Human Motion Recognition Using Fusion Network

Cheung Chi Wang

1. Abstract

This work proposed a fusion model that takes heterogeneous inputs of accelerometer data, coordinates data, and image data for classifying and recognizing human motions. For accelerometer data, we use convolutional block attention module models (CBAM) to handle the data. Moreover, ResNet10 is used for training the model with image data and Bidirectional Long Short-Term Memory (Bi-LSTM) is used for taking coordinate data inputs. By comparing result of the fusion models and that of the model which take only single kind of input, we could investigate whether a fusion model perform better in the human motion recognition task. By using the data from Berkeley Multimodal Human Action Database (MHAD), an accuracy of 86.59% is achieved by the proposed fusion model while the accuracy of model that take image data or accelerometer data only is 76.02% and 80.47% respectively.

2. Introduction

Human Motion Recognition (HMR) has been a heated topic in AI for decades. From evaluating patients' movement who suffer from physical illness in the healthcare field to detecting suspicious human motion for security, this technology can be applied to various industrial fields. In the recent year, multi-various models have been introduced to tackle this problem. However, not so much of the architecture has adapted the fusion model method to solve the problem. Moreover, these models mostly used RGB images, Kinect images, and video files as the input data. **This project aims to provide a fusion model to recognize various types of human activities that take not only image or video input but also sensor data as input.**

By comparing the performance of a fusion model with that of other models, we would know whether a fusion model can perform better than the architecture solely composed of one model. In expectation, training different models with different inputs with a reasonable fusion network will be more robust than any used, single model in the fusion model. Due to time and data constraints, this research will only focus on

detecting and recognizing the motions which their data are available in the dataset that this project would be adopted.

3. Related work

First and foremost, we will look at various projects that used multimodal fusion networks. Multimodal fusion AI is not an edge-breaking machine learning approach; therefore, some of the projects have been implementing the multimodal approach to achieve better accuracy and performance. For instance, multimodal has been implemented to predict the need for hospitalization for patients infected with COVID-19 using medical history and treatment data [1]. The model takes five different types of input (Demographic information, Medications, Comorbidities, CPT codes, and Lab results) and gives the output after extracting features from the inputs.

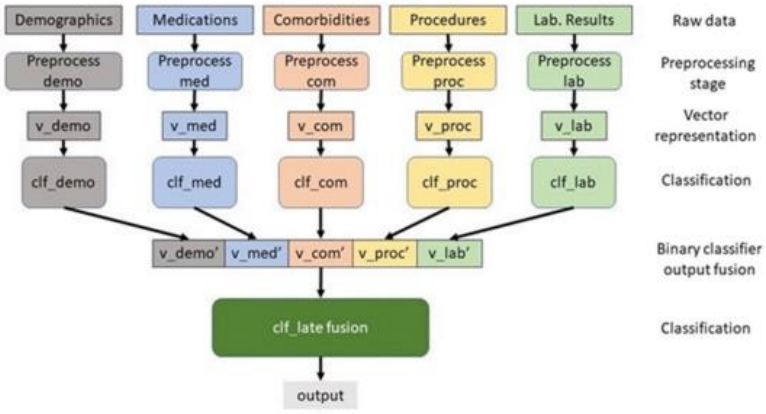


Figure 1
Fusion AI model architecture for prediction of need of hospitalization

Moreover, the fusion model can also be implemented for real-time event detection that takes images and text input. The research has used DenseNet to extract image features and BERT to extract text data [2].

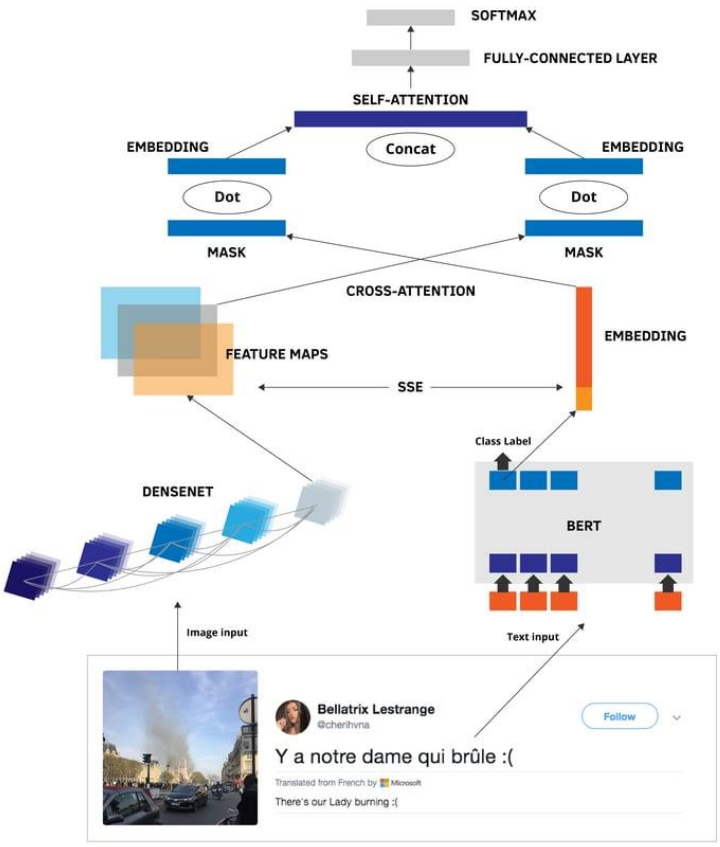


Figure 2
Framework of fusion model for detecting real-time events

All of the above examples have demonstrated the feasibility of the fusion model and multimodal approach in other fields of machine learning. Therefore, the fusion model can detect human action with improved accuracy.

