HUMAN ACTIVITY RECOGNITION USING MACHINE LEARNING

Project Work Submitted in partial fulfillment of the requirements for award the degree of

MASTER OF COMPUTER APPLICATIONS

To the

Thiruvalluvar University, Serkkadu, Vellore-

632115 By

Ms.J.VARSHINI

(Register Number:31820P08024)

Guided By

Ms.M.PRITHI, MCA., M.Phil.,

MAY-2022

MARUDHAR KESARI JAIN COLLEGE FOR WOMEN (AFFILIATED TO THIRUVALLUVAR UNIVERSITY)

Serkkadu, Vellore – 632115

HUMAN ACTIVITY RECOGNITION USING MACHINE LEARNING

Project Work Submitted in partial fulfillment of the requirements for award the degree of

MASTER OF COMPUTER APPLICATIONS

To the

Thiruvalluvar University, Serkkadu, Vellore-632115 By

Ms.J.VARSHINI

(Register Number:31820P08024)

Guided By

Ms.M.PRITHI, MCA., M.Phil.,

MAY-2022

MARUDHAR KESARI JAIN COLLEGE FOR WOMEN (AFFILIATED TO THIRUVALLUVAR UNIVERSITY)

Serkkadu, Vellore – 632115

HUMAN ACTIVITY RECOGNITION USING MACHINE LEARNING

Bonafide work done by

Ms. J. VARSHINI

Reg No: 31820P08024

project work submitted in partial fulfillment

of the requirements for the degree of

Master of Computer Applications

To the **MARUDHAR KESARI JAIN COLLEGE FOR WOMEN**

VANIYAMBADI, Affiliated College from Thiruvalluvar University,

Serkkadu, Vellore-632115.

Internal Guide Head of the Department

Submitted for the Viva- Voice Examination held on ______________

External Examiner External Examiner

ACKNOWLEDGEMENT

I thank my Almighty for giving me all strength, courage and blessing to design and complete this project.

It was very exciting for me to work on the project of **"HUMAN ACTIVITY RECOGNITION USING MACHINE LEARNING".** During this process of project, I have gained both practical as well as theoretical knowledge of great significance.

I would like to thank our Management for providing the necessary infrastructure and facilities to fulfill the requirement for my project.

I would like to express my gratitude and thanks to our Principal **Dr.M.Inbavalli, M.Sc., MCA., M.Phil., Ph.D.,** for being a source of strength and helped us to surmount the obstacles our way to the successful completion of our challenging work.

I convey my sincere thanks and gratitude to **Ms.S.Abirami, MCA., M.Phil.,** Head Department of Computer Applications for their support and guidance throughout the project.

Words fail to thank adequately to our Project guide, **Ms.M.Prithi, MCA., M.Phil.,** Assistant Professor, who gave me all supports and helped me to clear all confusions regarding the project.

I would also like to express my gratitude to the lab administrators for considering the requirements for my project and satisfying me whenever I required them. It would have been highly impossible to carry forward the project without their assistance.

I thank to my parents and friends who supported me throughout the course of this project. I thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work.

I sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

TABLE OF CONTENTS

ABSTRACT

The features of image or video data set are extracted using different kinetic models associated with spatial or temporal feature leaning. Also, many deep layer trained models have been successfully used in this field to reach the fundamental goal of this model which is recognition and categorization of activity taking place. These activities can be of different varying nature such as day to day activities like running, jogging, eating, sitting, etc. There can be numerous types of activities in different fields like healthcare, childcare, security or work safety. Human Activity Recognition has a very significant role in different fields like human computer interaction, video surveillance system, robotics, daily monitoring, wildlife observation, etc. With the use of different datasets like UCF-101, HMDB-51, Hollywood2, Sports-1M and training them this task of recognition of activity can be efficiently done. The implementation of Convolutional Neural Network (CNN) model for image recognition with the help of OpenCV helps successful working of this model. Such application of different datasets on activity recognition model has helped in easy categorization of activity based on its nature whether normal or anomalous and suspicious.

INTRODUCTION

With rapid developments in the field of activity recognition and proposition of many new models based on scientific and technological developments immense progress in this field can be seen and observed. The development in deep learning and OpenCV with highly trained datasets have opened a new door of opportunities for upcoming research in this field. Such progress can lead to authentic and useful application of such models in this digitally equipped world for the well-being of all living beings. The use of new and advanced technology in this field by different researchers and developers have resulted in numerous applications of these models. Due to such highly trained models the activities taking place at real time can be monitored in very effective and optimum manner. Anomalous or suspicious activities can be treated with handy methods ensuring peace and harmony in the society of living beings. This can be also very useful in creating a smart home environment as well as smart healthcare service with the help of regular monitoring. Many security issues can be handled carefully and the damage to be caused can be minimized. Such effective application of these models in day-to-day life can also ensure the psychological wellbeing of people without concerns of the harm due to such activities. The human activity recognition model can be implemented with the use of camera module which captures the raw data that serves as an input to the recognition system. By creating different frames of such input data categorization of activity is done after feature extraction. Such activity is then identified as normal or suspicious and immediate alert is sent to the authority.

1.1 SYSTEM REQUIREMENTS

1.1.1 HARDWARE REQUIREMENTS

Processor	I5
Speed	3 GHz
RAM	8 GB (min)
Hard Disk	500 GB
Key Board	Standard Windows Keyboard
Mouse	Two or Three Button Mouse
Monitor	SVGA

1.2.2 SOFTWARE REQUIREMENTS

- Operating System: Linux, Windows/7/10
- Server: Anaconda, Jupyter, PyCharm
- Front End: tkinter |GUI toolkit
- Server-side Script: Python, AIML

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In recent times, smart phones are playing a vital role to recognize the human activities and became a well-known field of research. Detail overview of various research papers on human activity recognition are discussed in this paper. Artificial Intelligence (AI) models are developed to recognize the activity of the human from the provided UCI online storehouse. The data chosen is multivariate and we have applied various classification techniques Random Forest, KNN, Neural Network, Logistic Regression, Stochastic Gradient Descent and Naive Bayes to analyze the human activity.

2.1.1 DRAWBACKS

- It is not practical to represent multiple overlapping features and long-term dependencies.
- Number of parameters to be evaluated is huge. So, it needs a large data set for training.
- It requires huge amount of training in order to obtain better results.

