PROJECT ON

HUMAN ACTION RECOGNITION

By:-

SAGAR MONDAL [11901621034]

DIVYA GUPTA [11901621022]

PRIYASHREE NANDI [11901621030]

AMIT KUMAR RAM [11901621035]

SUBHANKAR ROY [11901621027]

A Comprehensive report has been submitted in partial fulfilment of the requirement for the degree of

"BACHELOR OF TECHNOLOGY"

IN ELECTRICAL ENGINEERING

SILIGURI INSTITUTE OF TECHNOLOGY

AFFILIATED TO

Maulana Abul Kalam Azad University of Technology

Under the supervision of

Mr. Ripam Kundu

Sikharthy Infotech Pvt. Ltd.

ACKNOWLEDGEMENT

We are thankful to our Institute ***Siliguri Institute Of Technology*** for providing necessary facilities. It is a great privilege for us to express our profound gratitude to **HOD Mr. ARUP DAS** and other faculties for their constant guidance, valuable suggestions, supervision and inspiration throughout the course work. We would like to express our gratitude to our supervisor for his kind cooperation and encouragement which helped us to do this project. The valuable guidance and interest taken by our respected mentors have been a motivation and source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We are also indebted to head of the department **Mr. Arup Das** of Electrical Engineering Technology for permitting us to pursue the project. We would like to take this opportunity to extend our sincere thank to our training teacher **Mr. Ripam Kundu** , also other technical staff members of the department who gave us their precious time and tok care of all the facilities during this entire period.

THANK YOU.

_____

_____

_____

_____

_____

CERTIFICATE OF APPROVAL

The project report titled **"HUMAN ACTION RECOGNITION"** prepared by **Sagar Mondal,** Roll No:11901621034; **Divya Gupta.** Roll No:11901621022; **Priyashree Nandi,** Roll No:11901621030; **Subhankar Roy,** Roll No: 11901621027; Amit **Kumar Ram,** Roll No:11901621035; is hereby approved and certified as a creditable study in technological subjects performed in a way sufficient for its acceptance for partial fulfilment of the degree for which it is submitted. Under the direct supervision and guidance of **Mr.RIPAM KUNDU.** It is to be understood that by this approval, the undersigned do not, necessarily endorse or approve any statement made, opinion expressed or conclusion drawn Thereon, but approve the project only for the purpose for which it is submitted.

_____	_____
Shilpi ma'am & Dipayan sir	**Mr. Ripam Kundu.**
(Trainer of the Head)	(Name of Trainer of project)
Date: 10/02/2023	Date: 10/02/2023
_____	_____
Mr. Mithun Chakraborty.	**Mr. Arup Das.**
(Head of the College)	(Head of the department)

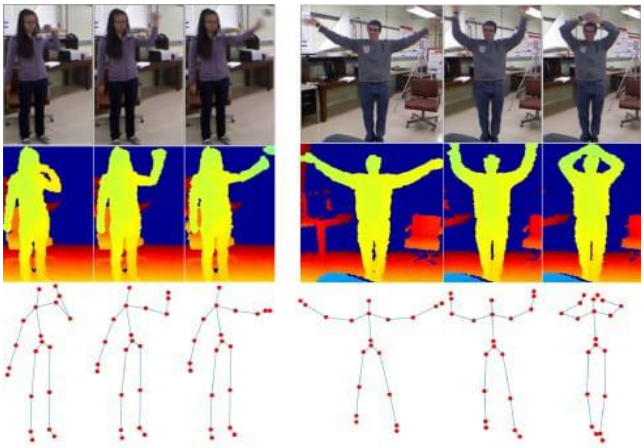
ABSTRACT

Human Activity Recognition is an active field of research and scientific development in which various models have been proposed using different methods for identification and categorization of activities using Machine Learning. 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 Convolution 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. According to the identified nature an alert is sent through server to the authority concerning the happening of anomalous activity taking place at real time. Due to such application of this model many harmful activities can be avoided or at least negative consequences of such activities can be minimized.