Human motion recognition research has attracted attention because of its advantages of wide application in various fields of AI. In [3], [4], [5], the authors have explored different training methods for predicting human motion recognition, such as convolutional neural networks (CNNs), restricted Boltzmann RBM, recurrent neural network (RNN), long short-term memory (LSTM) network, bidirectional long short-term memory (Bi-LSTM) network, Support vector machine (SVM), and Multilayer perceptron (MLP). Especially for RNN, it is an excellent choice to model the complex dynamics of various actions in the video because its architecture allows it to store and access the long-range contextual information of a temporal sequence. This way, an RNN can learn to map from the history of previous inputs to each output. However, they are challenging to train due to the "vanishing gradient problem". LSTM and Bi-

LSTM have been proposed to solve these problems.

In [6], this is similar research to use a fusion model for human activity recognition. In this research, the authors have proposed a multimodal approach to detect human action by frame images, skeleton data, and accelerometer. In the architecture, Bi-LSTM has been used to detect the sequence of skeleton coordinates, Resnet10 is used for extracting features from frame images, and CNN, CBAM have been used to extract features from the spectrograms of accelerometer data. My work will be taking reference from this research and striving to improve the performance of the above work. In [5], the authors have proposed a method integrating vision data such as RGB, silhouettes, and skeletons. Using skeleton joint features in activity recognition of older adults has also been proposed in [7]. In addition, research integrating various sensor data such as accelerometers, gyroscopes, and magnetic field signals to recognize behaviour has been conducted [8].

Figure 3
RGB, depth, skeleton, IR image for HMR

Therefore, my work will use three kinds of input, namely skeleton images, skeleton joint features (coordinate), as well as accelerometer data.

One of the challenges for the multimodal approach is to align the time domain of the three different data. To put it precisely, the challenge is aligning accelerometer data with the other two kinds of skeleton data. In [9], the authors have proposed a CNN-based deep learning model to integrate the video and inertial sensing signal to detect

human activities. In this research, a continuous motion was expressed using a three-dimensional video volume and an input translated from a one-dimensional acceleration signal into a two-dimensional image form using a spectrogram. Therefore, we will need to normalize the video frames for data preparation and in [10], global contrast normalization (GCN), local normalization, and histogram equalization have been used for the normalization process.

4. Data

In this work, the data that will be used come from the Berkely MHAD datasets [11] and some of my data, such as video data for preliminary testing and construction of the skeleton data extractor.

The MHAD dataset contains 11 actions performed by seven male and five female subjects aged 23-30 years, except for one elderly subject. All the subjects performed five repetitions of each action, yielding about 660 action sequences corresponding to about 82 minutes of total recording time. The 11 actions available are Jumping in place, Jumping jacks, Bending - hands up all the way down, Punching (boxing), Waving - two hands, Waving - one hand (right), Clapping hands, Throwing a ball, Sit down then stand up, Sit down, and Stand up.

Various kinds of data are provided in the dataset, namely, accelerometer data, microphone data, motion capture data, camera data, and Kinect image data. The information, format, as well as sizes of the data, can be found in the table of Figure 4.

System	Data Format	Collected Data	Approximate Size per Subject
PhaseSpace Motion Capture	ASCII	XYZ 3D position of 43 markers UNIX time stamps Frame numbers	0.6 GB
Multi-Stereo Cameras	PGM ASCII	Bayer format (GRBG) UNIX time stamps	31 GB
Microsoft Kinect	PPM PGM ASCII	RGB color images 16-bit depth map UNIX time stamps	34 GB
Accelerometers (Shimmer)	ASCII	Acceleration in XYZ axes UNIX time stamps	4 MB
Microphones	WAV	Uncompressed audio at 48kHz	150 MB

Figure 4
Table of data information of the MHAD dataset

Multiview video and acceleration data from this dataset will be used. The multiview video data were in a 640 × 480 PGM file and consisted of 12 camera data points, while the three-axis accelerometer data (X, Y, Z axes) and the corresponding time stamps are contained in .txt files. Due to storage constraints, only front-view camera data (Cluster 1) would be used for simplicity.