2.2 PROPOSED SYSTEM

They applied the developed models to recognize human actions and achieved superior performance in comparison to baseline methods. Applied 3D CNNs to large scale video classification. Their spatiotemporal networks demonstrate significant performance improvements compared to strong feature-based baselines in a dataset of 1 million YouTube videos from 487 categories. In this paper, we develop a 3D CNN to integrate multisource of visual data. Each type of data stream provides several adjacent frames as input. In the experiments, we show that our model outperforms the baseline method based on hand-crafted features.

2.2.1 FEATURES

- It automatically detects the important features without any human supervision.
- They are great at handling image classification.
- They use the same knowledge across all image locations.

3.1 FILE DESIGN

The most creative and challenging face of the system development is system design. It provides the understanding and procedurals details necessary for the logical and physical stage of development. It designing a new system, the system analysis must have a clear understanding of the objectives, which the design is aiming to fulfil.

The first step is to determine how the output is produced and in what format. Second, input data and master files have to designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing.

Design of the system can be defined as a process of a system in sufficient details to permit its physical realization. Thus system design is a solution to how to approach to the creation of a new system. This important phase provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The design step provides a data design architectural design and procedural design.

3.2 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usual form of processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of. The input is designed in such a way so that it provides security and ease of use with retaining the privacy, Input Design considered the following things:

- What data should be given input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES:

1.Input design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy.

3.3 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need to the user. Efficient and intelligent output design improves the system's relationship to help user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

3.4 DATABASE DESIGN

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database management system manages the data accordingly. Database design involves classifying data and identifying interrelationships. This theoretical representation of the data is called an ontology.

The ontology is the theory behind the database's design. Database Design is defined as a collection of steps that help with designing, creating, implementing, and maintaining a business’s data management systems. The main purpose of designing a database is to produce physical and logical models of design for the proposed database system.

3.5 SYSTEM DEVELOPMENT

PYTHON

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective. It ranges from simple automation tasks to gaming, web development, and even complex enterprise systems. These are the areas where this technology is still the king with no or little competence: Machine learning as it has a plethora of libraries implementing machine learning algorithms. Python is a one-stop shop and relatively easy to learn, thus quite popular now. What other reasons exist for such universal popularity of this programming language and what companies have leveraged its opportunities to the max? Let’s talk about that. Python technology is quite popular among programmers, but the practice shows that business owners are also Python development believers and for good reason. Software developers love it for its straightforward syntax and reputation as one of the easiest programming languages to learn. Business owners or CTOs appreciate the fact that there’s a framework for pretty much anything – from web apps to machine learning. Moreover, it is not just a language but more a technology platform that has come together through a gigantic collaboration from thousands of individual professional developers forming a huge and peculiar community of aficionados. So, what is python used for and what are the tangible benefits the language brings to those who decided to use it? Below we’re going to discover that. Productivity and Speed It is a widespread theory within development circles that developing Python applications is approximately up to 10 times faster than developing the same application in Java or C/C++. The impressive benefit in terms of time saving can be explained by the clean object-oriented design, enhanced process control capabilities, and strong integration and text processing capacities. Moreover, its own unit testing framework contributes substantially to its speed and productivity.

PYCHARM

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. Choose the best PyCharm for you.

PyCharm is available in three editions

- Community (free and open-sourced): for smart and intelligent Python development, including code assistance, refactorings, visual debugging, and version control integration.
- Professional (paid): for professional Python, web, and data science development, including code assistance, refactorings, visual debugging, version control integration, remote configurations, deployment, support for popular web frameworks, such as Django and Flask, database support, scientific tools (including Jupyter notebook support), big data tools.
- Edu (free and open-sourced): for learning programming languages and related technologies with integrated educational tools.
- For details, see the editions comparison matrix.

Supported languages

To start developing in Python with PyCharm you need to download and install Python from python.org depending on your platform.

PyCharm supports the following versions of Python:

Python 2: version 2.7

Python 3: from the version 3.6 up to the version 3.10

Besides, in the Professional edition, one can develop Django, Flask, and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins (go to Settings | Plugins or PyCharm | Preferences | Plugins for macOS users, to find out more or set them up during the first IDE launch).

SUPPORTED PLATFORMS

REQUIREMENT	MINIMUM	RECOMMENDED
RAM	4 GB of free RAM	8 GB of total system RAM
CPU	Any modern CPU	Multi-core CPU. PyCharm supports multithreading for different operations and processing making it faster the more CPU cores it can use.
Disk space	2.5 GB and another 1 GB for caches.	SSD drive with at least 5 GB of free space.
Monitor resolution	1024x768	1920×1080
Operating System	Officially released 64-bit versions of the following: Microsoft Windows 8 or later macOS 10.13 or later any Linux distribution that supports Gnome, KDE, or Unity DE. PyCharm is not available for some Linux distributions such as RHEL6 or CentOS6, that do not include GLIBC 2.14 or later. Pre-released versions are not supported.	Latest 64-bit version of Windows, macOS, or Linux (for example, Debian, Ubuntu, or RHEL)

SPYDER

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software. It is released under the MIT license.

Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.

Spyder is extensible with first-party and third-party plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows, on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Spyder uses Qt for its GUI and is designed to use either of the PyQt or PySide Python bindings. QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either backend.

