19ECE499 Project Phase II

SKELETAL ANALYSIS USING COMPUTER VISION AND DEEP LEARNING TECHNIQUES TO DETECT STROKE

ADARSH SINGH	CB.EN.U4CCE19002
ADITHYA A S	CB.EN.U4CCE19004
ASHWIN G	CB.EN.U4CCE19011
NIKHIL A	CB.EN.U4CCE19038
ROHAN C	CB.EN.U4CCE19045

Project ID

DSHVLCCE01

Guided by

Dr. Santhosh Kumar C

INTRODUCTION TO THE PROBLEM

PROBLEM STATEMENT:

- Shortage of skilled neurologists and inaccessibility for people from rural areas restrict the facilities of good healthcare to the people.
- People fail to diagnose the neurological disorders early, hence problem becomes worse.
- The diagnostic process involves invasive procedures such as MRI, which may not be suggested for infants and young children.
- Progress and feedback of the treatment is not monitored properly.

MOTIVATION & OBJECTIVES

- Model which identifies potential features for gait tracking.
- Visual representation of real-time posture using graphs.
- LSTM learning model to improve the accuracy of the stroke detection model.
- Real-time system to detect stroke in person, enabling the deployment wherever feasible.
- To develop a diagnostic tool for stroke detection in children can have a great impact on public health, improving the lives of the kids and their families.

LITERATURE SURVEY

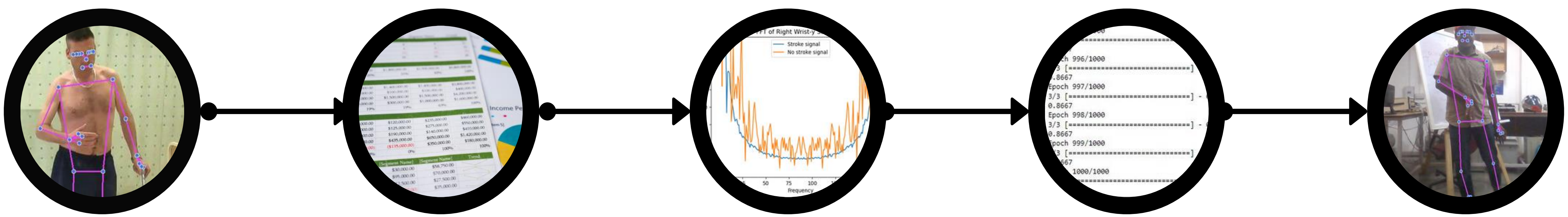
TITLE	AUTHOR	Year, Journal	Summary
A deep learning-based approach for the classification of gait dynamics in subjects with a neurodegenerative disease	Paragliola, Giovanni, and Antonio Coronato	2020, Springer	• addressing gait dynamics by exploring deep learning algorithms like LSTM and CNN. • Carried out various experiments to prove the accuracy of the proposed model • able to obtain a classification accuracy that is 3.9% higher than existing models
Deep Convolutional and LSTM Neural Network Architectures on Leap Motion Hand Tracking Data Sequences	K. Kritsis, M. Kaliakatsos-Papakostas, V. Katsouros, A. Pikrakis	2020, EUSIPCO	• Focuses on addressing and mitigating hand gesture recognition problem by implementing either CNN, LSTM or a combination of both • Input is a multidimensional time series signal attained from a Leap Motion Sensor while the output is a predefined set of gestures • Proposed CNN-LSTM and deep CNN models demonstrate recognition rates of 94% hence outperforming previously proposed models

TITLE	AUTHOR	Year, Journal	Summary
Human action recognition and art interaction based on convolutional neural network	Z. Cai, Y. Yang, L. Lin	2020, Chinese Automation Congress (CAC)	• Utilizes CNN to study the human detection algorithm • Uses vital frame extraction, joint point optimization, and improved bottom-up algorithm to create a new neural network • able to obtain an accuracy of 93%
Fall detection system based on real-time pose estimation and SVM	Y. Chen, R. Du, K. Luo, Y. Xiao	2021, IEEE	• Proposes a method combining pose estimation using yolov5 for accurate fall detection in surveillance videos • Involves creating a dataset from video frames, training an improves network and then using the optimized model to detect and locate targets in the video • Drastically improves the effectiveness to detect falls or Activities of Daily Living (ADL) events in each frame using SVM
Human Body Pose Estimation and Applications	Amrutha, K., P. Prabu, Joy Paulose	2021, IEEE	• Proposes a real-time approach for sign language detection and recognition in videos • Holistic pose estimation method of MediaPipe is utilized • Able to attain a high recognition accuracy for various sign language words performed by 5 singers in natural background