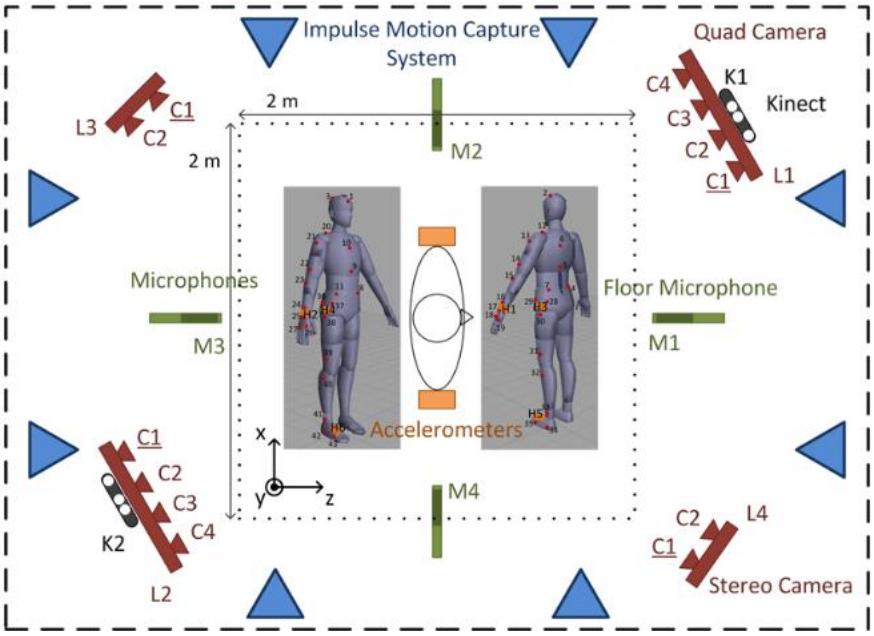


Figure 5
Diagram of the data acquisition system

In the research, eight actions from the dataset would be used for the input data that feed into the fusion models. The corresponding eight actions are *Jumping in place, Jumping jacks, Bending - hands up all the way down, Punching (boxing), Waving - two hands, Waving - one hand (right), Clapping hands, and Throwing a ball.*

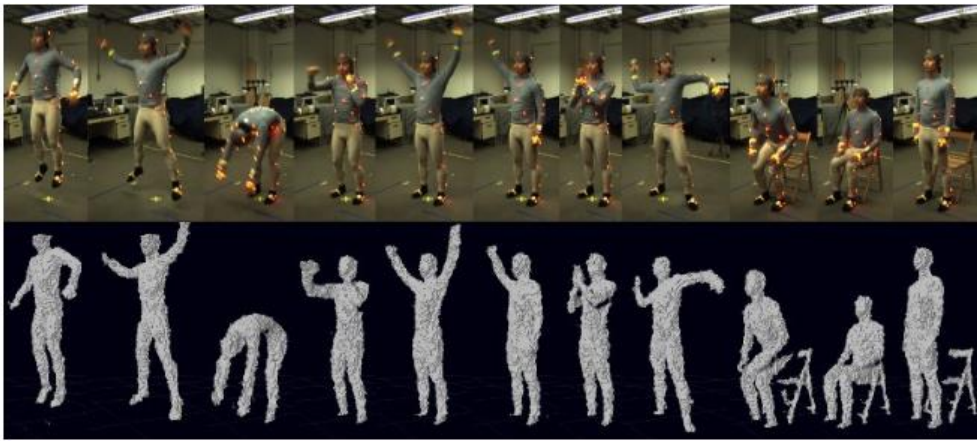


Figure 6
Snapshots from all the actions available in the dataset are displayed together with the corresponding point clouds obtained from the Kinect depth data.

5. Approach

The method I intended to use in my approach is defining the progress into several stages. The first is the preliminary data processing stage. The second stage is to train different models using a single kind of input to indicate the project's baselines. And for the last stage, which is merging the models built in the second stage to construct the fusion model. Using various metrics, the performance of different models will be evaluated and compared for the finding of this project.

5.1 Data Preprocessing

MediaPipe, a framework that provides a machine learning solution for computer vision, will extract skeleton data from video[12]. In [13], an AI Gym Tracker project was implemented using MediaPipe to find landmarks of human joints. Therefore, by connecting these landmarks, we can obtain the skeleton data from a video. The next step is to construct the skeleton images from these landmarks, and hence the background of the original images is removed.

Figure 7
Sample photo after removing background of data using MediaPipe

By finding the pose landmarks of the person, we can output x and y-coordinate data of landmarks of a person. Landmarks are referred to the significant joint points of the human body. The landmark model in MediaPipe Pose predicts the location of 33 poses landmarks from the image data that feed into it. For this research, I only extracted 22 landmarks as they already had sufficient data for motion recognition analysis. The reason is that we do not need any facial data in this work, and these data could lead to unnecessary noise in the coordinates data, therefore, only landmarks

from number 11 (left shoulder) to 32 (right foot) are extracted using MediaPipe (See Figure 8).

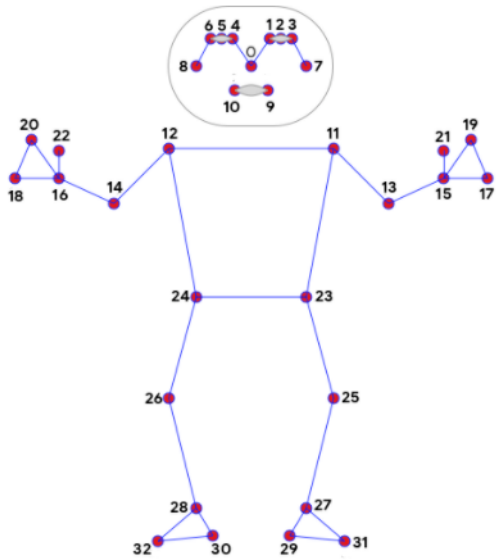


Figure 8

33 pose landmarks

By writing the x, and y-coordinates data of the 22 landmarks of the images to a CSV file, each row of the CSV file will contain the label and the 44 coordinates of the data.

```
0.733954668045044 0.26878654956817627 0.6768304109573364 0.26021888852119446 0.7631619572639465
0.3717852830886841 0.6905635595321655 0.3567628860473633 0.7365032434463501 0.4726296365261078
0.6528353691101074 0.42373716831207275 0.7251421213150024 0.4944092929363251 0.6270288825035095
0.4413214325904846 0.7040931582450867 0.4923876225948334 0.6285145282745361 0.4417997896671295
0.7051621079444885 0.48498713970184326 0.6357876658439636 0.4357246458530426 0.6596217155456543
0.452903151512146 0.6325671076774597 0.4346509277820587 0.4000207185745239 0.4958391487598419
0.42420390248298645 0.49218615889549255 0.4002942442893982 0.6279653310775757 0.40413686633110046
0.618043065071106 0.43201860785484314 0.6496304273605347 0.4388052523136139 0.6405354142189026
0.29712095856666565 0.672799289226532 0.31134453415870667 0.6609736680984497
```