FEATURES

Features include:

- An editor with syntax highlighting, introspection, code completion
- Support for multiple IPython consoles
- The ability to explore and edit variables from a GUI
- A Help pane able to retrieve and render rich text documentation on functions, classes and methods automatically or on-demand
- A debugger linked to IPdb, for step-by-step execution
- Static code analysis, powered by Pylint
- A run-time Profiler, to benchmark code
- Project support, allowing work on multiple development efforts simultaneously
- A built-in file explorer, for interacting with the file system and managing projects
- A "Find in Files" feature, allowing full regular expression search over a specified scope

- An online help browser, allowing users to search and view Python and package documentation inside the IDE
- A history log, recording every user command entered in each console
- An internal console, allowing for introspection and control over Spyder's own operation

PLUGINS

Available plugins include:

- Spyder-Unit test, which integrates the popular unit testing frameworks Pytest, Unit test and Nose with Spyder
- Spyder-Notebook, allowing the viewing and editing of Jupyter Notebooks within the IDE

    - Download Spyder Notebook
    - Using conda: conda install spyder-notebook -c spyder-ide
    - Using pip: pip install spyder-notebook

- Spyder-Reports, enabling use of literate programming techniques in Python
- Spyder-Terminal, adding the ability to open, control and manage cross-platform system shells within Spyder

    - Download Spyder Terminal
    - Using conda: conda install spyder-terminal -c spyder-ide
    - Using pip: pip install spyder-terminal

- Spyder-Vim, containing commands and shortcuts emulating the Vim text editor
- Spyder-AutoPEP8, which can automatically conform code to the standard PEP 8 code style
- Spyder-Line-Profiler and Spyder-Memory-Profiler, extending the built-in profiling functionality to include testing an individual line, and measuring memory usage

ANACONDA PYTHON

Anaconda is a package manager, an environment manager, a Python/R data science distribution, and a collection of over 7,500+ open-source packages. Anaconda is free and easy to install, and it offers free community support.

Get the Anaconda Cheat Sheet and then download Anaconda.

Want to install conda and use conda to install just the packages you need? Get Miniconda.

Anaconda Navigator or conda?

After you install Anaconda or Miniconda, if you prefer a desktop graphical user interface (GUI) then use Navigator. If you prefer to use Anaconda prompt (or terminal on Linux or macOS), then use that and conda. You can also switch between them.

You can install, remove, or update any Anaconda package with a few clicks in Navigator, or with a single conda command in Anaconda Prompt (terminal on Linux or macOS).

- **To try Navigator**, after installing Anaconda, click the Navigator icon on your operating system's program menu, or in Anaconda prompt (or terminal on Linux or macOS), run the command anaconda-navigator.
- **To try conda**, after installing Anaconda or Miniconda, take the 20-minute conda test drive and download a conda cheat sheet.

Packages available in Anaconda

- Over 250 packages are automatically installed with Anaconda.

  Over 7,500 additional open-source packages (including R) can be individually installed from the Anaconda repository with the conda command.
- Thousands of other packages are available from Anaconda.org.
- You can download other packages using the pip install command that is installed with Anaconda. Pip packages provide many of the features of conda packages and in some cases they can work together. However, the preference should be to install the conda package if it is available.
- You can also make your own custom packages using the conda build command, and you can share them with others by uploading them to Anaconda.org, PyPI, or other repositories.

Previous versions

Previous versions of Anaconda are available in the archive. For a list of packages included in each previous version, see Old package lists.

Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it does not matter which one you download, because you can create new environments that include any version of Python packaged with conda. See Managing Python with conda.

Tkinter – Python

Tk/Tcl has long been an integral part of Python. It provides a robust and platform independent windowing toolkit, that is available to Python programmers using the tkinter package, and its extension, the tkinter.tix and the tkinter.ttk modules.

The tkinter package is a thin object-oriented layer on top of Tcl/Tk. To use tkinter, you don't need to write Tcl code, but you will need to consult the Tk documentation, and occasionally the Tcl documentation. tkinter is a set of wrappers that implement the Tk widgets as Python classes.

tkinter's chief virtues are that it is fast, and that it usually comes bundled with Python. Although its standard documentation is weak, good material is available, which includes: references, tutorials, a book and others. tkinter is also famous for having an outdated look and feel, which has been vastly improved in Tk 8.5. Nevertheless, there are many other GUI libraries that you could be interested in. The Python wiki lists several alternative GUI frameworks and tools.

Main tkinter module.

tkinter.colorchooser

Dialog to let the user choose a color.

tkinter.commondialog

Base class for the dialogs defined in the other modules listed here.

tkinter.filedialog

Common dialogs to allow the user to specify a file to open or save.

tkinter.font

Utilities to help work with fonts.

tkinter.messagebox

Access to standard tk dialog boxes.

tkinter.scrolledtext

Text widget with a vertical scroll bar built in.

tkinter.simpledialog

Basic dialogs and convenience functions.

tkinter.ttk

Themed widget set introduced in Tk 8.5, providing modern alternatives for many of the classic widgets in the main tkinter module.

Additional modules:

_tkinter

A binary module that contains the low-level interface to Tcl/Tk. It is automatically imported by the main tkinter module, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter.

idlelib

Python's Integrated Development and Learning Environment (IDLE).Based on tkinter.

tkinter.constants

Symbolic constants that can be used in place of strings when passing various parameters to Tkintercalls. Automatically imported by the main tkinter module.

tkinter.dnd

(experimental) Drag-and-drop support for tkinter. This will become deprecated when it is replaced with the Tk DND.