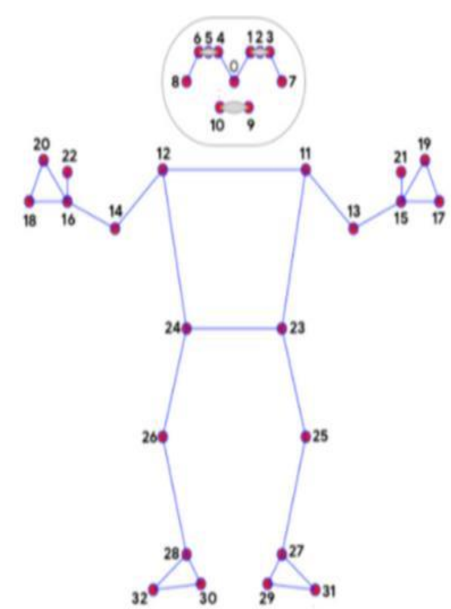
DETAILED PROPOSED MODEL

BLOCK DIAGRAM

Pose Estimation → **Data collection and preprocessing** → **Data Analysis** → **Model Training and validation** → **Real-time detection**



1. Pose Estimation



- The tool used for pose estimation is **MediaPipe,** a framework developed by google.
- It works by tracing semantic key points/joints in human body using pre-trained deep learning models.
- Provides coordinates of those detected pose points.
- Points we will be considering is **7,8,9,10,11,12,13,14,15,16**.

2. Data collection and Preprocessing

- **X** and **Y** coordinates of pose points collected by MediaPipe is scaled and exported to a **CSV** file.
- This data is then flattened to **100*2112** and **100*176*12** and exported to separate **HPF5** files.
- Those files are split into two for training and testing (**80-20**).

3. Data Analysis

- **Time domain analysis** and **frequency domain analysis** was performed on the coordinate points collected.
- This is done to extract meaningful features such as **mean**, **standard deviation** and analyze movement patterns of each joint.

4. Model Training and Validation

- We have trained 3 different models such as **SVM, LSTM AND CNN** under different configurations.
- They have been validated for accuracy and the scores where compared, to find out the most efficient model.

- ### Support Vector Machine (SVM)
  - It is a simple machine learning model which finds a hyperplane that separates data into classes, maximizing the margin between them.
  - The data we used to train is the **standard deviation** values of each of the pose points of multiple persons with stroke and without stroke.

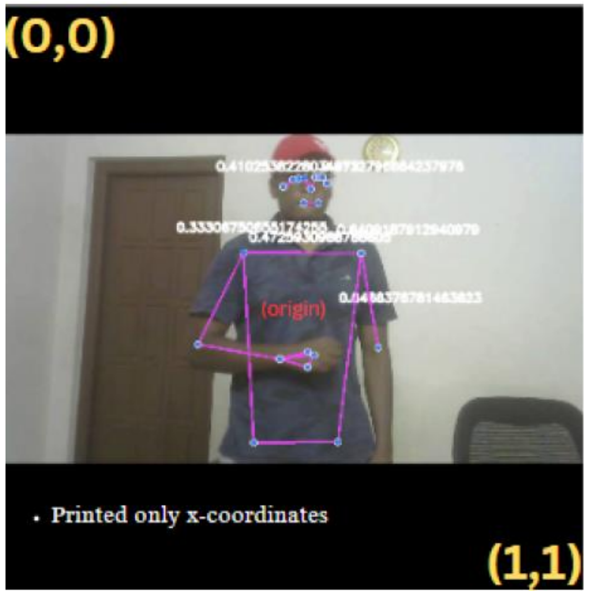
- ### Long Short Term Memory (LSTM)
  - It is a type of recurrent neural network (RNN) architecture designed to process and model sequential data.
  - The architecture we built is sequential which consists of a **LSTM layer, dropout layer** and an fully connected **dense layer.**
  - Additionally added **ReLU** activation to bring linearity in the data and to remove negative values.
  - Input shape of data to the model is **100*2112.**
  - **Sigmoid** activation function is used in fully connected dense layer to get output.
- ### Convolution Neural Network (CNN)
  - It is a powerful type of deep learning model which is designed to learn and extract spatial features from the previous input layer data by performing convolution operations through various layers.
  - The architecture we built is sequential which consists of a **convolution layer, Pooling layer, flattening layer, dropout layer (0.5)** and an fully connected **dense layer.**
  - Added **ReLU** activation function to convolution layer.
  - A **12 channel 1D** CNN is used so, input shape of data to the model is **100*172*12.**
  - **Sigmoid** activation function is used in fully connected dense layer to get output.

