ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse711032021.pdf
https://doi.org/10.30534/ijatcse/2021/711032021

A systematic Approach to Human Motion Recognition using Deep Learning

Shaashwat Agrawal[1], Anish Jamedar[2], Mangamudi Yashwanth[3], Dr. Swarnalatha P[4]
[1]School of Computer Science and Engineering, VIT, Vellore, 632014, India, shaashwat.agrawal2018@vitstudent.ac.in
[2]School of Computer Science and Engineering, VIT, Vellore, 632014, India, jamedar.anish2019@vitstudent.ac.in
[3]School of Computer Science and Engineering, VIT, Vellore, 632014, India, mangamudi.yashwanth2019 @vitstudent.ac.in
[3]**Department of Information Security, VIT, Vellore, 632014, India, swarnalatha.vit.ac.in**

ABSTRACT

The application of Human Motion Analysis (HMA) under Computer Vision (CV) is an emerging field which entails various applications such as gait analysis, behavioural cloning and animation of motion, intent detection, etc. For such motion analysis various open source datasets have been created that help analyze motion behaviour. Motion Capture (mocap) files have been used extensively to store motion data and analyze them. Although the weightage of these applications can be huge in modern technology, not much work on human motion recognition has been done using mocap datasets. In this paper, we propose a systematic approach to human motion recognition using software engineering, data analysis and deep learning algorithms. A Deep Learning (DL) model using Gated Recurrent Network (GRU) for the classification of human motion. CMU mocap dataset is used for analyzing motion data and modelling the DL framework. The trained algorithm is tested using accuracy and Mean Absolute Error (MAE) and a user live feed as performance metrics. A 90.1% validation accuracy is obtained on final evaluation.

Key words : human motion recognition, motion capture, deep learning algorithm, software engineering, etc.

1.INTRODUCTION

Motion analysis is a classification procedure under computer vision, image processing and fast photography applications to recognize development. Motion analysis is utilized in computer vision, picture preparation, high velocity photography and machine vision that reviews techniques and applications in which at least two sequential pictures from a picture arrangement are handled to create data dependent on the clear movement in the images. In certain applications, the camera is fixed comparative with the scene and articles are moving around in the scene, in certain applications the scene is pretty much fixed and the camera is moving, and sometimes both the camera and the scene are moving. Uses of motion analysis can be found in rather different zones, like reconnaissance, medication, entertainment world, car accident safety, ballistic gun studies, biological science, flame propagation, and route of self-governing vehicles to give some example models. Human motion recognition is also such an application of Motion analysis. Human motion analysis [1] [2] through a frame of postures which are exhibited by the human body are called Human motion. In the spaces of medication, sports, video observation, physical therapy [3],and kinesiology [4], HMA has become an analytical and symptomatic tool. HMA is

most regularly utilized for video reconnaissance, explicitly programmed movement checking for security purposes. Most endeavors in this space depend on state-space draws near, in which successions of static stances are measurably dissected and contrasted with demonstrated developments. Format coordinating is an elective technique whereby static shape designs are contrasted with previous prototypes. Human motion can't be plotted or deciphered utilizing 3d graphs. So for this we use BVH files which can store the human motion by consuming very less data compared to the video files.

BVH file is used in motion capturing data format. The BVH files are principally utilized as a standard portrayal of developments in the liveliness of humanoid structures. The CMU (Carnegie Mellon University) dataset [5] contains 2500 BVH files representing different types of activities. Some extraordinary activities, for example, swordplay and cartwheel are additionally remembered for this dataset. Since BVH files contain human motions [6], this can be utilized in training the AI model using DL algorithms. Deep learning is a sub- domain of Artificial intelligence and Machine Learning that follows the operations of the human cerebrum for handling the datasets and perfect decision making. Some examples of DL are Remote helpers, vision for driverless vehicles [7], illegal tax avoidance, face acknowledgment and some more. In our project we train the AI model using both GRU and LSTM models. Both the models take the input from the sequential data and learn from the sequential data. It is a time-series data of about 110 frames per second and fits

perfectly to GRU model. The GRU is the modified version of Recurrent Neural networks (RNN) and is quite like a LSTM in terms of architecture as shown in Fig 1.

But in conventional RNN, where the organization is prepared by means of back propagation [8] through time, an issue known as detonating and disappearing slopes, are available, causing a detonating angle issue. This is when huge mistake slopes begin gathering, bringing about critical changes to the neural organization model during preparing, which as a result keeps a model from preparing with the accessible information, and makes the prepared model be temperamental. Disappearing slopes is the point at which the inclinations of misfortune capacities become excessively little (approaches zero), at that point the organization turns out to be progressively difficult to prepare, as the loads and predispositions of the underlying layers are not refreshed adequately with the instructional courses.