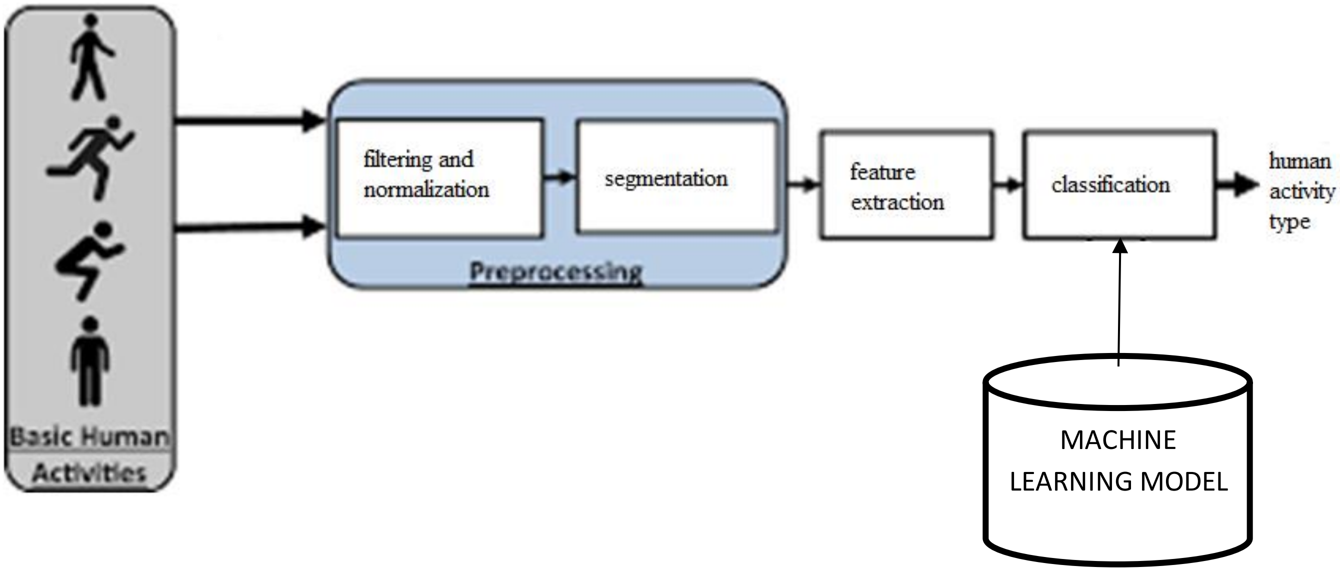
tkinter.tix

(deprecated) An older third-party Tcl/Tk package that adds several new widgets. Better alternatives for most can be found in tkinter.ttk.

turtle

Turtle graphics in a Tk window.

SYSTEM ARCHITECTURE



3.5.1 DESCRIPTION OF MODULES

MODULE

- CONVOLUTION NEURAL NETWORKS
- BUILD FACE RECOGNITION MODEL WITH CNN

MODULE DESCRIPTION

- **CONVOLUTION NEURAL NETWORKS**

Convolutional neural network (CNN) is a deformation of multi-layer perceptron inspired by biological vision and the most simplified preprocessing operation. It is essentially a forward feedback neural network. The biggest difference between convolutional neural network and multilayer perceptron is network. The first few layers are composed of a convolutional layer and a pooled layer alternately cascaded to simulate a simple cascade of cells and complex cells for high level feature extraction in the visual cortex.

The convolutional neurons respond to a portion of the input from the previous layer (called the local receptive field, with overlap between the regions), extracting higher-level features of the input; the neurons of the pooled layer are input to the previous layer. A portion of the area (no overlap between the areas) is averaged or maximized to resist slight deformation or displacement of the input. The latter layers of the convolutional neural network are typically an output layer of a number of fully connected layers and a classifier.

- **BUILD FACE RECOGNITION MODEL WITH CNN**

At present, face recognition algorithms can be roughly divided into two categories:

(1) Representation-based methods. The basic idea is to convert two-dimensional face input into another space, and then use statistical methods to analyze face patterns, such as Eigen face, Fisher face, and SVM.

(2) A feature-based method generally extracts local or global features and then sends a classifier for face recognition, such as recognition based on set features and HMM.

Convolutional neural network for face recognition can be considered as a feature-based method. It is different from traditional artificial feature extraction and high-performance classifier design for features. Its advantage is that feature extraction is performed by layer-by-layer convolution dimension reduction, and then through multi-layer nonlinear mapping, the network can automatically learn from the unprocessed training samples to form a feature extractor and classifier that adapts to the recognition task. This method reduces the requirements on the training samples, and the number of layers of the network.

4. TESTING

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. In the testing process we test the actual system in an organization and gather errors from the new system operates in full efficiency as stated. System testing is the stage of implementation, which is aimed to ensuring that the system works accurately and efficiently.

In the testing process we test the actual system in an organization and gather errors from the new system and take initiatives to correct the same. All the front-end and back-end connectivity are tested to be sure that the new system operates in full efficiency as stated. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently.

The main objective of testing is to uncover errors from the system. For the uncovering process we have to give proper input data to the system. So we should have more conscious to give input data. It is important to give correct inputs to efficient testing.

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus, the system testing is a confirmation that all is correct and an opportunity to show the user that the system works. Inadequate testing or non-testing leads to errors that may appear few months later.

This will create two problems, Time delay between the cause and appearance of the problem. The effect of the system errors on files and records within the system. The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the system to its limits.

The testing process focuses on logical intervals of the software ensuring that all the statements have been tested and on the function intervals (i.e.,) conducting tests to uncover errors and ensure that defined inputs will produce actual results that agree with the required results. Testing has to be done using the two common steps Unit testing and Integration testing. In the project system testing is made as follows:

The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. This is the final step in system life cycle. Here we implement the tested error-free system into real-life environment and make necessary changes, which runs in an online fashion. Here system maintenance is done every month or year based on company policies, and is checked for errors like runtime errors, long run errors and other maintenances like table verification and reports.

Integration Testing is a level of software testing where individual units are combined and tested as a group.

The purpose of this level is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration testing.

METHOD

Any of Black Box Testing, White Box Testing, and Gray Box Testing methods can be used. Normally, the method depends on your definition of ‘unit’.

TASKS

- Integration Test Plan
- Prepare
- Review
- Rework
- Baseline
- Integration Test Cases/Scripts
- Prepare
- Review
- Rework
- Baseline
- Integration Test
- Perform

UNIT TESTING

Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be working satisfactorily as regard to the expected output from the module.

BLACK BOX TESTING

Black box testing, also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

WHITE-BOX TESTING

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing).

GREY BOX TESTING

Grey box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. To test the Web Services application usually the grey box testing is used. Grey box testing is performed by end-users and also by testers and developers.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization

□ The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node

□ The Route add operation is done only when there is a Route request in need

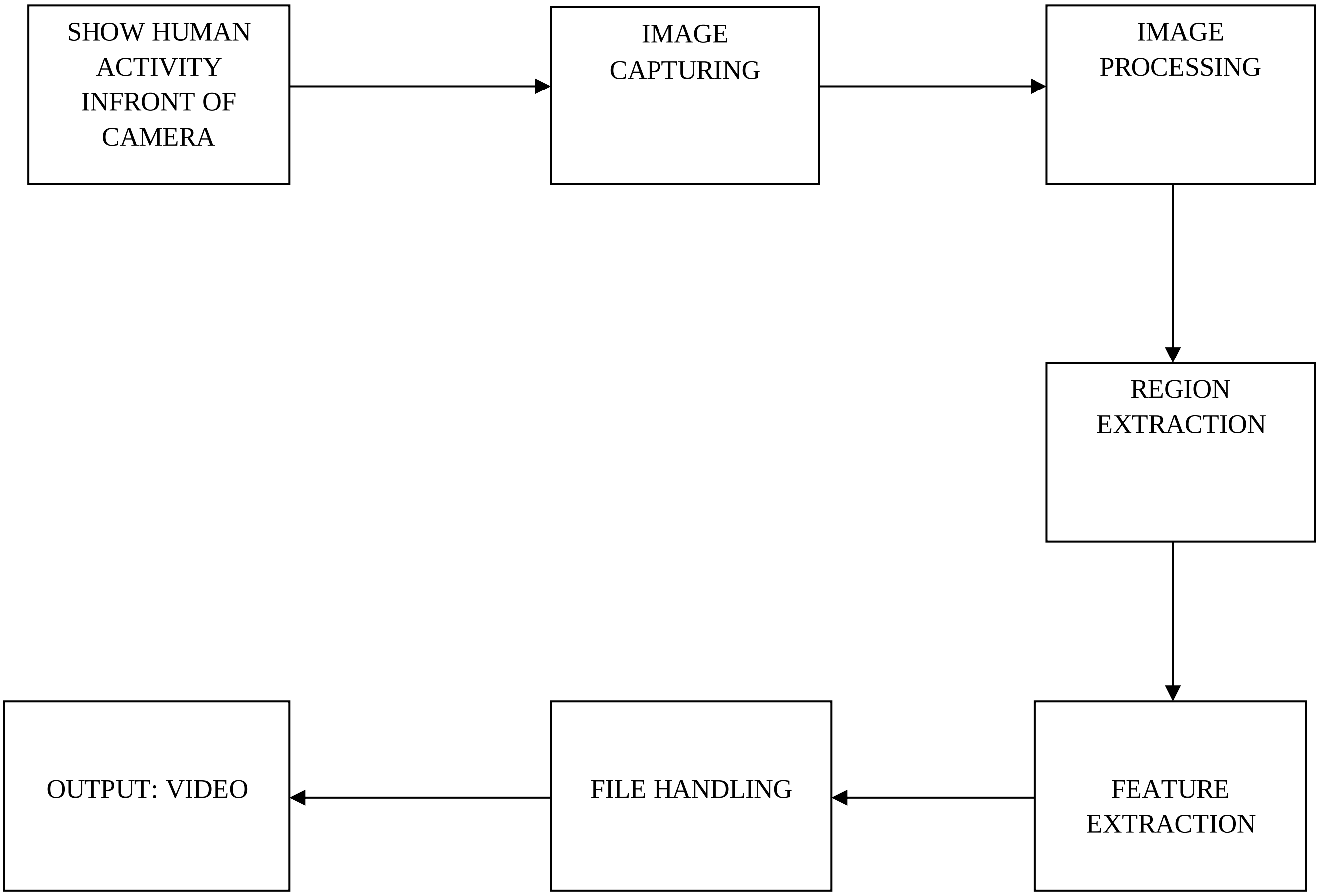
□ The Status of Nodes information is done automatically in the Cache Updating process

BUILD THE TEST PLAN

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

5.APPENDICES

A. DATAFLOW DIAGRAM



B. SAMPLE CODING

```
# Required imports

from collections import deque

import numpy as np

import cv2



# Parameters class include important paths and constants

class Parameters:

    def __init__(self):

        self.CLASSES = open("model/action_recognition_kinetics.txt"

                            ).read().strip().split("\n")

        self.ACTION_RESNET = 'model/resnet-34_kinetics.onnx'

#         self.VIDEO_PATH = None

        self.VIDEO_PATH = "test/example1.mp4"

        # SAMPLE_DURATION is maximum deque size

        self.SAMPLE_DURATION = 16

        self.SAMPLE_SIZE = 112




# Initialise instance of Class Parameter

param = Parameters()



# A Double ended queue to store our frames captured and with time
# old frames will pop

# out of the deque

captures = deque(maxlen=param.SAMPLE_DURATION)


# load the human activity recognition model

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

net = cv2.dnn.readNet(model=param.ACTION_RESNET)


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

# Take video file as input if given else turn on web-cam

# So, the input should be mp4 file or live web-cam video

vs = cv2.VideoCapture(param.VIDEO_PATH if param.VIDEO_PATH else 0)


while True:

    # Loop over and read capture from the given video input

    (grabbed, capture) = vs.read()


    # break when no frame is grabbed (or end if the video)`

    if not grabbed:

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

        break


    # resize frame and append it to our deque
```

```
capture = cv2.resize(capture, dsize=(550, 400))
captures.append(capture)

# Process further only when the deque is filled
if len(captures) < param.SAMPLE_DURATION:
    continue

# now that our captures array is filled we can
# construct our image blob
# We will use SAMPLE_SIZE as height and width for
# modifying the captured frame
imageBlob = cv2.dnn.blobFromImages(captures, 1.0,
                                   (param.SAMPLE_SIZE,
                                    param.SAMPLE_SIZE),
                                   (114.7748, 107.7354, 99.4750),
                                   swapRB=True, crop=True)

# Manipulate the image blob to make it fit as as input
# for the pre-trained OpenCV's
# Human Action Recognition Model
imageBlob = np.transpose(imageBlob, (1, 0, 2, 3))
imageBlob = np.expand_dims(imageBlob, axis=0)
```

```
    # Forward pass through model to make prediction
    net.setInput(imageBlob)
    outputs = net.forward()
    # Index the maximum probability
    label = param.CLASSES[np.argmax(outputs)]

    # Show the predicted activity
    cv2.rectangle(capture, (0, 0), (300, 40), (255, 255, 255), -1)
    cv2.putText(capture, label, (10, 25), cv2.FONT_HERSHEY_SIMPLEX,
                0.8, (0, 0, 0), 2)

    # Display it on the screen
    cv2.imshow("Human Activity Recognition", capture)

    key = cv2.waitKey(1) & 0xFF
    # Press key 'q' to break the loop
    if key == ord("q"):
        break

# ========================RUN COMMAND==========================
# python recognise_human_activity.py
```

