A MAJOR PROJECT REPORT ON

RECOGNITION OF HUMAN ACTIVITY USING ENSEMBLE LEARNING OF MULTIPLE CONVOLUTIONAL NEURAL NETWORK

Submitted in the partial fulfilment of the requirements for the award of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

SUBMITTED BY

Gyanaji Harshitha	**Bharadwaj Yagnik Rao**	**Gandla Uday Kiran**
20BK1A1244	**20BK1A1214**	**21BK5A1204**

Under the esteemed guidance of
Dr. R. KALYANI

Giving Wings To Thoughts

DEPARTMENT OF INFORMATION TECHNOLOGY

St. Peter’s Engineering College (UGC Autonomous)

Approved by AICTE, New Delhi, and NAAC with ‘A’ Grade,

Affiliated to JNTU, Hyderabad, Telangana

2020-2024 DEPARTMENT OF INFORMATION TECHNOLOGY

Create an Application

We assume you are in your project folder. In our main “myproject” folder, the same folder then manage.py −

```
$ python manage.py startapp myapp
```

You just created myapp application and like project, Django create a “myapp” folder with the application structure −

```
myapp/
   __init__.py
   admin.py
   models.py
   tests.py
   views.py
```

__init__.py − Just to make sure python handles this folder as a package.

admin.py − This file helps you make the app modifiable in the admin interface.

models.py − This is where all the application models are stored.

tests.py − This is where your unit tests are.

views.py − This is where your application views are.

Get the Project to Know About Your Application

At this stage we have our "myapp" application, now we need to register it with our Django project "myproject". To do so, update INSTALLED_APPS tuple in the settings.py file of your project (add your app name) −

```
INSTALLED_APPS = (
   'django.contrib.admin',
   'django.contrib.auth',
   'django.contrib.contenttypes',

   'django.contrib.sessions',

   'django.contrib.messages',

   'django.contrib.staticfiles',

   'myapp',

)
```

