Configuration Manual

MSc Research Project
Data Analytics

Bilge Su Erdogan
Student ID: 22196145

School of Computing
National College of Ireland

Supervisor: Mohammed Hasanuzzaman

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Bilge Su Erdogan
Student ID:	22196145
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Mohammed Hasanuzzaman
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	1090
Page Count:	4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Bilge Su Erdogan
22196145

1 Abstract

The aim of this project is to design a fall prediction system with a hybrid GCN-LSTM model that can be integrated with real-time systems and has higher accuracy and reliability than existing methods using visual data. The biggest contribution of this research is to reduce false positives with a novel approach by weighting the relationship of certain joints of the human body and their distance from the ground.

2 System Requirements

- RAM: 8GB at least 16 or higher preferred.
- EFPDS datasets. -> https://gram.web.uah.es/data/datasets/fpds/index.html
- OpenPose
- CUDA Nvidia preferred

3 Software Requirements

- Python Version: Python 3.8 or higher.
- Visual Studio Code

4 Installation Guide

4.1 Environment Installation

- Install 3.7 or higher version of Python
- Install OpenPose and CUDA if your workstation has a compatible GPU

4.2 Install Required Packages

All necessary packages must be installed. Sample commands below:

pip install opencv-python
pip install sklearn
pip install matplotlib

4.3 Necessarry Libraries:

```
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.optim import Adam
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
import os
import json
```

Figure 1: Import Commands

5 Data Preparation

5.1 Data Collection

In this project EFPDS dataset was used. There are three different zip folder train, validation, test. Since dataset includes thousands of images, downloading them by manually was a better solution. Official website is https://gram.web.uah.es/data/datasets/fpds/index.html . It is open dataset to public.

5.2 Data Preprocessing:

With this step all images will be prepared for OpenPose usage. With this class we are cleaning, resizing and sharpening images to get better keypoints of people on OpenPose. This class expects two input paths. It reads every image from input_folder and pre-processed images are saved on output_folder. Please give correct paths while using this class. In the dataset train has different folder under train as split1, split2 so I flattened all of them under train folder so every image was under same folder.
Do not forget to run this step each folder train, validation and test.

```
class FallPredictionPipeline:
    def __init__(self):
        self.data_preparation = DataPreparation(
            input_folder="./openpose/examples/train",
            output_folder="./openpose/examples/train/pre=processed_images"
        )
```

Figure 2: Data Preprocessing

5.3 Data Processing and Feature Extraction:

After having preprocessed images, we can use OpenPose to extract key points of people skeleton from images. To run OpenPose use this command:

```
bin\OpenPoseDemo.exe --image_dir train/preprocessed_images --write_json train/output --write_images train/processed --scale_number 4 --scale_gap 0.25 --model_pose BODY_25
```

This command creates key points and and processed images on output and processed folders. When you have keypoint outputs then we are ready to complete Feature Extraction step.

FeatureExtraction class also expects two different input path. While the input folder is used to read the key points created with OpenPose, the output_folder is used as the path where the calculated data will be saved as labels.

```
self.feature_extractor = FeatureExtractor(
    input_folder="./openpose/train/output",
    output_folder="./data/json_output/featureExtraction/train"
)
```

Figure 3: Feature Extraction

6 Model Implementation:

GCN and LSTM models are created here. The data obtained with the Feature Extraction step is given as input to the GCN model. The calculated labels and keypoints are given to the GCN model to learn the decision according to the FallDecision label. The outputs obtained from the GCN model are given as input to the LSTM model and it is aimed to complete the temporal learning here.

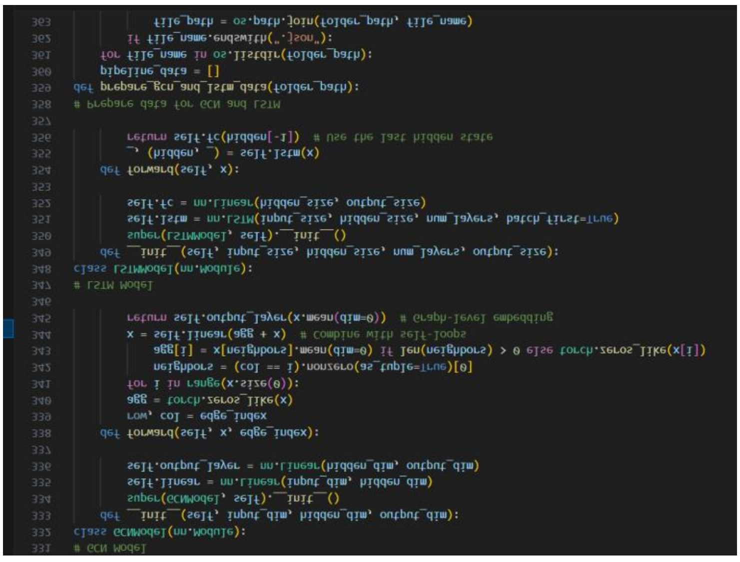

Figure 5: GCN & LSTM Model

7 Model Evaluation

There are couple of parameters to have better performance. The learning rate was initially set to 0.001. Training was performed with combinations of 32 and 64 dimensions and different numbers of epochs and different learning rates. Parameters can be changed on fallprediction.py file. Hyperparameter settings were made to find the most suitable values. Finally, among the evaluation criteria, F1 score, Accuracy, Precision and Recall metrics were considered. With these I was able to calculate False Positive rates as well.

Epoch	Dim	F1	Accuracy	Precision	Recall
10	64	80.92%	82.92%	79.82%	82.04%
10	32	83.45%	84.02%	76.82%	91.33%
25	64	84.06%	84.97%	79.02%	89.78%
25	32	84.15%	84.56%	76.92%	92.88%
50	64	82.76%	82.92%	74.63%	92.88%
50	32	80.58%	81.83%	76.24%	85.45%

Table 1: Results with different parameters

8 Execution of the Code

1. Download EFPDS Dataset from given URL
2. Unzip and flatten the subfolders into a folder
3. Open the folder in Visual studio Code
4. Run the Data Preparation step first with correct paths
5. Run OpenPose commands on your command window with correct paths
6. Run Feature Extraction with correct paths
7. Train & Evaluate Model with correct paths

9 Conclusion

According to the results obtained, it is possible to say that the study was completed successfully, and it was determined that it would make a great contribution to the literature for real-time fall detection applications with some improvements. When examined in detail, the important points contributing to the success of the model can be defined as the requirement of a powerful GPU and sufficient and well-detailed dataset. The layers and epoch numbers of the model are other factors affecting the success of the model. Thanks to the hybrid model, the strengths of both models were revealed in this study, and higher success was achieved compared to standalone models.