C. SAMPLE INPUT

REQUIREMENTS

albumentations==1.1.0
cached-property==1.5.2
certifi==2021.10.8
charset-normalizer==2.0.12
clang==5.0
cycler==0.11.0
dpcpp-cpp-rt==2022.0.3
flatbuffers==1.12
fonttools==4.30.0
google-auth-oauthlib==0.4.6
google-pasta==0.2.0
graphviz==0.19.1
grpcio==1.34.1
idna==3.3
image==1.5.33
imageio==2.15.0
intel-cmplr-lib-rt==2022.0.3
intel-cmplr-lic-rt==2022.0.3
intel-opencl-rt==2022.0.3
intel-openmp==2022.0.3
kiwisolver==1.3.2

astunparse==1.6.3
libopencv==0.0.1
libpng-bins==0.0.3
libtiff==0.4.2
Markdown==3.3.6
mkl==2022.0.3
mkl-fft==1.2.0
mkl-random==1.2.1
networkx==2.6.3
numpy==1.21.5
oauthlib==3.2.0
opencv-python==4.5.5.64
openssl-python==0.1.1
opt-einsum==3.3.0
protobuf==3.19.4
pyasn1==0.4.8
pyasn1-modules==0.2.8
pydot==1.4.1
python-xz==0.4.0
pytz==2018.4

```
PyWavelets==1.3.0
qudida==0.0.4
requests==2.27.1
requests-oauthlib==1.3.1
rsa==4.8
scikit-image==0.18.1
scikit-learn==0.20.0
simple-sqlite==2.1.1
sqlparse==0.4.2
tbb==2021.5.2
tbb4py==2021.5.2
Tcl==0.2
tensorboard-data-server==0.6.1
tensorboard-plugin-wit==1.8.1
termcolor==1.1.0
tifffile==2021.11.2
tk==0.1.0
typing-extensions==3.7.4.3
urllib3==1.26.8
vc==2018.7.10
Werkzeug==2.0.3
wincertstore==0.2
wrapt==1.12.1
```

TRAINED DATASET

abseiling

air drumming

answering questions

applauding

applying cream

archery

arm wrestling

arranging flowers

assembling computer

auctioning

baby waking up

baking cookies

balloon blowing

bandaging

barbequing

bartending

beatboxing

bee keeping

belly dancing

bench pressing

bending back

bending metal

biking through snow

blasting sand

blowing glass

blowing leaves

blowing nose

blowing candles

bobsledding

bookbinding

bouncing on trampoline

bowling

braiding hair

breading or breadcrumbing

breakdancing

brush painting

brushing hair

brushing teeth

building cabinet

building shed

bungee jumping

busking

canoeing or kayaking

capoeira

carrying baby

cartwheeling

carving pumpkin

catching fish

catching or throwing baseball

catching or throwing frisbee

catching or throwing softball

celebrating

changing oil

changing wheel

checking tires

cheerleading

chopping wood

clapping

clay pottery making

clean and jerk

cleaning floor

cleaning gutters

cleaning pool

cleaning shoes

cleaning toilet

cleaning windows

climbing a rope

climbing ladder

climbing tree

contact juggling

cooking chicken

cooking egg

cooking on campfire

cooking sausages

counting money

country line dancing

cracking neck

crawling baby

crossing river

crying

curling hair

cutting nails

cutting pineapple

cutting watermelon

dancing ballet

dancing charleston

dancing gangnam style

dancing macarena

deadlifting

decorating the christmas tree

digging

dining

disc golfing

diving cliff

dodgeball

doing aerobics

doing laundry

doing nails

drawing

dribbling basketball

drinking

drinking beer

drinking shots

driving car

driving tractor

drop kicking

drumming fingers

dunking basketball

dying hair

eating burger

eating cake

eating carrots

eating chips

eating doughnuts

eating hotdog

eating ice cream

eating spaghetti

eating watermelon

egg hunting

exercising arm

exercising with an exercise ball

extinguishing fire

faceplanting

feeding birds

feeding fish

feeding goats

filling eyebrows

finger snapping

fixing hair

flipping pancake

flying kite

folding clothes

folding napkins

folding paper

front raises

frying vegetables

garbage collecting

gargling

getting a haircut

getting a tattoo

giving or receiving award

golf chipping

golf driving

golf putting

grinding meat

grooming dog

grooming horse

gymnastics tumbling

hammer throw

headbanging

headbutting

high jump

high kick

hitting baseball

hockey stop

holding snake

hopscotch

hoverboarding

hugging

hula hooping

hurdling

hurling (sport)

ice climbing

ice fishing

ice skating

ironing

javelin throw

jetskiing

jogging

juggling balls

juggling fire

juggling soccer ball

jumping into pool

jumpstyle dancing

kicking field goal

kicking soccer ball

kissing

kitesurfing

knitting

krumping

laughing

laying bricks

long jump

lunge

making a cake

making a sandwich

making bed

making jewelry

making pizza

making snowman

making sushi

making tea

marching

massaging back

massaging feet

massaging legs

massaging person's head

milking cow

mopping floor

motorcycling

moving furniture

mowing lawn

news anchoring

opening bottle

opening present

paragliding

parasailing

parkour

passing American football (in game)

passing American football (not in game)

peeling apples

peeling potatoes

petting animal (not cat)

petting cat

picking fruit

planting trees

plastering

playing accordion

playing badminton

playing bagpipes

playing basketball

playing bass guitar

playing cards

playing cello

playing chess

playing clarinet

playing controller

playing cricket

playing cymbals

playing didgeridoo

playing drums

playing flute

playing guitar

playing harmonica

playing harp

playing ice hockey

playing keyboard

playing kickball

playing monopoly

playing organ

playing paintball

playing piano

playing poker

playing recorder

playing saxophone

playing squash or racquetball

playing tennis

playing trombone

playing trumpet

playing ukulele

playing violin

playing volleyball

playing xylophone

pole vault

presenting weather forecast

pull ups

pumping fist

pumping gas

punching bag

punching person (boxing)

push up

pushing car

pushing cart

pushing wheelchair

reading book

reading newspaper

recording music

riding a bike

riding camel

riding elephant

riding mechanical bull

riding mountain bike

riding mule

riding or walking with horse

riding scooter

riding unicycle

ripping paper

robot dancing

rock climbing

rock scissors paper

roller skating

running on treadmill

sailing

salsa dancing

sanding floor

scrambling eggs

scuba diving

setting table

shaking hands

shaking head

sharpening knives

sharpening pencil

shaving head

shaving legs

shearing sheep

shining shoes

shooting basketball

shooting goal (soccer)