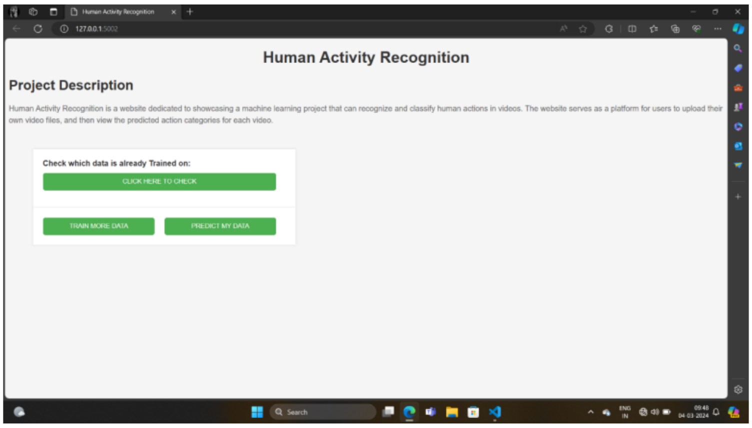
SOURCE CODE

```
from flask import Flask, request,render_template
from flask_restful import Resource, Api
from flask_httpauth import HTTPBasicAuth
from werkzeug.security import generate_password_hash, check_password_hash
from datapreparation.data_prep import data_prep
from train import train_class
from predict import predict_class
import json
from werkzeug.utils import secure_filename
import os
import shutil

app=Flask(_name_)
app.config['UPLOAD_FOLDER']='raw'

api=Api(app)
data_prep_instance=data_prep()
train_instance=train_class()
predict_instance=predict_class()
with open('config.json', 'r') as f:
    data = json.load(f)

auth=HTTPBasicAuth()
users={
    "admin":generate_password_hash("okayboss")
}

@auth.verify_password
def verify_password(username,password):
    if username in users and check_password_hash(users.get(username),password):
        return username
@app.route('/')
# @auth.login_required
def index():
    return render_template('index.html', msg="I'm working")
@app.route('/data_present',  methods = ['GET', 'POST'])
def trained_data():
    listOfTrained=data_prep_instance.data_trained(data['csv_path'])
    return render_template('index.html', list_status=listOfTrained)
@app.route('/train',  methods = ['GET', 'POST'])
def train():
    return render_template('train.html')
@app.route('/upload',  methods = ['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        files = request.files.getlist('files[]')
        folder_name = request.form['text']

 create_folder_path=app.config['UPLOAD_FOLDER']
    if not os.path.exists(create_folder_path):
        # create the folder
        os.mkdir(create_folder_path)
        print("Folder created successfully.")
    else:
        print("Folder already exists.")

         for file in files:
        filename = secure_filename(file.filename)
        try:
            directory = app.config['UPLOAD_FOLDER'] +'/'+ folder_name
            os.mkdir(directory)
        except FileExistsError:
            pass
        file.save(os.path.join(directory, filename))

data_prep_instance.process_data(data['images_dir'],data['csv_path'],data['pose_model'],
data['body_dict'],'train')


        # check if the folder exists before attempting to delete it
    if os.path.exists(create_folder_path):
        # use shutil.rmtree() function to delete the folder and all its contents
        shutil.rmtree(create_folder_path, ignore_errors=True)
        print("Folder deleted successfully.")
    else:
        print("Folder does not exist.")

    return render_template('train.html', file_status=""Files uploaded successfully!")


@app.route('/data_prep_fun',  methods = ['GET', 'POST'])
```

```
def data_prep_fun():
    acc=train_instance.train_model(data['csv_path'])
    return render_template('train.html', data_prep_fun_status="Model accuracy is: " + 
str(acc))



@app.route('/predict',  methods = ['GET', 'POST'])
def predict():
    return render_template('predict.html')



@app.route('/upload_predict',  methods = ['GET', 'POST'])
def upload_predict_file():
    if request.method == 'POST':
        file = request.files['file']
        folder_name = 'unknow'
    os.mkdir('upload')
    filename = secure_filename(file.filename)
    try:
        directory = 'upload' +'/'+ folder_name
        os.mkdir(directory)
    except FileExistsError:
        pass
    file.save(os.path.join(directory, filename))


data_prep_instance.process_data(data['predict_video'],data['predict_csv'],data['pose_m
odel'],data['body_dict'],'predict')


    if os.path.exists('upload'):
        # use shutil.rmtree() function to delete the folder and all its contents
        shutil.rmtree('upload', ignore_errors=True)
        print("Folder deleted successfully.")
    else:
        print("Folder does not exist.")

    return render_template('predict.html', file_status="Predict File uploaded successfully!")



@app.route('/prdict_data',  methods = ['GET', 'POST'])
def prdict_data():
    result=predict_instance.predict_model(data['predict_csv'])
    return render_template('predict.html', predict_status=result)



@app.route('/backtohome', methods = ['GET', 'POST'])
def backtohome():
    return render_template('index.html')



if _name== 'main_':
    app.run(debug=True,host='0.0.0.0',port=5002
```

EXPERIMENTAL RESULTS

Home Page :



