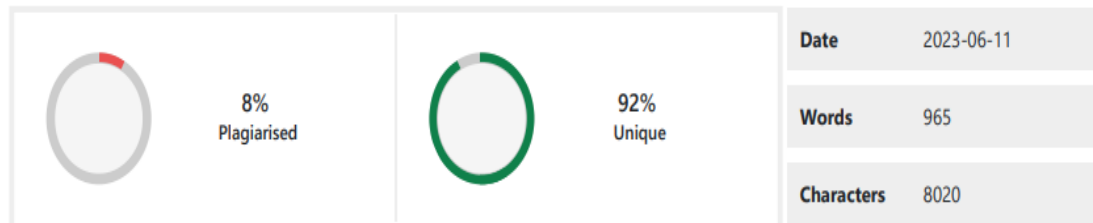
1	Check whether the human is present in frame or not.	Object detection dataset i.e., Human. Yes,	Human is found in frame. Yes, Human is found in frame.	Pass
---	---	--	--	------

2	Check whether the module can detect multiple persons in frame.	Object detection dataset i.e., Human. Yes,	Multiple humans are found in frame. Yes, Multiple humans are found in frame.	Pass
---	--	--	--	------

3	Check whether the module can detect only human object.	Object detection dataset i.e., Human. Yes,	Module can detect human object present in frame. Yes, Module can detect human object present in frame.	Pass
---	--	--	--	------



PLAGIARISM SCAN REPORT



Content Checked For Plagiarism

3.3) Use Case Diagram

Admin –

Load Test data

Train the model

Test the Performance

Deploy Model

Action Recognition

Administrator –

Alert Generation

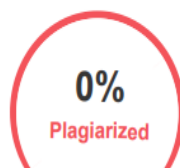
Figure 3.1 Use Case Diagram

3.4) Project Costing:



Jun 11, 2023

Plagiarism Scan Report



Characters:5324

Words:835

Sentences:42

Speak Time:
7 Min

Excluded URL

None

Content Checked for Plagiarism

he choice of the number of epochs depends on various factors, including the complexity of the problem, the size of the dataset, and the convergence behavior of the model. It is often determined through experimentation and monitoring the model's performance on a separate validation dataset. Early stopping techniques can also be employed to automatically stop training when the model's performance plateaus or starts to degrade, thus preventing overfitting. Once the training phase of the model is completed, the number of frames generated can show different actions that is spread across various frames. Thus, it becomes important that the action which is repeated maximum number of times in the frames is prioritized. This is where a Softmax Function comes into the

12.Ethics:

Declaration of Ethics As A Computer Science & Engineering Student,
I believe it is Unethical To,

1. Surf the internet for personal interest and non-class related purposes during classes
2. Make a copy of software for personal or commercial use
3. Make a copy of software for a friend
4. Loan CDs of software to friends
5. Download pirated software from the internet
6. Distribute pirated software from the internet
7. Buy software with a single user license and then install it on multiple Computers
8. Share a pirated copy of software
9. Install a pirated copy of software

13.References:

1. Vibekananda Dutta, Teresa Zielinska, “Prognosing Human Activity Using Actions Forecast and Structured Database”, IEEE Journal Paper, Volume 8, Page no. 6098 – 6116, 03 January 2020.
2. Yujiao Cheng, Masayoshi Tomizuka, “Long-Term Trajectory Prediction of the Human Hand and Duration Estimation of the Human Action”, IEEE Journal Paper, Volume 7, Page no. 247 – 254, 02 November 2021.
3. Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander Hauptmann, Li Fei-Fei, “Peeking into the Future: Predicting Future Person Activities and Locations in Videos”, IEEE Conference Paper, 09 April 2020.
4. M.S. Ryoo, “Human Activity Prediction: Early Recognition of Ongoing Activities from Streaming Videos” IEEE Conference Paper, 12 January 2012.
5. Junwei Liang¹, Lu Jiang², and Alexander Hauptmann, “SimAug: Learning Robust Representations from Simulation for Trajectory Prediction”.
6. David Jardim, Luís Miguel Nunes, and Miguel Sales Dias, “Human Activity Recognition and Prediction” Microsoft Language and Development Center, Lisbon, Portugal 2 Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal 3 IT - Instituto de Telecomunicações, Lisbon, Portugal 4 ISTAR-IUL, Lisbon, Portugal.
7. Junwei Liang, “From Recognition to Prediction: Analysis of Human Action and Trajectory Prediction in Video” www.lti.cs.cmu.edu.
8. Ong Chin Ann, Bee Theng Lau “Human activity recognition: A review” IEEE Conference Paper: March 2015.
9. Yu Kong, Yun Fu “Human Action Recognition and Prediction: A Survey” [International Journal of Computer Vision , 28 March 2022.](#)
10. Harshayu Girase ,Haiming Gang, Srikanth Malla, Jiachen Li ,Akira Kanehara, Karttikeya Mangalam, Chiho Choi, “Long Term and Key Intentions for Trajectory Prediction”, IEEE International Conference on Computer Vision October 2021.