shot put

shoveling snow

shredding paper

shuffling cards

side kick

sign language interpreting

singing

situp

skateboarding

ski jumping

skiing (not slalom or crosscountry)

skiing crosscountry

skiing slalom

skipping rope

skydiving

slacklining

slapping

sled dog racing

smoking

smoking hookah

snatch weight lifting

sneezing

sniffing

snorkeling

snowboarding

snowkiting

snowmobiling

somersaulting

spinning poi

spray painting

spraying

springboard diving

squat

sticking tongue out

stomping grapes

stretching arm

stretching leg

strumming guitar

surfing crowd

surfing water

sweeping floor

swimming backstroke

swimming breast stroke

swimming butterfly stroke

swing dancing

swinging legs

swinging on something

sword fighting

tai chi

taking a shower

tango dancing

tap dancing

tapping guitar

tapping pen

tasting beer

tasting food

testifying

texting

throwing axe

throwing ball

throwing discus

tickling

tobogganing

tossing coin

tossing salad

training dog

trapezing

trimming or shaving beard

trimming trees

triple jump

tying bow tie

tying knot (not on a tie)

tying tie

unboxing

unloading truck

using computer

using remote controller (not gaming)

using segway

vault

waiting in line

walking the dog

washing dishes

washing feet

washing hair

washing hands

water skiing

water sliding

watering plants

waxing back

waxing chest

waxing eyebrows

waxing legs

weaving basket

welding

whistling

windsurfing

wrapping present

wrestling

writing

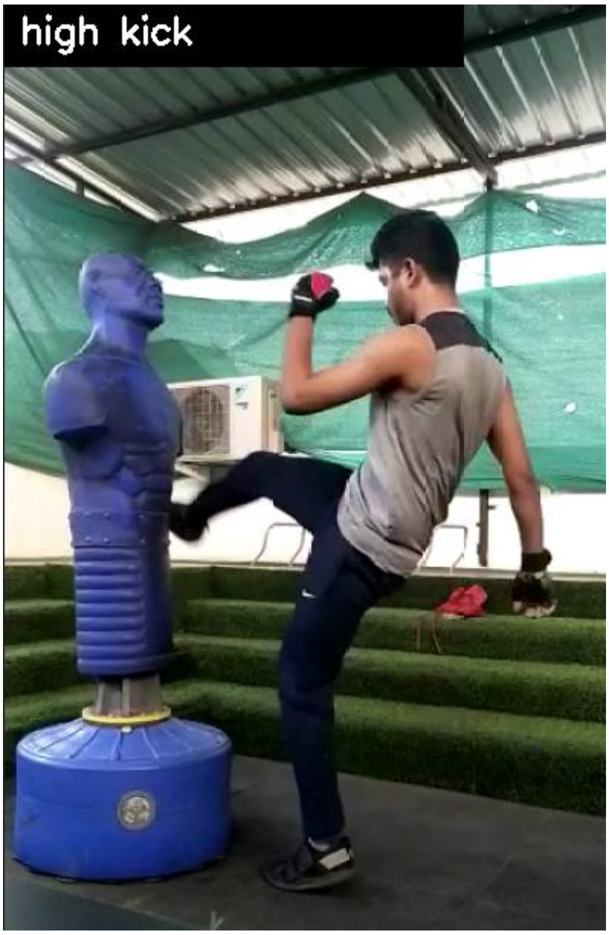
yawning

yoga

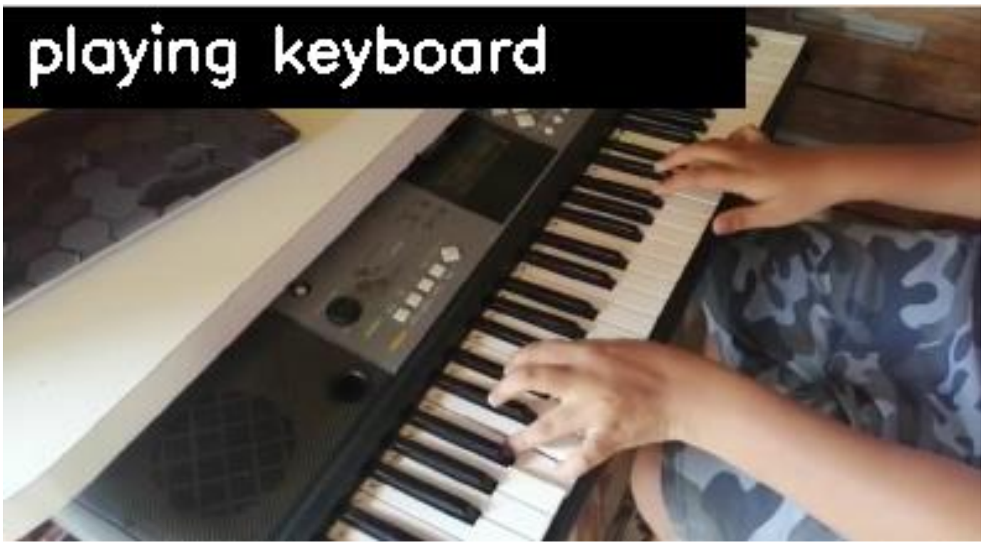