Figure 9

Sample of coordinates data

While the acceleration data consisted of six three-axis wireless accelerometers measuring the wrist, ankle, and hip movements, the accelerometer signal data were obtained from the three-axis accelerometer sensor data from six places on the body, i.e., both wrists, both hips, and both feet, from the 12 subjects.

```
0.900391 0.215820 0.092773 1309539264.049021
0.922852 0.220703 0.080078 1309539264.071012
0.918945 0.154297 0.077148 1309539264.095024
0.885742 0.174805 0.052734 1309539264.133021
0.877930 0.153320 0.014648 1309539264.159013
0.819336 0.127930 -0.027344 1309539264.205023
0.751953 0.076172 -0.083984 1309539264.229001
0.659180 0.087891 -0.085938 1309539264.252024
0.565430 0.029297 -0.131836 1309539264.296022
0.472656 0.004883 -0.126953 1309539264.319025
```

Figure 10
Samples of accelerometer data

The first three data represent the three axes accelerometer data in the X, Y, and Z channels, while the last float number of each data (such as 1309539264.049021) represents the timestamp of the data. The timestamp will be removed from the data as it is useless after constructing the segmentation of the accelerometer data.

After extracting these three kinds of data from the dataset, preprocessing is needed to be done for each of the data. For the skeleton images, normalization by converting the image to a size of 100x100 and framing will be performed to stack up the images to eliminate the body difference between individuals like height and body shape and represent the flow of time (around 1 second for each stack of images) to improve the performance of the model.

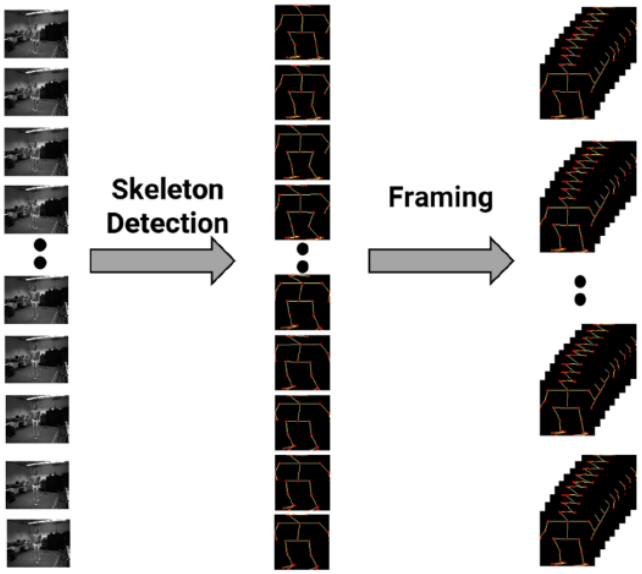


Figure 11
Preprocess of skeleton data

For the coordinates data, the same amount of coordinate data as the stacked skeleton images data will be flattened to one input, around 1 second, when fed into the model.

For the accelerometer data, by using a 1s non-overlapping window, we can conduct signal processing and segmentation to divide the accelerometer data into data with 1 second time length. Therefore, the three kinds of input can be matched as representing the same time length for each input when feeding into the fusion model.

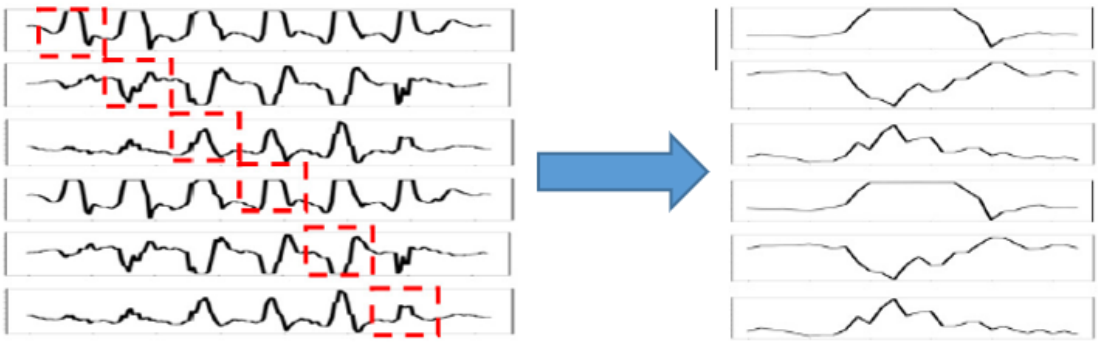


Figure 12
Preprocess of accelerometer data

5.2 Training models by using single input

The second stage is to train the human motion recognizer using a single-model approach. To put it precisely, variants of detectors will be made by different models accepting the different single types of input, such as using only skeleton images for training the CNN model, using accelerometer data to train the CBAM model, etc. The result will be recorded for further analysis and comparison with the performance of the fusion model that will be completed in the last stage. These results from different single models will be used as the baseline to compare in work. For evaluation metrics, data loss and accuracy will be used as the main comparator for the performance of various models.