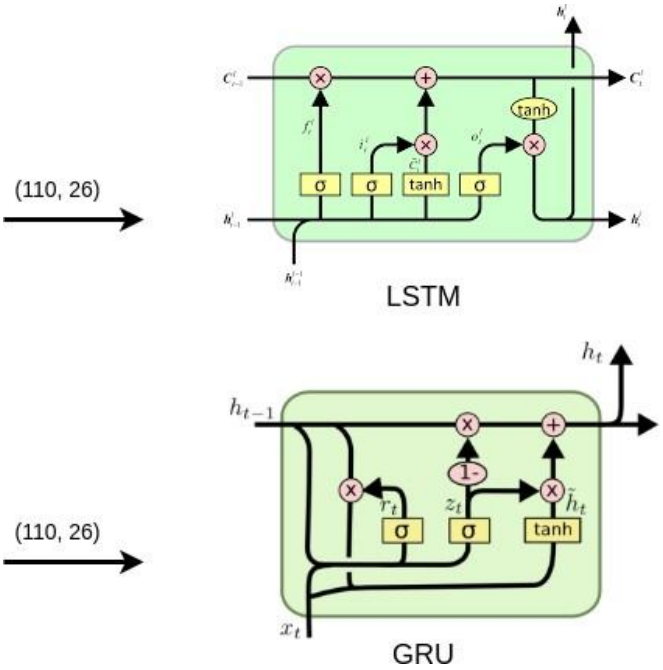


Fig. 1: Cell Architectures of LSTM and GRU

In this paper, we propose a systematic approach to the analysis and understanding of human motion. A software engineering approach is utilized for the development of human motion applications. First, an User Interface (UI) is created, followed by analysis of various motions (i.e running, walking, jumping) using data visualization and finally an AI model is trained based on preprocessed data to recognize human motion. The main contributions of this paper are:

• The work utilizes BVH files instead of images for application of motion recognition. Through this the efficiency of the AI model is boosted as redundant data and noise present in an image do not interfere with the training of the model.

• Implementation of an UI for better analysis of the dataset, the motions in it and the AI model. Microscopic details like frame rate and aspect ratio of each motion can be analyzed using proper visualization which otherwise is difficult to detect.

• An experimental setup is created using the CMU dataset on which GRU based Recurrent models are created. Various hyper-parameters are taken to fine tune the model. The performance of the model is not only measured though error functions but also through user live-feed.

Section II is used to perform an extensive review on the state-of-the-art algorithms used in the domain of DL for motion analysis. In Section III a detailed methodology of our paper is discussed followed by the results shown in Section IV. Lastly, the obtained analysis is summarized in Section V.

2. LITERATURE SURVEY

In this section, a detailed review of the domains of human motion recognition and DL is conducted. The limitations of the current research is provided for DL based motion analysis techniques and algorithms.

2.1 Human Motion Analysis

Nowadays research on Human motion recognition[9] has been increasing rapidly. For the classification of human motions, various techniques are used. Regular strategies are Dynamic Time Warping (DTW) [10], Hidden Markov Models (HMM), Decision Trees and Support Vector Machine (SVM) [11]. In some works we can find the recognition of human motion [12] using joint action in conjunction with SVM. SVM with a polynomial kernel is also suitable for human motion recognition. Motion recognition with RGB -D [13] data has also been in demand. But we can also find some limitations in using these techniques like The detriment of utilizing DTW is the hefty computational weight needed to figure out the ideal time arrangement way. A few elective strategies have been proposed for lessening the calculation of DTW calculations. On the off chance that the time stretch is excessively short, Markov models are improper on the grounds that the individual removals are not arbitrary, but instead are deterministically related on schedule. This model [14] proposes that Markov models are by and large unseemly throughout adequately brief time frame stretches. SVM algorithm is not applicable for huge data sets. SVM doesn't perform very well when the dataset [15] has more commotion for example target classes are covering. In situations where the quantity of highlights for every information point surpasses the quantity of preparing information tests, the SVM [16] will fail to meet expectations. RGB-D sensors have critical disadvantages including restricted estimation ranges (e.g., inside 3 m) and mistakes top to bottom estimation increment with distance from the sensor regarding 3D thick planning

2.2 Deep Learning Algorithms

These days, with the quick advancement of machine learning procedures, bunches of DL strategies have been generally utilized in the applications of human motion classification [17][26]. A technique for human activity classification utilizing deep convolutional GAN (DCGAN) and deep CNN (DCNN) [18] is used to improve results under the state of small training sample number. And also we can find the combination of both CNN and LSTM methods [19] . Although there are some drawbacks in these methods like Destiny Estimation in GAN, we actually can't foresee the precision of the thickness of the assessed model and express that this picture is sufficiently denser to push ahead with. These metrics of the information produced are as yet choosing physically [20]. GANs are the exquisite instrument of Data Generation however because of Unstable Training and unaided learning technique it gets more enthusiastically to prepare and create yield. Most important is making any changes after developing the most complicated task in DCGAN. In the same way, Models like CNN are essentially slower because of activities like maxpool. In the event that the CNN [21] has more layers, the preparation interaction takes a great deal of time if the PC doesn't comprise a decent GPU. It also requires a huge Dataset to process and train the model. LSTM [22] has been intended to beat these issues that emerge in RNN structures. LSTM and RNNs [23] are basically the same, with the distinction being that secret layers in LSTMs [24] contain memory blocks with cells rather than non-direct units, which can store data over long time frames. At the end of the day, customary RNN [25] cells have a solitary interior layer that follows up on the present status (h.) and info (x), while a LSTM cell contains three such layers.