Contents

INTRODOCTION

First of we need to understand - what is the machine learning? Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behaviour. Artificial intelligence systems are used to perform complex tasks in a way that is similar to how humans solve problems.



The aim of our project is to understand human behaviour and assign a label to each action. It has a wide range of applications, and therefore has been attracting increasing attention in the field of computer vision. Take a video or image input and identify the action that is being performed by the person in the video. The actions can be anything from applying makeup, dancing, singing, football, swimming etc. Human action recognition is the very interesting problem where the main goal is to analyse a video and recognize the action being performed by the person in the video. There are two type of action Some simple action like standing can be identified by using just a single frame for more complex action such as walking running bending or falling might require more than one frames to identify the action correctly.

HUMAN ACTIVITIES ARE DIVIDED IN DIFFERENT CATEGORY

The goal of human activity recognition is to examine activities from video sequences or still images. Motivated by this fact, human activity recognition systems aim to correctly classify input data into its underlying activity category. Depending on their complexity, human activities are categorized into:

(i) Gestures

(ii) Atomic actions

(iii) human-to-object or human-to-human interactions

(iv) Group actions

(v) Behaviours and

(vi) Events.

Gestures are considered as primitive movements of the body parts of a person that may correspond to a particular action of this person .

Atomic actions are movements of a person describing a certain motion that may be part of more complex activities.

Human-to-object or human-to-human interactions are human activities that involve two or more persons or objects.

Group actions are activities performed by a group or persons.

Human behaviours refer to physical actions that are associated with the emotions, personality, and psychological state of the individual .Finally, **Events** are high-level activities that describe social

actions between individuals and indicate the intention or the social role of a present.

BENEFITS

Activity recognition is the basis for the development of many potential applications in health, wellness, or sports:

MONITOR HEALTH

Analyze the activity of a person from the information collected by different devices.

DISCOVER ACTIVITY PATTERNS

Discover which are the variables that determine which activity a person is doing.

DETECT ACTIVITY

Calculate a predictive model that can recognize a person's activity from the signals received by the sensors.

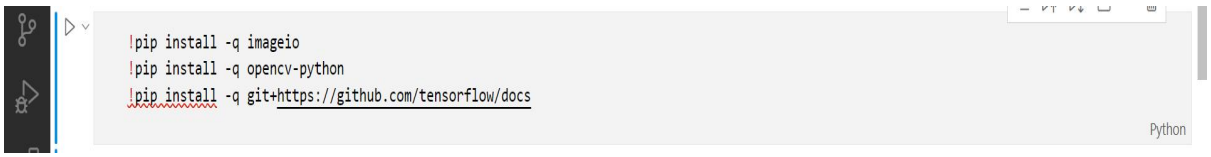
IMPROVE WELLBEING

Design individualized exercise tables to improve the health of a person.

INTRODUCTION TO UCF101 DATASET

- Currently the largest dataset of human actions.
- Consists of 101 action classes distributed over 13k clips and 27 hours
  of video data.
- Consists of realistic user-uploaded videos where most of them are taken from random YouTube.
- Currently the most challenging dataset of actions due to its large number of classes, a large number of clips, and also unconstrained
  nature of such clips.
- Aims to encourage further research into action recognition by learning and exploring new realistic action categories.

CODE IMPLEMENTATION



```
!pip install -q imageio
!pip install -q opencv-python
!pip install -q git+https://github.com/tensorflow/docs
```

Python

At first we have to install top libraries

Now we can define the uses of libraries in simple way-

imageio: Imageio is a Python library that provides an easy interface to read and write a wide range of image data, including animated images, volumetric data, and scientific formats. It is cross-platform.

Open cv: **OpenCV-Python** is a library of Python bindings designed to solve computer vision problems. cv2.circle () method is used to draw a circle on any image.

tensorflow:The TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining. Using production-level tools to automate and track model training over the lifetime of a product, service, or business process is critical to success.

```
# TensorFlow and TF-Hub modules.
from absl import logging

import tensorflow as tf
import tensorflow_hub as hub
from tensorflow_docs.vis import embed

logging.set_verbosity(logging.ERROR)

# Some modules to help with reading the UCF101 dataset.
import random
import re
import os
import tempfile
import ssl
import cv2
import numpy as np

# Some modules to display an animation using imageio.
import imageio
from IPython import display

from urllib import request  # requires python3
```

Now we are importing liberis for our uses.