To evaluate the performance of the models which are taking single input, ResNet10, ResNet34, and ResNet50 have been chosen for training for image input. The difference between these three models is the depth of layers as well as the number of CNN blocks inside the model. However, due to the preprocessing of the image data, like removing the background and converting to a skeleton image, ResNet34 and ResNet50 have no significant improvement in performance by comparing to that

ResNet10. Hence. ResNet10 will be chosen for constructing the model.

Moreover, for the sequence data, which is the coordinates data and the accelerometer data, LSTM, Bi-LSTM, BERT, and Attention model have been chosen in consideration for handling these data. The difference between LSTM and Bi-LSTM is that the input of the Bi-LSTM flows in both directions and hence can utilize information from both sides. Hence Bi-LSTM will be used instead of LSTM only for handling the features of coordinates data as these data are a kind of time-series data containing past and future information.

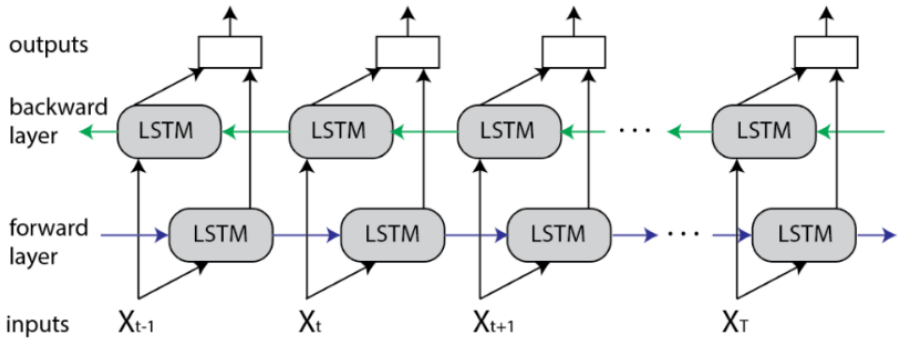


Figure 13
The architecture of Bi-LSTM model

Furthermore, the Attention model will combine the convolution block with 1D convolutional layers and the Attention module to process the 1D accelerometer input. This model is proposed in [14], a lightweight and general module for training. However, for the BERT model, after implementing the model, the size of the model has become too large, and it is not very feasible to use the BERT pre-trained model in this project. Therefore, CBAM will be implemented for handling the accelerometer data.

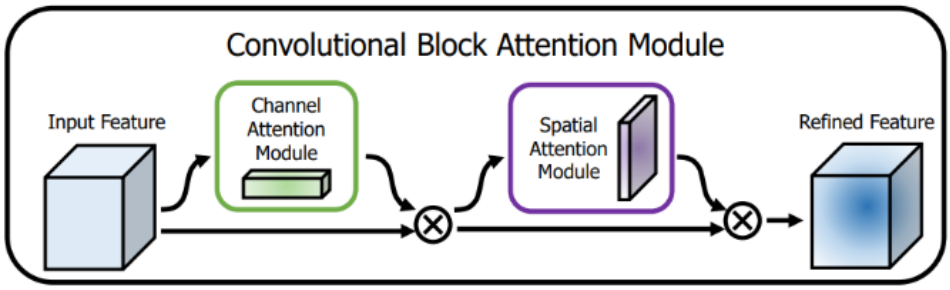


Figure 14
The architecture of CBAM model

5.3 Constructing fusion models and evaluation

The fusion model will be completed for the last stage by combining several models as the feature extraction network. For the fusion network, the features of these models will be concatenated and output by the fully connected layers and softmax function. After getting the result of the fusion model, the evaluation will be done by comparing the accuracy and loss of training as well as the validation set and computing the confusion matrix. In this research, two fusion models will be made to evaluate the result. The first one is to combine skeleton images and the accelerometer data. The second one is to combine all three kinds of data and extract features from them. After the feature extraction stage, 512 features should be extracted from the image data, 256 features should be extracted from the accelerometer data, and 12800 features should be extracted from the coordinate data. The next step will be the fusion network which concatenates the three kinds of features, and then the next layer will be a fully-connect layer with 512 nodes and then the output layer with eight classes of action (See Figure 15).