From the survey we have identified that the methodology implemented in the research of human motion recognition needs to be modified to better analyze the data and the resultant model.

3. METHODOLOGY

In this section, a detailed explanation of our systematic approach is provided. Fig 2 shows the architecture of the proposed algorithm. The development of the application is done using a software engineering approach. The initial step begins with the choosing of an appropriate software life-cycle model. We choose the incremental model since it complements a research based project with many wide functionalities. It gives us the benefit of developing the software in small mod-ules and thus making the implementation easier. The architecture also represents software design principles.

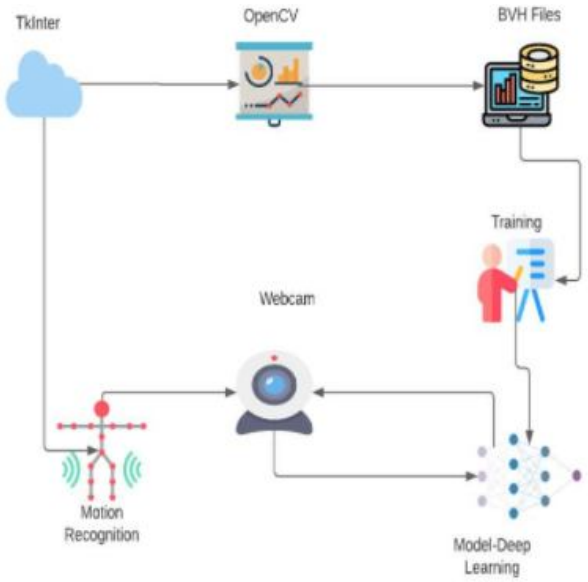


Fig. 2: Architecture for systematic approach to human motion recognition

Algorithm 1 describes the flow of each sub-module in the application. The process starts with the implementation of the UI in subsection III-A, followed by analysis of data using data visualization in subsection III-B and finally AI model specification in subsection III-C.

Algorithm 1 Systematic Approach to Human Motion Recognition

M ← motion name

E ← Number of epochs

```
1: function USER INTERFACE(η)
2:     OPTA <- {Data Visualization, Motion Recognition}
3:     if( OPTA == 1):
4:         DATA VISUALIZATION(motion)
5:     else:
6:         LIVE FEED
7: function DATA VISUALIZATION(M)
8:     M <- Running, Walking, Jumping
9:     Extract motion from CMU dataset
10:    Visualize M
11: function TRAINING
12:    w0 <- Initialization of weights
13:    Normalize Data
14:    Convert data into batches. 16:
15:    for E <- 0 to size(N) do
16:    Train w0
17: function LIVE FEED
18:    Load w0
19:    Access webcam
20:    while true do
21:        Extract frame
22:        frame <- w0(frame)
23:        Display frame
24:    End
```

3.1 User Interface

Tkinter is the most used Graphical User Interface (GUI) library in Python. Python when used with Tkinter provides a quick and simple approach to make GUI applications. Tkinter gives an incredible article-arranged interface to the Tk GUI toolbox. The layered methodology utilized in developing Tkinter GUI gives Tkinter the entirety of the upsides of the TK library. This makes early forms of Tkinter significantly more steady and solid than if it had been reworked without any preparation. This is a continuation of the Python perspective, since the language dominates at rapidly constructing models. Python scripts that utilize Tkinter don't expect changes to be ported from one stage to the next. Tkinter is accessible for any stage that Python is carried out for, to be specific Microsoft Windows, X Windows, and Macintosh. Tkinter is presently remembered for any Python appropriation. And, no supplementary modules are needed to run scripts utilizing Tkinter.The first two lines allow you to create a full window. The window's subtitle is placed on the third line, and it enters its occasion circle on the fourth.

```
from Tkinter import *
root = Tk( )
root.title("Simple application")
root.mainloop( )
```

Using the above library a simple UI is created considering design principles of usability and Consistency. In the home page, there are 2 options: Data Visualization and Motion Recognition. On selecting data visualization options, there are 3 main motion choices that are available, namely running, walking and jumping. The selected motion is then visualized using the OpenCV window. A for loop is run for the number of frames and a skeleton figure is drawn for the given 2D points. On selecting motion recognition, the live feed of the user is turned on and their predicted motion can be viewed.