Details of libraries:

Tensorflow: The Subclassing API provides a define-by-run interface for advanced research. Create a class for your model, then write the forward pass imperatively. Easily author custom layers, activations, and training loops. Run the “Hello World” example below, then visit the tutorials to learn more.

Random: **In Python**, **random** numbers are not generated implicitly; therefore, it provides a **random** module in order to generate **random** numbers explicitly. **random** module **in Python** is used to create **random** numbers.

Re: Python has **a built-in package called** re, which **can be used to work with Regular Expressions.**

Os: The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules.

Tempfile: Tempfile is a **Python module** used in a situation, where we need to read multiple files, change or access the data in the file, and gives output files based on the result of processed data. Each of the output files produced during the program execution was no longer needed after the program was done.

Ssl: Requests verifies SSL certificates for HTTPS requests, just like a web browser. SSL Certificates are small data files that digitally bind a cryptographic key to an organization’s details.

Cv2: The cv2 is a **cross-platform library** designed to solve all computer vision-related problems. We will look at its application and work later in this article. But first, let us try to get an overview of the function through its definition.

Numpy: **Numpy** is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with **Python**.

Urllib: Urllib package is the URL handling module for python. It is used to fetch URLs (Uniform Resource Locators). It uses the urlopen function and is able to fetch URLs using a variety of different protocols.

Helper Function for the UFC101 dataset.

Code:

```
# Utilities to fetch videos from UCF101 dataset
UCF_ROOT = "https://www.crcv.ucf.edu/THUMOS14/UCF101/UCF101/"
_VIDEO_LIST = None
_CACHE_DIR = tempfile.mkdtemp()
unverified_context = ssl._create_unverified_context()

deflist_ucf_videos():
  global_VIDEO_LIST
  ifnot_VIDEO_LIST:
    index = request.urlopen(UCF_ROOT,
context=unverified_context).read().decode("utf-8")
    videos = re.findall("(v_[\w_]+\.avi)", index)
    _VIDEO_LIST = sorted(set(videos))
  returnlist(_VIDEO_LIST)

deffetch_ucf_video(video):
  cache_path = os.path.join(_CACHE_DIR, video)
  ifnotos.path.exists(cache_path):
    urlpath = request.urljoin(UCF_ROOT, video)
    print("Fetching %s =>%s" % (urlpath, cache_path))
    data = request.urlopen(urlpath,
context=unverified_context).read()
    open(cache_path, "wb").write(data)
  returncache_path

defcrop_center_square(frame):
  y, x = frame.shape[0:2]
  min_dim = min(y, x)
  start_x = (x // 2) - (min_dim // 2)
  start_y = (y // 2) - (min_dim // 2)
  returnframe[start_y:start_y+min_dim,start_x:start_x+min_dim]
```

```
defload_video(path, max_frames=0, resize=(224, 224)):
  cap = cv2.VideoCapture(path)
  frames = []
  try:
    whileTrue:
      ret, frame = cap.read()
      ifnotret:
        break
      frame = crop_center_square(frame)
      frame = cv2.resize(frame, resize)
      frame = frame[:, :, [2, 1, 0]]
      frames.append(frame)

      iflen(frames) == max_frames:
        break
  finally:
    cap.release()
  returnnp.array(frames) / 255.0

defto_gif(images):
  converted_images = np.clip(images * 255, 0, 255).astype(np.uint8)
  imageio.mimsave('./animation.gif', converted_images, fps=25)
  return embed.embed_file('./animation.gif')
```

Get the kinetics-400 labels:

Code:

```
KINETICS_URL = "https://raw.githubusercontent.com/deepmind/kinetics-
i3d/master/data/label_map.txt"
withrequest.urlopen(KINETICS_URL) asobj:
  labels = [line.decode("utf-8").strip() forlineinobj.readlines()]
print("Found in total %d labels." % len(labels))
```