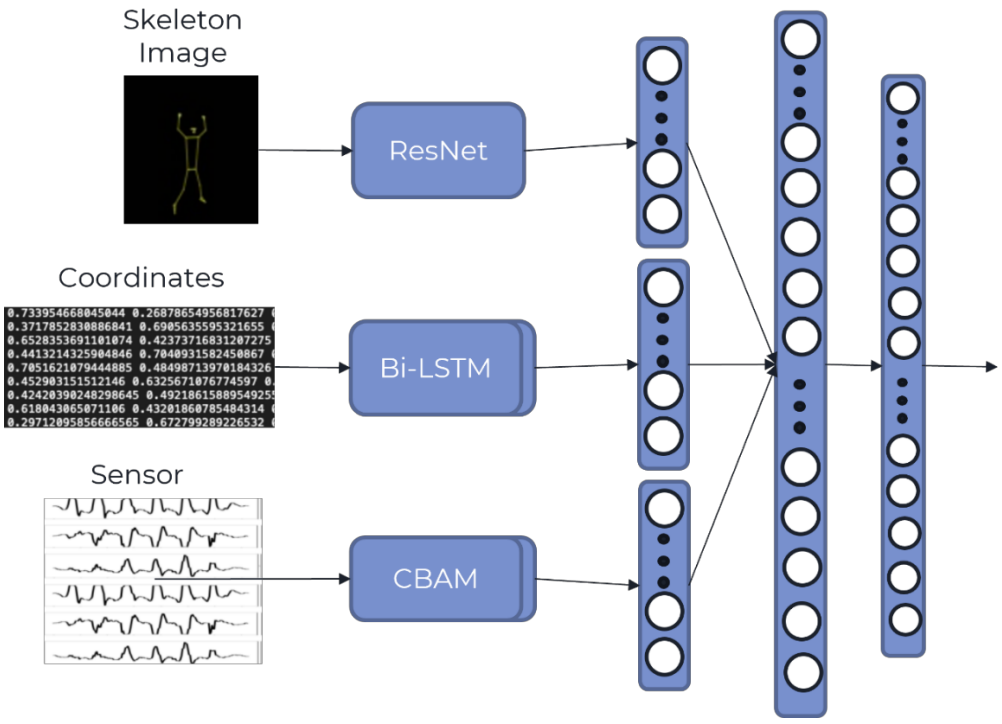


Figure 15

The proposed architecture of the fusion model

6. Experiment

I have built different single models and fusion models to compare with the performance of the proposed model. For instance, using image data to train Resnet10, using accelerometer data to train CBAM, using coordinates data to train the Bi-LSTM model, combining accelerometer data and image data to train Resnet10+CBAM, as well as the proposed CBAM+Bi-LSTM+Resnet10 fusion model. For all the models mentioned above, the optimizer implemented is the Adam optimizer with a learning rate of 0.01. Moreover, the training and validation epoch of the models is set to 100 epochs for better analysis of the result.

6.1 Single models result

For the accuracy of using skeleton images to train the ResNet10 model, basically, this is an image classification task. It achieves an accuracy of up to 76.02%. Whereas the accuracy of using accelerometer data to train the CBAM model is slightly better than that of using image data, which is 80.47%.

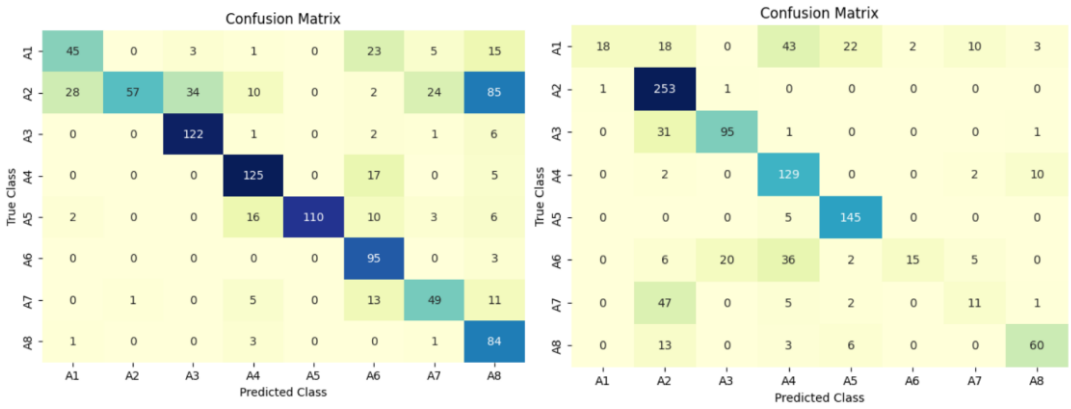
However, using solely coordinates data only for training the Bi-LSTM model has yet to perform well. Some analysis has been done to investigate this issue, the loss of training and validating did not converge throughout the training progress and the model's accuracy remained at around 20%.

After investigating the training log and the data input to the model, the reasons behind this could be different coordinates data distribution for each of the recordings of the same motion class. The variance of these coordinates data could be enormous even within the same class of motion.

The second reason is that the result can be affected by the MediaPipe pose model that detects the coordinates of the landmarks of the person's body. Without normalisation of the coordinate data, the result could vary badly and lead to an ineffective prediction of human motion. As a result, the coordinate data should be used along with the original skeleton images as the input to the model. The above will be discussed in the later section.

By investigating the confusion matrix of the two models, the result shows that the prediction for the classes like A6, A7 and A8 is not performing as well as other classes, especially for the accelerometer-only model. These classes are A06(waving-

one hand (right)), A07 (clapping hands), and A08 (throw balls), which are the classes for the motion of using hands and arms. By analysis, these data could be very similar in terms of the accelerometer data. Moreover, without the movement of the legs, it is difficult for the model to differentiate the class from another arm's motion. For the image classifier, the performance for the class involving jumping is slightly worse than the performance of predicting other classes. The model could get confused when the person is jumping in the air.