3.2 Data Visualisation

The digital representation of facts and statistics is known as data visualisation. Data visualisation makes it simple to evaluate and understand trends, outliers, and patterns in data by using visual elements such as charts, graphs, and maps. A chosen file from the pickle files is loaded and stored in the RAM. Its data is corrected and a skeleton is formed given the 2d motion data. The motion of the skeleton is then visualized frame by frame in an OpenCV window. This is positional data, not statistical data. So we Visualise instead of graphs. The positional data can be understood better when we visualise the data.

CMU (Carnegie Mellon University) Motion Capture (mo-cap) dataset is an publicly accessible dataset on motion information. The CMU dataset contains around 2500 BVH (BioVision Hierarchy) files ranging from 200 KB - 5 MB in size. It also contains a CSV file that marks each BVH document with the kind of movement present. We use CMU motion capture dataset to envision the information. Motion (i.e. a person running, walking) is usually stored in the form of video files. But raw motion data can be stored in mocap files like BVH, C3D, FBX etc. BVH has .bvh as a file format used to define a skeleton structure and its connections. It is a mocap file which stores position/turn of each joint per frame. The movements can be viewed directly by importing this file or also by imparting this movement to 3d Characters/Rig.

BVH file part 1 as shown in Fig 3 recognises Hips part as root, which says it has no-parent joint. It is the starting point of the nested structure of parent and child joints. The root part describes the location of hips in three-dimensional space. Below the root part there are joint sections containing information that shows the location of the skeletal joint according to its parent joint. The root section, each of the joint, also provides channels for time-framed sequence of translation or co-ordinates provided in the next part of the BVH file.

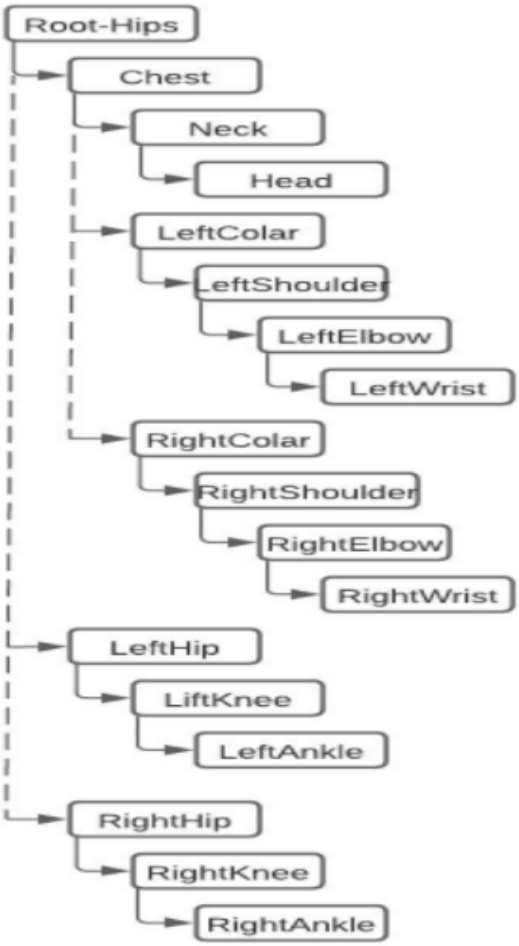


Fig. 3: Sample Hierarchy of a BVH file

BVH file part 2 begins with word motion, and next data specifying, first the no. of frames, next, sample rate per sec after the word frame time, and last, a no. of lines correlating to the no. of frames, each line giving the translation or rotation coordinates to be analyzed according to the channels in the previous part of file. The CMU dataset contains BVH files that can't be directly interpreted by python. These files are then converted to a CSV file using a python library. From the CSV files containing 3D coordinates, 2D motion is extracted, normalized and saved in the form of pickle files.

To visualize the data, positional and size corrections are made and make the data suitable for training models by converting the values into a time series format. A particular motion is chosen and given then a random pickle file is loaded and stored in the RAM. Then, it's data is corrected and a skeleton is formed given the 2D motion data. The motion of the skeleton is then visualised frame by frame in an OpenCV window. The frame size of each window is 1,

$$\text{Shape}(F) \leftarrow (480, 640) \quad (1)$$

Where F is the ith frame. For the formation of the complete motion, each such frame of key-points is stacked one after the other 2:

$$\text{Motion} \leftarrow \{F1, F2, F3, Fi, ..., FN\} \quad (2)$$

As per eq, each such frame is extracted one by one in a loop and visualized at a decent frame rate.

3.3 AI Model Live-feed

The saved pickle files are used to train the AI model. A CSV file containing the name of the file and its respective motion label is imported. From all this data, respective input and output are stored in separate variables. A Neural Network is defined using TensorFlow Library. The Network is Recurrent in nature due to the sequential data.The input and output variables are fed to the model to train it. Around 30 iterations are run. The model parameters are tuned every time to make the output performance better. Using web cam the live feed is predicted and processed by using particular joints in our skeleton as points. The key points obtained are passed through the trained AI model every frame. The prediction is shown in the window with the live webcam feed.