Output:

```
Found in total 400 labels.
```

Get UCF101 Dataset:

Code:

```
ucf_videos = list_ucf_videos()
categories = {}
forvideoinucf_videos:
```

```
  category = video[2:-12]
  ifcategorynotincategories:
    categories[category] = []
  categories[category].append(video)
print("Found in total %d videos in overall %d categories." %
(len(ucf_videos), len(categories)))

print("\n")
head1 = "CATEGORY"
head2 = "No. of Videos"
head3 = "Details"
print("",head1,"\t  ",head2,"\t\t\t",head3)
forcategory, sequencesincategories.items():
  summary = ", ".join(sequences[:2])
  print("%-20s    %4d           %s, ..." % (category, len(sequences),
summary))
```

Output:

```
Output exceeds the size limit. Open the full output data in a text editor
Found in total 13320 videos in overall 101 categories.


  CATEGORY         No. of Videos                    Details
ApplyEyeMakeup           145          v_ApplyEyeMakeup_g01_c01.avi, v_ApplyEyeMakeup_g01_c02.avi, ...
ApplyLipstick            114          v_ApplyLipstick_g01_c01.avi, v_ApplyLipstick_g01_c02.avi, ...
Archery                  145          v_Archery_g01_c01.avi, v_Archery_g01_c02.avi, ...
BabyCrawling             132          v_BabyCrawling_g01_c01.avi, v_BabyCrawling_g01_c02.avi, ...
BalanceBeam              108          v_BalanceBeam_g01_c01.avi, v_BalanceBeam_g01_c02.avi, ...
BandMarching             155          v_BandMarching_g01_c01.avi, v_BandMarching_g01_c02.avi, ...
BaseballPitch            150          v_BaseballPitch_g01_c01.avi, v_BaseballPitch_g01_c02.avi, ...
BasketballDunk           131          v_BasketballDunk_g01_c01.avi, v_BasketballDunk_g01_c02.avi, ...
Basketball               134          v_Basketball_g01_c01.avi, v_Basketball_g01_c02.avi, ...
BenchPress               160          v_BenchPress_g01_c01.avi, v_BenchPress_g01_c02.avi, ...
Biking                   134          v_Biking_g01_c01.avi, v_Biking_g01_c02.avi, ...
Billiards                150          v_Billiards_g01_c01.avi, v_Billiards_g01_c02.avi, ...
BlowDryHair              131          v_BlowDryHair_g01_c01.avi, v_BlowDryHair_g01_c02.avi, ...
BlowingCandles           109          v_BlowingCandles_g01_c01.avi, v_BlowingCandles_g01_c02.avi, ...
BodyWeightSquats         112          v_BodyWeightSquats_g01_c01.avi, v_BodyWeightSquats_g01_c02.avi, ...
Bowling                  155          v_Bowling_g01_c01.avi, v_Bowling_g01_c02.avi, ...
BoxingPunchingBag        163          v_BoxingPunchingBag_g01_c01.avi, v_BoxingPunchingBag_g01_c02.avi, ...
BoxingSpeedBag           134          v_BoxingSpeedBag_g01_c01.avi, v_BoxingSpeedBag_g01_c02.avi, ...
```

Fetch a random video:

Code:

```
video_path = fetch_ucf_video("v_LongJump_g01_c01.avi")
sample_video = load_video(video_path)
sample_video1 = load_video(video_path)[:100]
sample_video.shape
```

Output:

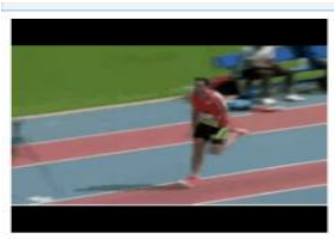
```
Fetching https://www.crcv.ucf.edu/THUMOS14/UCF101/UCF101/v_LongJump_g01_c01.avi => /tmp/tmpvodu3rj3/v_LongJump_g01_c01.avi

(143, 224, 224, 3)
```

Code for gif output:

```
to_gif(sample_video1)
```

Output:



Predict from the video:

Code:

```
i3d = hub.load("https://tfhub.dev/deepmind/i3d-kinetics-
400/1").signatures['default']
making labels and gating output:
code:
defpredict(sample_video):
  # Add a batch axis to the to the sample video.
  model_input = tf.constant(sample_video,
dtype=tf.float32)[tf.newaxis, ...]

  logits = i3d(model_input)['default'][0]
  probabilities = tf.nn.softmax(logits)

  print("Top 5 actions:")
  foriinnp.argsort(probabilities)[::-1][:5]:
    print(f"  {labels[i]:22}: {probabilities[i] * 100:5.2f}%")
final output:
predict(sample_video)
```

output:

```
Top 5 actions:
  long jump              : 62.05%
  triple jump            : 37.88%
  hurdling               :  0.06%
  pole vault             :  0.01%
  high jump              :  0.00%
```

OVERVIEW

From this project we learn about Human activity recognition (HAR) can be referred to as the art of identifying and naming activities using Artificial Intelligence (AI) from the gathered activity raw data by utilizing various sources (so-called libraries).

Examples : imageio ,Opencv, tensorflow, Random, Re, Os, Tempfile, Sal, Cv2, Numpy, Urllib.

Problem Statement

Activity detection is a major problem in smart videos surveillance. It is a fundamental problem in computer vision, i.e. to detect the activity of human in surveillance videos. These applicants need real-time detection performance, but it is generally very time consuming to detect the actual activity. The time consumption is due to the heavy size of the video clip of surveillance and low computation power of these systems. This heavy size is because of the resolution of the cctv camera. It becomes important to reduce the resolution of video clip and to detect what activity is been performed by the subjects. There are many solutions provided in deep learning until now, but none of them are efficient when there are lots of details in the video and it becomes difficult to detect the actual activity. In such case if rest of the details are compressed, it will be easy to apply attention to the actual activity.

Solution Approch

Action detection is one of the most challenging tasks in video processing. It can be useful in security systems and closed-circuit television (CCTV), facial emotion recognition software, sport event analytics, behaviour observation, statistics gathering, etc.Deep learning is an area of machine learning that uses feature learning techniques instead of task-specific algorithms. It uses multi-layered artificial neural networks that work similarly to neural networks in the human brain. But unlike the brain, a neural network is divided into separate layers each with a defined direction of data processing. A network with more than two layers is called a deep neural network (DNN).

Contribution

Sagar Mondal (Group leader): Code section

Subhankar Roy: Data Collection

Amit Kumar Ram: Code Testing & Libeary Importer

Priyashree Nandi: Report Writing & Output checking of the code

Divya Gupta: Power Point Presentation & UCF Dataset

Conclusion

Human activity analysis is a popular activity in the growing industry and we have applied different machine learning algorithm. Comparative study performed among the applied various techniques kNN, SVM, Random forest, Neural Networks, Logistic regression and Naïve Bays. In them, Logistic Regression and neural network gave good results whereas Naive Bays result was not good. The implementation of Neural Network on Python gave better results than the one provided in the Orange tool. The limitations of this work is though the efficiency of neural network is good, the model is not dynamic. The inability of getting trained with real time data will force us to train the model every time new data comes. In future, these results can be used for making smart watches and similar devices which can track a user's activity and notify him/her of the daily activity log. They can also be used for monitoring elderly people, prison inmates, or anyone who needs constant supervision.

References

[1] https://www.neuraldesigner.com/solutions/activity-recognition

[2] https://thecleverprogrammer.com/2021/01/10/human-activity-recognition-with-machine-learning/

[3] https://slideplayer.com/amp/10937846/

[4] https://upscwithnikhil.com/article/ethics/human-actions#:~:text=Human%20actions%20are%20defined%20as,one%20day%20by%20armed%20robbers

[5]https://www.frontiersin.org/articles/10.3389/frobt.2015.00028/full#:~:text=Depending%20on%20their%20complexity%2C%20human,%3B%20and%20(vi)%20event

[6] https://youtu.be/dQAwXaxczkk

Our Github link:

https://github.com/GSagarMandal/Action_Detection_ML.git