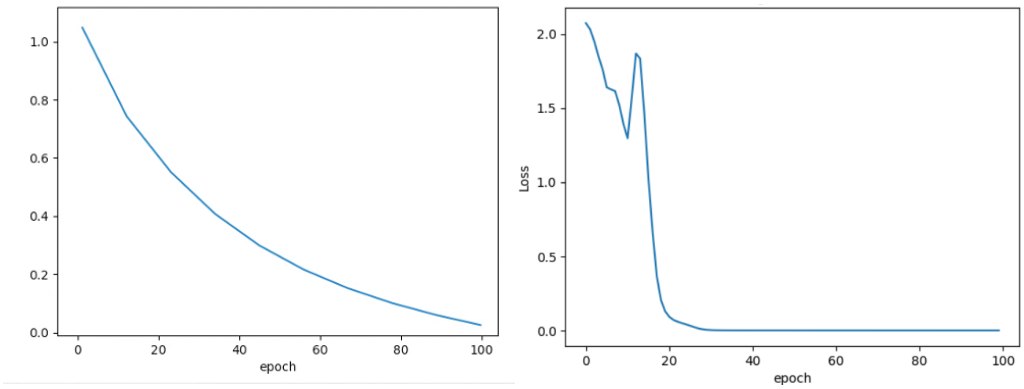


(a) **(b)**

Figure 15

(a) Confusion Matrix of image-only model,

(b) Confusion Matrix of accelerometer-only model



(a) **(b)**

Figure 16

(a) Validation loss of image-only model,

(b) Validation loss of accelerometer-only model

6.2 Fusion models result

For better analysis, apart from using three different kinds of input and models to construct the fusion model, another fusion model that used the skeleton image data and the accelerometer data has been built in this project. The accuracy of the image + accelerometer has achieved a higher accuracy than the other single model's performance which is 83.63%. For the proposed fusion model that take the three kinds of input, the accuracy is the best among all the constructed model by achieving a slightly higher accuracy of 86.59%.

The fusion model uses skeleton images and coordinates together for the fusion model because various studies and research has been done and revealed the convincing result of improvement using these two types of data. This project would not focus on investigating the performance of using these two related data.

By investigating the confusion matrix of the two fusion models, the performance of the two models on the class A6, A7, and A8 has improved significantly after combining two or three kinds of input. The overall performance among all the classes is quite good and satisfied.

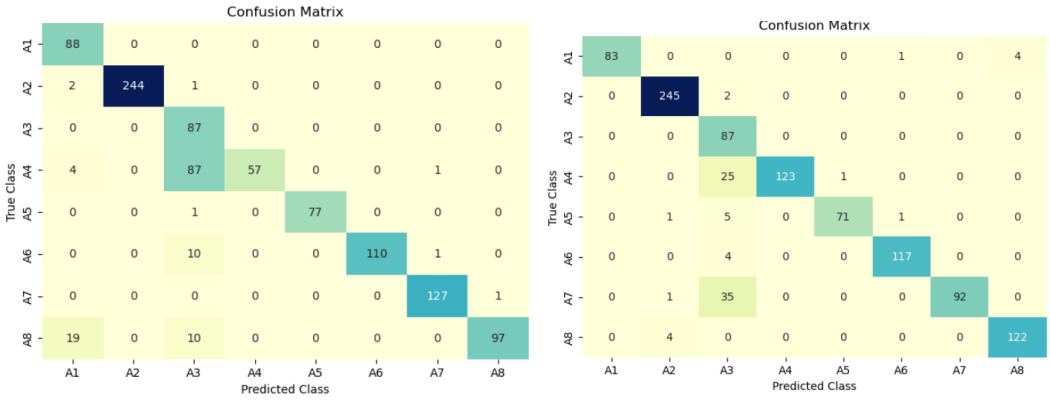


Figure 17

(a) Confusion Matrix of image + accelerometer model,

(b) Confusion Matrix of image + coordinate+ accelerometer model

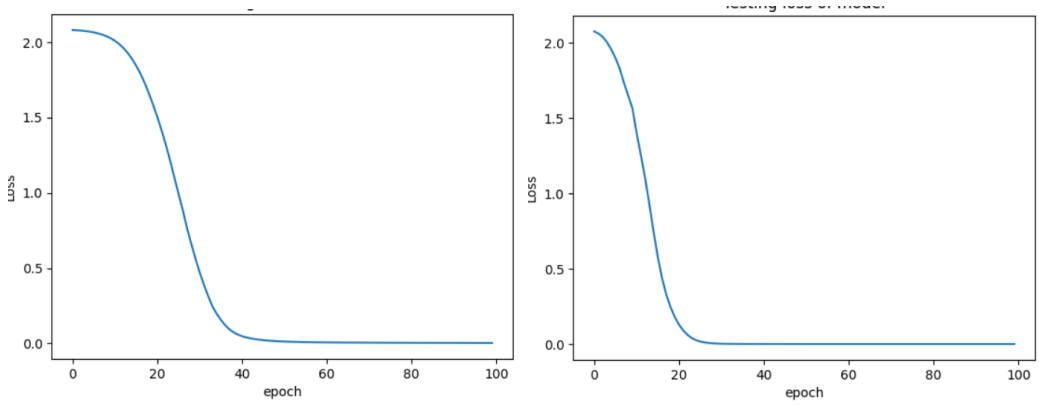


Figure 18

(a) Validation loss of image + accelerometer model,

(b) Validation loss of image + coordinate+ accelerometer model

6.3 Evaluation of different approaches to solve the task

The result has shown that the performance of the fusion model is obviously better than using either one kind of input within the three kinds of input. By inputting heterogeneous data into the model, the fluctuations in the recognition accuracy rate have been reduced compared to other methods. For example, it is challenging to classify arm motion by just using the three axes accelerometer data as they have very similar movement sequences during the action. Nonetheless, by combining the image data, the model can extract visual features to identify the motion instead of the time series of sensor data. Therefore, the prediction of these arm motion classes has been improved by using the fusion models.