4. RESULTS AND DISCUSSION

In this section, the specification of requirements, data handling, the architecture of Recurrent Neural Network used and the results obtained after tuning the model are discussed. Figures and comparisons are provided for each.

4.1 Requirements Validation

Following a software engineering based approach we validate the application and user requirements. In terms of hardware, the system on which the application is training requires a minimum recommended RAM of 8GB and a GPU(Graphical Processing Unit). In terms of software, the whole project is developed using the Python programming language. Python is supported across all Operating Systems including Windows, Mac, Linux, etc. Under python, certain libraries are used to implement the project. Pandas, NumPy, and bvh-convertor libraries are used to preprocess the motion capture data. OpenCV is used to visualize that data. Finally, to work with and train the AI model, the TensorFlow library is used. In terms of user requirements, SRS is divided into functional non functional requirements. Some of these requirements are

:Functional Requirements

– The main functional requirement is satisfactory performance accuracy by the AI model for prediction of human motion.

– The UI design must be consistent and should not contain any surprises.–The database of motion files must be accessible from the UI.

– A proper analysis of the dataset and motion files must be provided.

Non-Functional Requirements

– Performance Requirements: The product works flawlessly given the hardware and software constraints are met correctly. The other performance requirement includes a properly functioning camera or webcam to take the live feed.

– Security Requirement: The product accesses your webcam feed and extracts frames from it but all the data is restricted to the local system and is not susceptible to outside access.

– Correctness : The AI model is aptly trained and tuned on data which assures correctness of results.–Testing constraints : The product can easily be tested in our own systems. The output shown is very simple and easy to examine.

– Usability : The provided User Interface is very intuitive and easy to use.

– Scalability : The product once completely developed and deployed is easily scalable.

The above mentioned requirements have been validated. The Required functionality constraints are fulfilled and the results are shown. The obtained results are to the satisfaction of the user demands. The Table I shows test cases based on user requirements

4.2 Data Preprocessing

The initial dataset is in the form of BVH files. Since these files cannot directly be interpreted or be used for training an AI model, some preprocessing steps need to be performed. The data in a BVH file is converted and stored in CSV files. The obtained format is very similar to that of image coordinate system. For each key-point in our body, a

3-dimensional vector of coordinate values is obtained as shown in Fig 4.

This pattern is followed for each still frame in the compilation of the whole motion. After conversion, 2D vectors for all frames are extracted in the form of< xi,yi>. Once extracted, the processing required for visualization and training vary. For visualization, positional error of the points are corrected.

by multiplying and adding original points by constants. For training, the 2D data is normalized using eq (3).

$$\text{Normalize}(xi) = xi - \min(x)/\max(x) - \min(x) \quad (3)$$

The normalized values are converted to sequential batches of 100 frames each while the x and y coordinates are squeezed into a single valued vector shown in eq (4).

$$\text{Size}(\text{Xtrain}) \leftarrow (N \ 120 \ 26) \quad (4)$$

OpenCV, and numpy libraries are utilized to conduct data preprocessing effectively. These modules perform efficient processing with fast computation algorithms that utilize multi-processing. Although the GPU of the system may not be of use in this case, the throughput is still high.

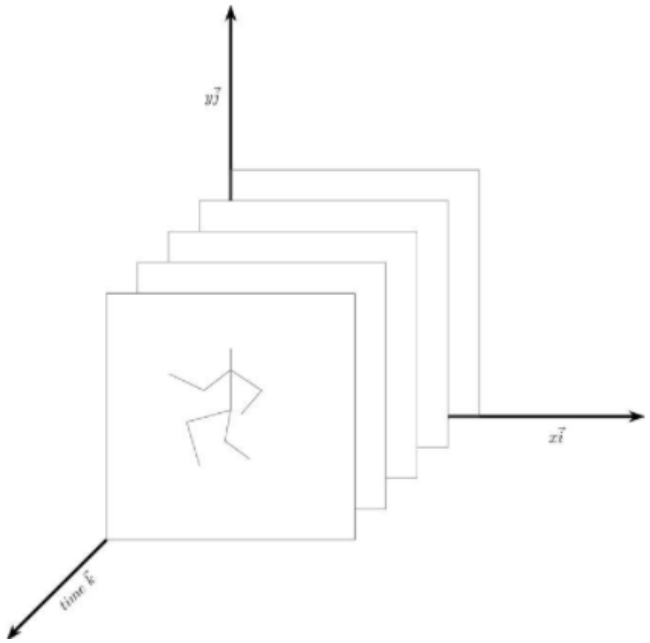


Fig. 4: Motion Visualization in 3D Geometry

4.3 DL Model Architecture

For the recognition of human motion, a RNN deep learning model is constructed and trained. The neural network hasa input shape of (110, 26) which means 110 time-stepsand 26 properties per time-step. Using the Sequential APIprovided by TensorFlow, a 4 layer network is created. To Work with sequential data the initial 2 layers are GRU layerswith 64 neurons each. These are followed by a Dense layer with 32 neurons and ReLU activation function. The final layer is another Dense layer with 3 neurons and Softmax Activation function. Since this is the output layer 3 neurons are included, 1 for each motion (i.e. running, walking and jumping). The loss function used is categorical cross-entropy and the optimizer used is RMSprop.