zumba

D.SAMPLE OUTPUT

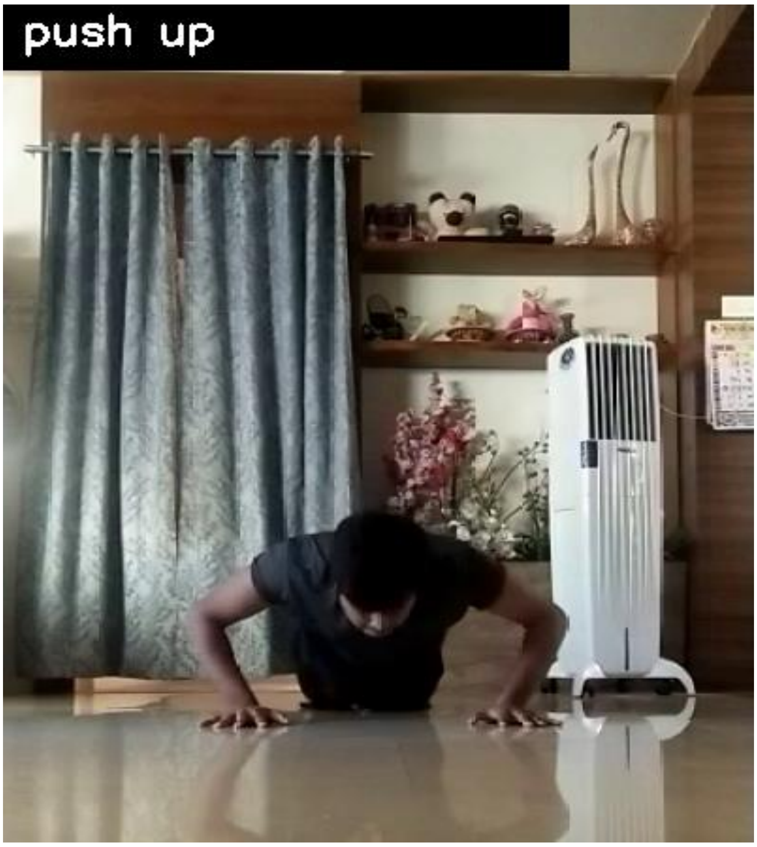
OUTPUT 1



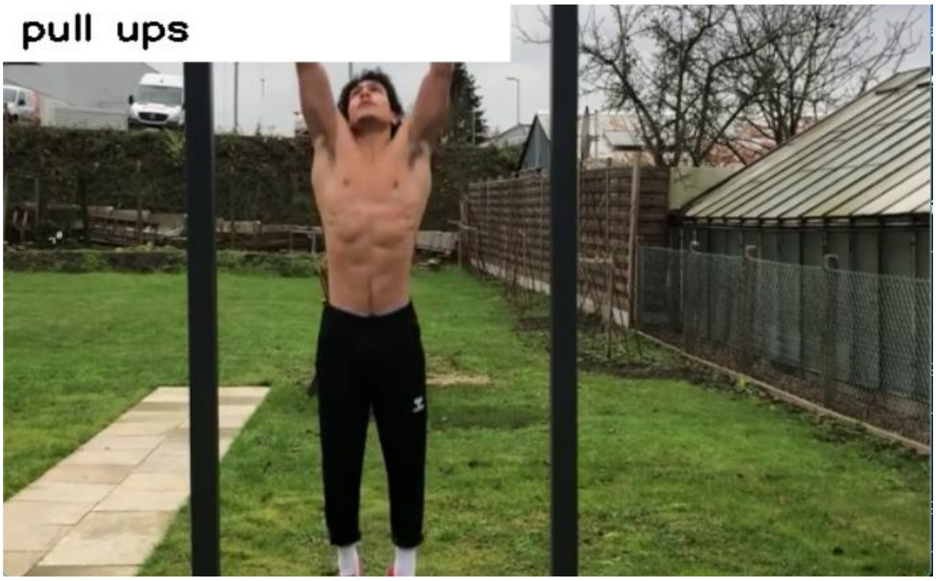
OUTPUT 2



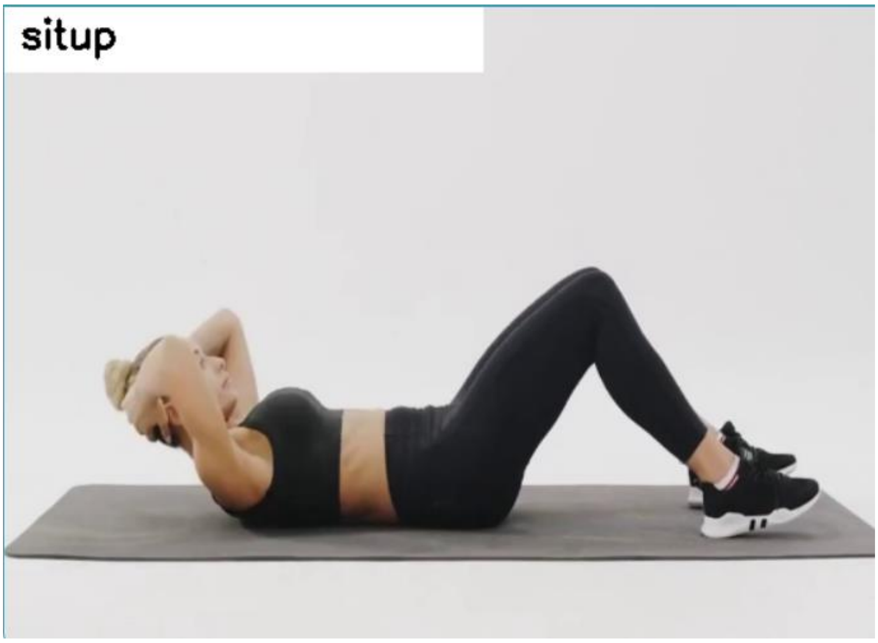
OUTPUT 3



OUTPUT 4



OUTPUT 5



6. CONCLUSION

We have proposed a Human Activity Recognition system using machine learning which deals with identification of activity based on its nature as normal or suspicious.

6.1 FUTURE ENHANCEMENT

In future enhancement, If such activity of anomalous nature is identified an immediate alert notification is sent to authority due to which further disheartening consequences can be minimized.

BIBLIOGRAPHY

1. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge L. Reyes Ortiz (2012). Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine. Springer International Workshop on Ambient Assisted Living. Lecture notes in Computer Science. Vol (7657), pp 216- 223.

2. Jun Liu, Amir Shahroudy, Dong Xu, Gang Wang (2016). Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. European Conference on Computer Vision., pp 816-833. Vol (9907) Oyelade, Oladipupo, Obagbuwa (2010). Application of k-Means Clustering algorithm for prediction of Students'

3. Academic Performance. International Journal of Computer Science and Information Security,7(1).292-295

4. JuhaVesanto (1999). SOM-Based Data Visualization Methods. Intelligent Data Analysis, Laboratory of Computer and Information Science, Helsinki University of Technology, P. O. Box 5400, FIN-02015 HUT, Finlandvol. 3(2), pp. 111-126.

5. Ivan Viola (2010). Information Theory in Computer Graphics and Visualization.