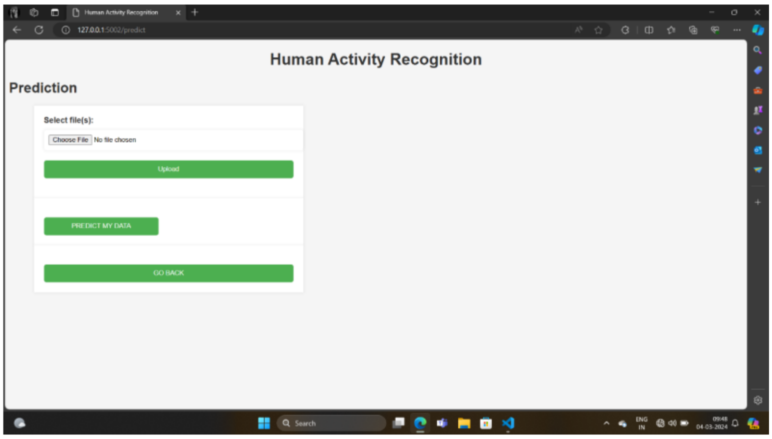
Human Activity Recognition

Project Description

Human Activity Recognition is a website dedicated to showcasing a machine learning project that can recognize and classify human actions in videos. The website serves as a platform for users to upload their own video files, and then view the predicted action categories for each video.

Check which data is already Trained on:

Prediction data Uploading:



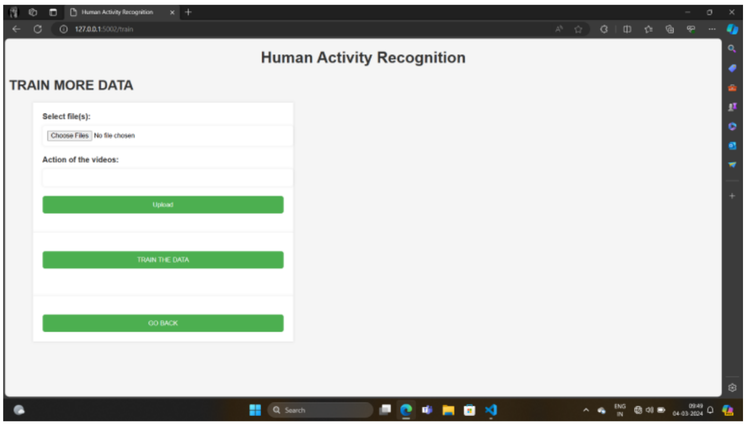
Human Activity Recognition

Prediction

Select file(s):

Choose File No file chosen

Data Training:



Human Activity Recognition

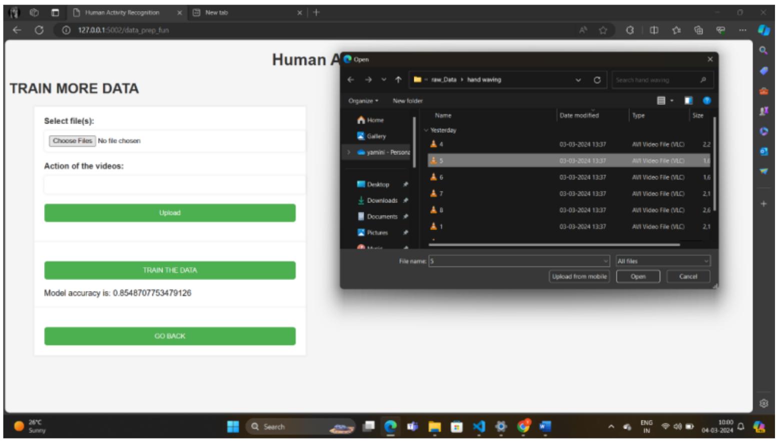
TRAIN MORE DATA

Select file(s):

Choose Files No file chosen

Action of the videos:

Data Uploading :



Human A

TRAIN MORE DATA

Select file(s): No file chosen

Action of the videos:

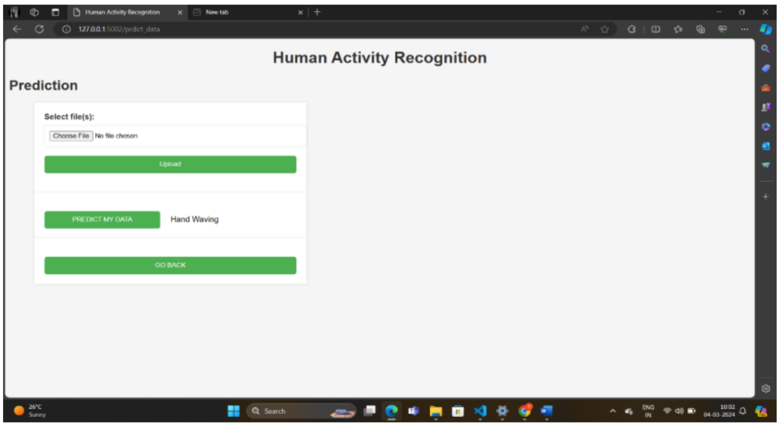
Model accuracy is: 0.8548707753479126

Name	Date modified	Type	Size
Yesterday			
4	03-03-2024 13:37	AVI Video File (VLC)	2,2
5	03-03-2024 13:37	AVI Video File (VLC)	1,6
6	03-03-2024 13:37	AVI Video File (VLC)	1,6
7	03-03-2024 13:37	AVI Video File (VLC)	2,1
8	03-03-2024 13:37	AVI Video File (VLC)	2,6
1	03-03-2024 13:37	AVI Video File (VLC)	2,1

File name: 5

All files

Data Prediction:



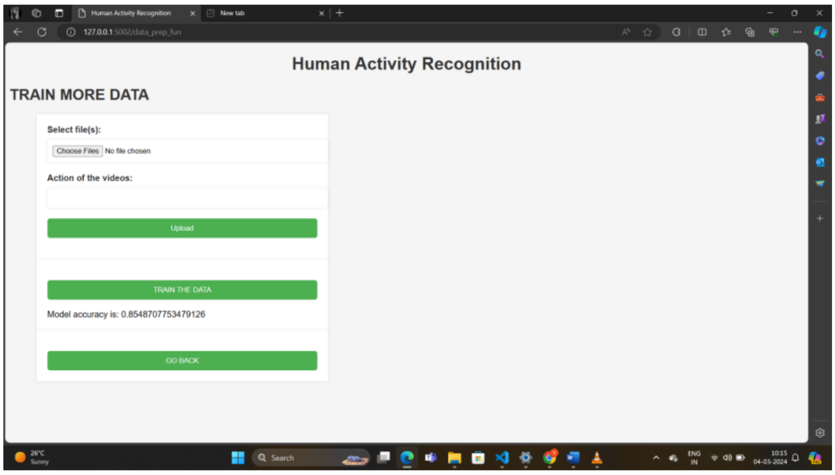
Human Activity Recognition

Prediction

Select file(s): No file chosen

Hand Waving

Model accuracy:



Human Activity Recognition

TRAIN MORE DATA

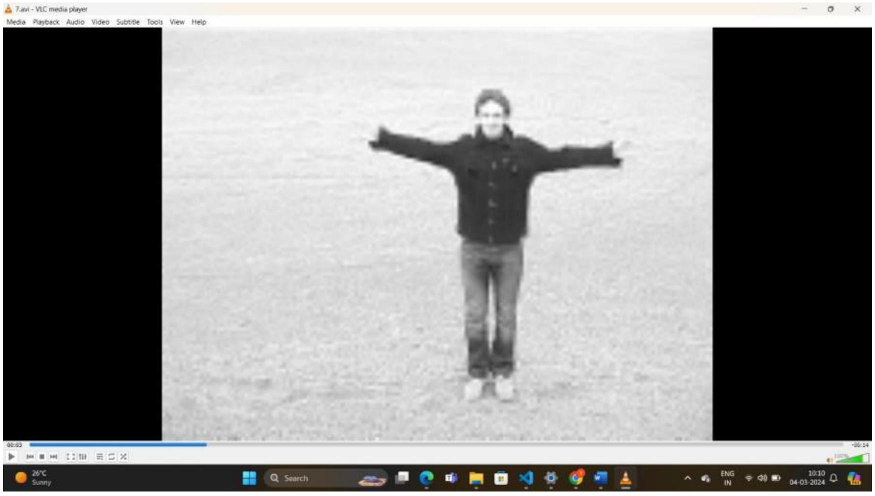
Select file(s): No file chosen

Action of the videos:

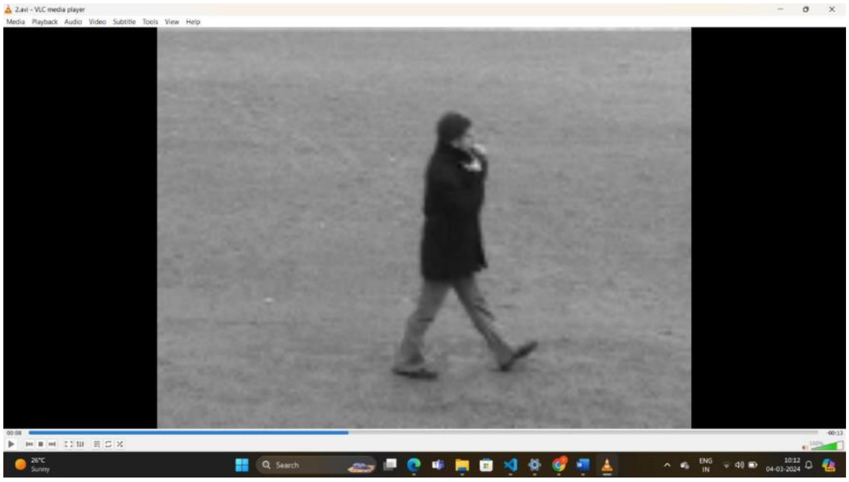
Model accuracy is: 0.8548707753479126

Data Input:

Hand waving:

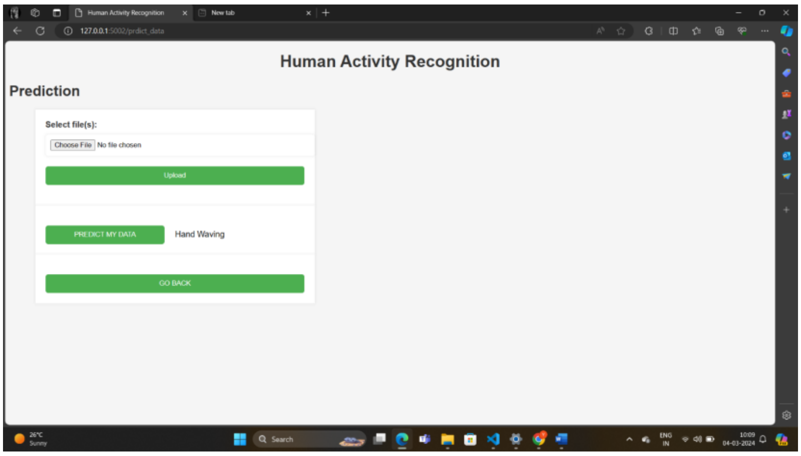


Walking:



Output:

Prediction:



Human Activity Recognition

Prediction

Select file(s):

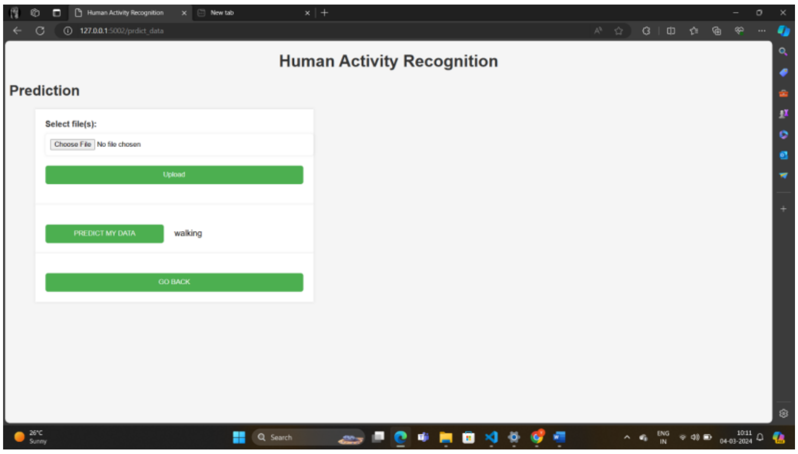
Choose File No file chosen

Upload

PREDICT MY DATA Hand Waving

GO BACK

Walking:



Human Activity Recognition

Prediction

Select file(s):

Choose File No file chosen

Upload

PREDICT MY DATA walking

GO BACK

CONCLUSION:

We presented three CNN based models as well as their ensembles for WISDM dataset of HAR. It was found that the performance of the ensemble model is better than that of individual models. One of the ensemble model performed better than the methods in the literature. In the dataset we used, we see a class imbalance such that we have 38% samples for walking class but hardly 5% for sitting and standing. In future, the results might be improved even more, if we can remove the class imbalance from dataset. Moreover, currently an ensemble of average of the three models is created but for further exploration, a future direction can be performing weighted ensemble learning such that the best performing model has the most effect in the ensemble

REFERENCES

[1] T. Plotz, N. Y. Hammerla, and P. L. Olivier, “Feature learning for ¨ activity recognition in ubiquitous computing,” in 22nd International Joint Conference on Artificial Intelligence, 2011.

[2] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, “LSTM networks for mobile human activity recognition,” in International Conference on Artificial Intelligence: Technologies and Applications (ICAITA 2016), 2016.

[3] L. K. Hansen and P. Salamon, “Neural network ensembles,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993–1001, 1990.

[4] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in 24th International Joint Conference on Artificial Intelligence, 2015.

[5] M. Panwar, S. R. Dyuthi, K. C. Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, and G. R. Naik, “CNN based approach for activity recognition using a wrist-worn accelerometer,” in 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2017, pp. 2438–2441.

[6] S. W. Pienaar and R. Malekian, “Human activity recognition using LSTM-RNN deep neural network architecture,” in 2019 IEEE 2nd Wireless Africa Conference (WAC). IEEE, 2019, pp. 1–5.

[7] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensorbased activity recognition: A survey,” Pattern Recognition Letters, vol. 119, pp. 3–11, 2019

[8] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, “A robust human activity recognition system using smartphone sensors and deep learning,” Future Generation Computer Systems, vol. 81, pp. 307–313, 2018.

[9] J. Sun, Y. Fu, S. Li, J. He, C. Xu, and L. Tan, “Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors,” Journal of Sensors, vol. 2018, pp. 1–10, 09 2018.

[10] K. Xia, J. Huang, and H. Wang, “LSTM-CNN Architecture for Human Activity Recognition,” IEEE Access, vol. 8, pp. 56 855–56 866, 2020.

[11] D. Ravi, C. Wong, B. Lo, and G. Yang, “Deep learning for human activity recognition: A resource efficient implementation on low-power devices,” in 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), 2016, pp. 71–76.

[12] M. Wozniak, M. Gra ´ na, and E. Corchado, “A survey of multiple classifier ˜ systems as hybrid systems,” Information Fusion, vol. 16, pp. 3–17, 2014.

[13] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” ACM SigKDD Explorations Newsletter, vol. 12, no. 2, pp. 74–82, 2011.

[14] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” arXiv preprint arXiv:1412.6806, 2014.

[15] M. Lin, Q. Chen, and S. Yan, “Network in Network,” arXiv preprint arXiv:1312.4400, 2013.

[16] C. A. Ronao and S.-B. Cho, “Human activity recognition with smartphone sensors using deep learning neural networks,” Expert Systems with Applications, vol. 59, pp. 235 244,2016.[Online].Available:

http://www.sciencedirect.com/science/article/pii/S0957417416302056