.Through the RMSpropoptimizer a learning rate (η) of0.0007is set after continuous fine tuning. The model is trained for about 20 epochs with a batch size of 16. The complete architecture of the GRU model can be seen in Fig 5. Similar to the GRU model, the LSTM is also constructed.

Test Case ID	Activity	Inputs	Expected Result	Actual Result	Status [Pass/ Fail]	Comments
TC-01	Click on Data Visualiza-tion, choose	Running	The motion file is extracted from the database and visualized	The motion file is extracted from the database and visualized	Pass	
TC-02	Click on Data Visualization, choose motion	Walking	The motion file is extracted from the database and visualized	The motion file is extracted from the database and visualized	Pass	
TC-03	Click on Data Visualization, choose motion	Jumping	The motion file is extracted from the database and visualized	The motion file is extracted from the database and visualized	Pass	
TC-04	Click	Motion	Live	Live	Pass	Correc

	on Motion Recog nition	from userwe bcam	Webca m feed with respect ive text predict ion: Runnin g or walkin g or jumpin g	Webca m feed with respect ive text predict ion: Runnin g or walkin g or jumpin g		t predict ion from the model
TC-05	Click on Motion Recog nition	Motion from userwe bcam	Live Webca m feed withres pective text predict ion: Runnin g or walkin g or jumpin g	Live Webca m feedwit hres pective text predict ion: Runnin g or walkin g or jumpin g	Fail	Invalid predict ion output from the AI model

TABLE I: Test Cases

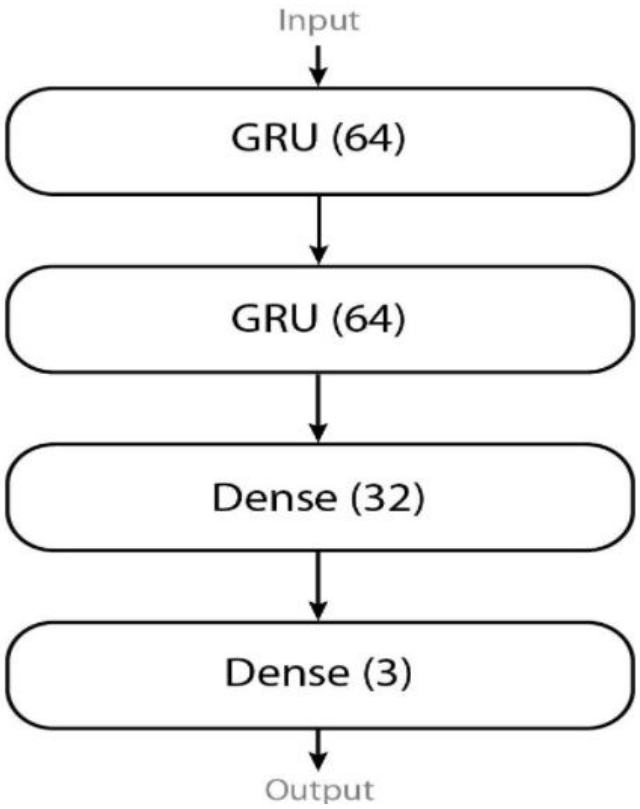


Fig. 5: Architecture for systematic approach to human motion recognition

4.4 Performance Measurement

The proposed DL model is trained and tuned continuously to get apt prediction results. Similar to the GRU layer, LSTMlayers are also trained to test and compare learning capabili-ties. The LSTM training does not obtain much of a result. The Training is not sustainable and consistent. Each new training of 30 epochs produces a varied output. Hence, the output of the LSTM model is unpredictable and not recommended for live prediction. The training graph can be seen in Fig 6

One the other hand, the obtained learning graph from theGRU training is an optimum one. With a test dataset of about0.1% the model performs adequately. The superiority of GRUover LSTM can also be seen through this comparison. The Output graph of training can be seen in Fig 7.From the output graph we can observe that, the validation accuracy in the first few epochs is 0.

This is because themodels learn slowly from the data. It is also observed that the validation accuracy and loss values fluctuate many times during the training. Through this it can be inferred that there are many local minima and maxima caused by the data at hand. The fact that running and walking motions, or even running or jumping motions are very similar by postures, the model may be facing difficulty distinguishing one from the other.