By comparing the training time of each model, it is confirmed that we can obtain a satisfactory result in motion recognition by using just the image classifier with a lower computational cost. The reason is that the input data is preprocessed to simple skeleton images by removing the background of the data and reducing the individual differences in the human body. Therefore, with a light-weighted CNN model ResNet10, the training time is only 40 seconds for one epoch. The variance of the performance of the image classifier is smaller than that of the model using accelerometer data. As a result, an image classifier is enough for simple human motion recognition tasks. The training time of the fusion models is 100 seconds and 150 seconds for one epoch for the image + accelerometer model and the image + coordinates + accelerometer model, respectively. Intuitively, the time cost of training these fusion models is higher than that of the single model. However, the performance will be better for complicated classifying tasks.

The limitations of this project are as follows, due to time and storage constraints, only five recordings of 5 repetitions for each motion have been used and downloaded from the MHAD datasets. Therefore, the size of the datasets for training the model is not very large, which could lead to degradation in training the models. Moreover, due to the limit of the dataset, only eight classes of motion are used for this recognition task. Therefore, the usefulness of this project is limited to detecting these motions. In fact, it is a significant hurdle for me to collect my own data, especially for the accelerometer data, which consists of 6 shimmers of three axes data. Although we could obtain accelerometer data using our phones, it is nearly impossible to have six devices tied to my body to collect the data. Apart from data collection, the models chosen for this project are only some commonly used models like ResNet, Bi-LSTM, and the Attention model. Intuitively, by using other advanced top-tier models such as EfficientNet, and BERT instead, the performance of the fusion models can be further improved.

7. Conclusion

In this project, a fusion model that takes not only RGB images but also the coordinates vectors and sensor data has been proposed to solve the human motion recognition task. Using the MHAD dataset, preprocessing of the data has been performed to obtain the skeleton images and coordinates of body landmarks using MediaPipe. By aligning these data as 1-second input, the fusion model's accuracy is higher than that of any other model that takes only single input. For future implementation, I will focus on fine-tuning the models by trying different models with more robust image and sequence analysis performance. Moreover, own data of doing different actions is expected to implement as the dataset for evaluating the model in real-case scenarios. I hope that eventually, a real-time application of human motion recognition can be achieved, in which this technology will be beneficial in various industrial applications.

8. Reference

[1] A. Tariq, L. A. Celi, J. M. Newsome, S. Purkayastha, N. K. Bhatia, H. Trivedi, J. W. Gichoya, and I. Banerjee, "Patient-specific COVID-19 resource utilization prediction using fusion AI model," *npj Digital Medicine*, vol. 4, no. 1, 2021.

[2] A. Jaimes, "Multi-modal fusion AI for real-time event detection," *Real-Time Event and Risk Detection*. [Online]. Available: https://www.dataminr.com/blog/multi-modal-fusion-ai-for-real-time-event-detection. [Accessed: 18-Mar-2023].

[3] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," *Lecture Notes in Computer Science*, pp. 216–223, 2012.

[4] S. Wan, Z. Gu, and Q. Ni, "Cognitive computing and wireless communications on the edge for Healthcare Service Robots," *Computer Communications*, vol. 149, pp. 99–106, 2020.

[5] P. Khaire, P. Kumar, and J. Imran, "Combining CNN streams of RGB-D and skeletal data for human activity recognition," *Pattern Recognition Letters*, vol. 115, pp. 107–116, 2018.

[6] J. Kang, J. Shin, J. Shin, D. Lee, and A. Choi, "Robust human activity recognition by integrating image and accelerometer sensor data using Deep Fusion Network," *Sensors*, vol. 22, no. 1, p. 174, 2021.

[7] K. Kim, A. Jalal, and M. Mahmood, "Vision-based human activity recognition system using depth silhouettes: A smart home system for monitoring the residents," *Journal of Electrical Engineering & Technology*, vol. 14, no. 6, pp. 2567–2573, 2019.

[8] F. Ordóñez and D. Roggen, "Deep Convolutional and LSTM Recurrent Neural Networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[9] H. Wei, R. Jafari, and N. Kehtarnavaz, "Fusion of video and inertial sensing for Deep Learning–based human action recognition," *Sensors*, vol. 19, no. 17, p. 3680, 2019.

[10] S. Kiran, M. Attique Khan, M. Younus Javed, M. Alhaisoni, U. Tariq, Y. Nam, R. Damaševičius, and M. Sharif, "Multi-layered deep learning features fusion for human action recognition," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 4061–4075, 2021.

[11] *Berkeley Mhad*. Berkeley MHAD | Teleimmersion Lab. (n.d.). Retrieved April 19, 2023, from https://tele-immersion.citris-uc.org/berkeley_mhad

[12] "Mediapipe," *mediapipe*. [Online]. Available: https://google.github.io/mediapipe/. [Accessed: 18-Mar-2023].

[13] "AI pose estimation with python and mediapipe | plus ai gym tracker project," *YouTube*, 14-Apr-2021. [Online]. Available: https://www.youtube.com/watch?v=06TE_U21FK4. [Accessed: 18-Mar-2023].

[14] Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018, July 18). *CBAM: Convolutional Block Attention Module*. arXiv.org. Retrieved April 20, 2023, from https://arxiv.org/abs/1807.06521