To overcome these circumstances the model is trained with various hyper-parameters and validation accuracy, loss of each are observed. Using this the best model is selected. The table II shows the various values and their output:On microscopic analysis of the Table II some patterns can be observed with the given model hyper-parameters. The Training of the GRU model is not stable as there fluctuations inaccuracy and loss values even with the incremental increase in number of epochs. This shows the ambiguity in the data present. It can also be observed that 7e−3 is the optimal learning rate for training with at least 30 epochs. Although the obtained accuracy values are high, the loss values are also relatively high.

Fig. 6: Output Performance of LSTM model

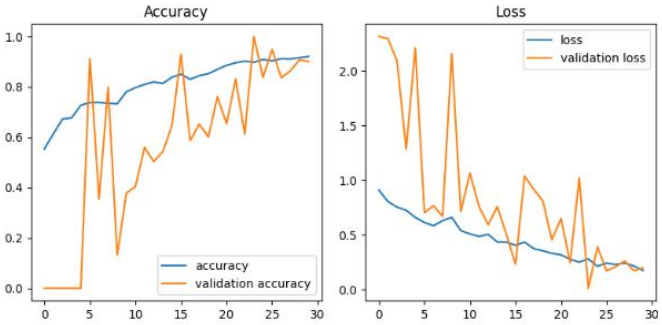
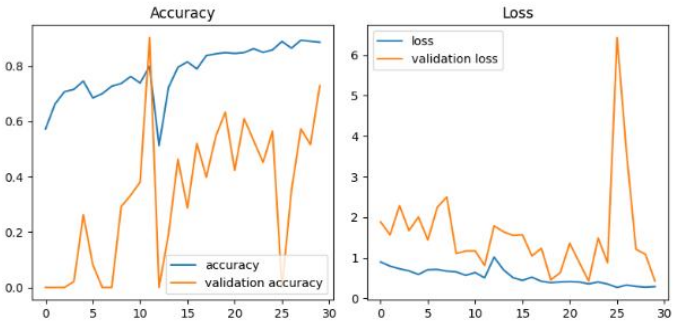


Fig. 7: Output Performance of GRU model



5. CONCLUSION

In this paper, we have proposed a systematic approach to the analysis of human motion. This methodology provides a deeper understanding of the data in question by giving a visual perspective. Some minute properties like motion frame rate of difference are detected through this method. The organized system using software engineering helps us plan out each step of analysis and testing. The deep learning trained model is able to provide significant results, with about 90.33% accuracy. Many optimization algorithms are tried, and multiple layers of tuning are performed on the model. Although the data is slightly ambiguous with many similarities between different motions, the model is able to aptly distinguish the different activities.

Although this implementation provides a good execution technique, there is some future work that can be done to improve the efficiency of the whole system. The prediction is performed using a partial portion of the dataset with only 3 output classes. The whole data can be used with multi-task learning as the final goal. The interface created may be user-friendly but is very simplistic. To provide a better understand- ing of the system, the UI must include more components in it. Finally, the AI model used is very straightforward. State-of- the-art algorithms, architecture and even optimizations should be included in the training and tuning process.

Number of Epochs	learning rate	GRU Model	
		Val. Accuracy	Val. Loss
20	0.0005	0.5933	0.6655
	0.0007	0.6644	0.5897
	0.0009	0.8711	0.4225
25	0.0005	0.8622	0.2925
	0.0007	0.7144	0.6166
	0.0009	0.7110	1.0261
30	0.0005	0.8422	0.4917
	0.0007	0.9033	0.2077
	0.0009	0.8889	0.2254

TABLE II : Performance Evaluation of Proposed algorithm with various hyper-parameters

ACKNOWLEDGEMENT

This paper and the research behind it would not have been possible without the exceptional support of my mentor, Dr Swarnalatha P. She has been supportive throughout the writing of the paper. The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

REFERENCES

[1] Mohammad Almasi. Human movement analysis from the egocentric camera view. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), pages 1026–1031. IEEE, 2020.

[2] Yaxu Xue, Zhaojie Ju, Kui Xiang, Jing Chen, and Honghai Liu. **Multimodal human hand motion sensing and analysis—a review.** IEEE Transactions on Cognitive and Developmental Systems, 11(2):162–175, 2018.

[3] Abdullah S Alharthi, Syed U Yunas, and Krikor B Ozanyan. Deep learning for monitoring of human gait: A **review.** IEEE Sensors Journal, 19(21):9575–9591, 2019.

[4] Ioannis Rallis, Apostolos Langis, Ioannis Georgoulas, Athanasios Voulodimos, Nikolaos Doulamis, and Anastasios Doulamis. **An embodied learning game using kinect** and labanotation for analysis and visualization of dance **kinesiology.** In 2018 10th international conference on virtual worlds and games for serious applications (VSGames), pages 1–8. IEEE, 2018.

[5] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F Troje. Movi: A large multipurpose motion **and video dataset.** arXiv preprint arXiv:2003.01888, 2020.

[6] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture **using a cnn coupled with a geometric prior.** IEEE transactions on pattern analysis and machine intelligence, 41(4):901–914, 2018.

[7] Qiping Chen, Yinfei Xie, Shifeng Guo, Jie Bai, and Qiang Shu. Sensing system of environmental perception technologies for driverless vehicle: A review of state of the art and challenges. Sensors and Actuators A: **Physical,** page 112566, 2021.

[8] Dylan Cashman, Genevieve Patterson, Abigail Mosca, Nathan Watts, Shannon Robinson, and Remco Chang. Rainbow:Visualizing learning via backpropagation gradients in rnns. IEEE Computer Graphics and Applications, 38(6):39–50, 2018.

[9] Seonghye Kim, Kiichi Hirota, Takahiro Nozaki, and Toshiyuki Murakami. Human motion analysis and its application to walking stabilization with cog and **zmp.** IEEE Transactions on Industrial Informatics, 14(11):5178–5186, 2018.

[10] Jong-Sung Kim and Myung-Gyu Kim. **3d articulated** human motion analysis system using a single low-cost rgb-d sensor. In 2018 International Conference on Information and Communication Technology Convergence (ICTC), pages 1446–1448, 2018.

[11] Milad Nazarahari, Alireza Noamani, Niloufar Ahmadian, and Hossein Rouhani. Sensor-to-body calibration procedure for clinical motion analysis of lower limb using magnetic and inertial measurement units. Journal of Biomechanics, 85:224–229, 2019.

[12] Hans Kainz, Martin Hajek, Luca Modenese, David J. Saxby, David G. Lloyd, and Christopher P. Carty. Reliability of functional and predictive methods to estimate the hip joint centre in human motion analysis in healthy adults. Gait Posture, 53:179–184, 2017.

[13] Timothy R. Derrick, Antonie J. van den Bogert, Andrea Cereatti, Raphael Dumas, Silvia Fantozzi, and Alberto Leardini. Isb recommendations on the reporting of intersegmental forces and moments during human motion analysis. Journal of Biomechanics, 99:109533, 2020.

[14] Tomasz Hachaj, Marek R. Ogiela, Marcin Piekarczyk, and Katarzyna Koptyra. Advanced human motion analysis and visualization: Comparison of mawashi-geri kick of two elite karate athletes. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pages 1–7, 2017.

[15] Ivan Mutis, Abhijeet Ambekar, and Virat Joshi. Realtime space occupancy sensing and human motion analysis using deep learning for indoor air quality control. Automation in Construction, 116:103237, 2020.

[16] Haifeng Wu, Qing Huang, Daqing Wang, and Lifu Gao. A cnn-svm combined model for pattern recognition of knee motion using mechanomyography signals. Journal of Electromyography and Kinesiology, 42:136–142, 2018.

[17] Shudong Yang, Xueying Yu, and Ying Zhou. **Lstm** and gru neural network performance comparison study: Taking the Yelp review dataset as an example. In 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), pages 98–101, 2020.

[18] Xiaoran Shi, Yaxin Li, Feng Zhou, and Lei Liu. **Human** activity recognition based on deep learning methods. In 2018 International Conference on Radar (RADAR), **pages 1–5.** IEEE, 2018.

[19] Kun Xia, Jianguang Huang, and Hanyu Wang. Lstm-cnn architecture for human activity recognition. IEEE Access, 8:56855–56866, 2020.

[20] Zhigang Tu, Wei Xie, Qianqing Qin, Ronald Poppe, Remco C Veltkamp, Baoxin Li, and Junsong Yuan. Multi-stream cnn: Learning representations based on human-related regions for action recognition. Pattern **Recognition,** 79:32–43, 2018.

[21] Ajay Shrestha and Ausif Mahmood. **Review of deep** learning algorithms and architectures. IEEE Access, 7:53040–53065, 2019.

[22] Rogerio E da Silva, Jan Ondrej, and Aljosa Smolic. ´ Using lstm for automatic classification of human motion **capture data.** In VISIGRAPP (1: GRAPP), pages 236–243, 2019.

[23] Schalk Wilhelm Pienaar and Reza Malekian. **Human** activity recognition using lstm-rnn deep neural network **architecture**. In 2019 IEEE 2nd Wireless Africa Conference (WAC), pages 1–5. IEEE, 2019.

[24] Anuradhi Malshika Welhenge and Attaphongse Taparugssanagorn. Human activity classification using a long short-term memory network. Signal, Image **and Video Processing,** 13(4):651–656, 2019.

[25] Sakorn Mekruksavanich and Anuchit Jitpattanakul. Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models. Electronics, 10(3):308, 2021.

[26] Mustafa, Syed Khalid, et al. "Interesting Applications of Mobile Robotic Motion by using Control Algorithms." (2020).