5. Real-time stroke detection

- Live video is fed to the model to get the prediction.

TABULAR AND GRAPHICAL RESULTS

1. Pose Estimation



- The **(x,y)** coordinates for each key point are expressed as a value between 0 and 1
- **(0,0)** represents the top-left corner of the image.
- **(1,1)** represents the bottom-right corner of the
- image.

2. Data collection and Preprocessing

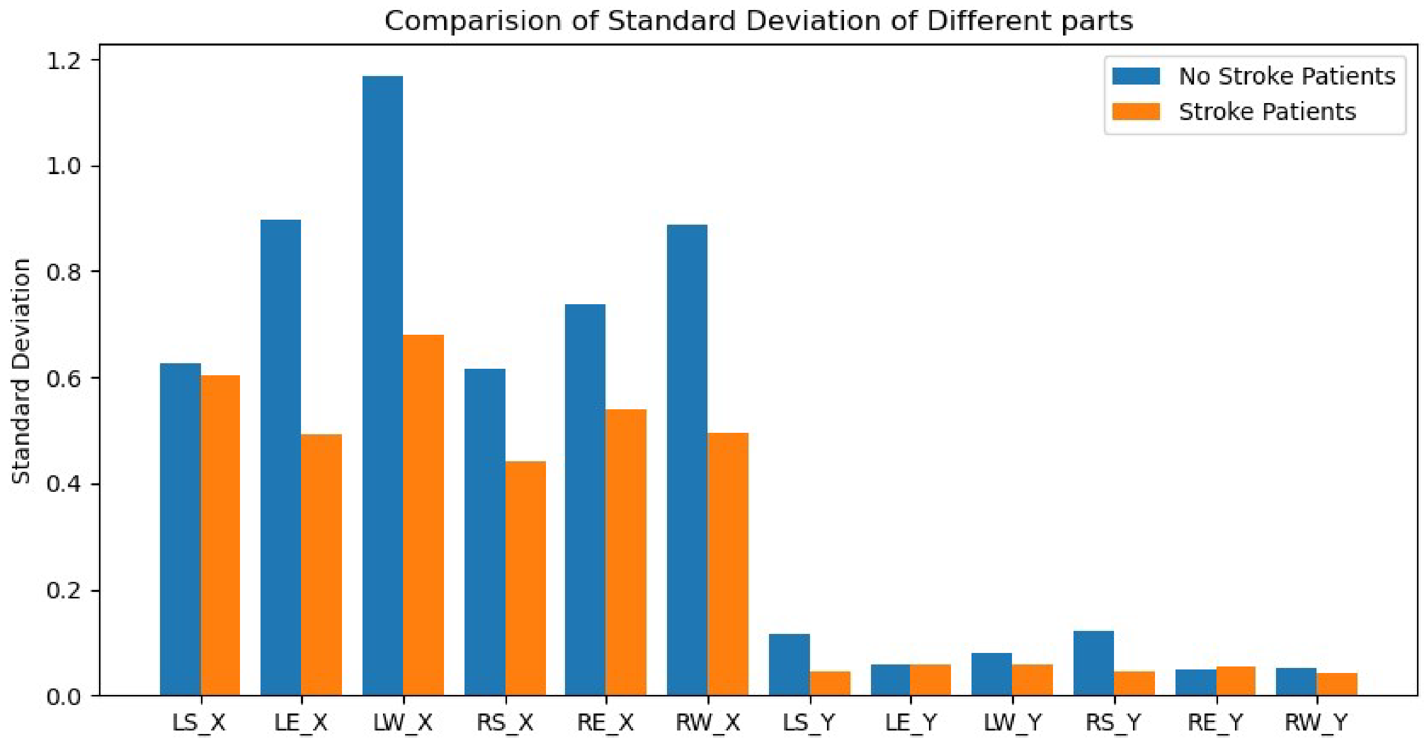
	Left Shoulder-x	Right Shoulder-y	Left Elbow-x	Left Elbow-y	Left Wrist-x	Left Wrist-y	Right Shoulder-x	Right Shoulder-y	Right Elbow-x	Right Elbow-y	Right Wrist-x	Right Wrist-y
0	0.520817	0.221843	0.561017	0.323849	0.587648	0.431031	0.428291	0.238881	0.397328	0.360406	0.430622	0.449738
1	0.520703	0.221466	0.561056	0.324329	0.587828	0.430984	0.428055	0.238883	0.397867	0.360290	0.430962	0.450007
2	0.520487	0.221345	0.561079	0.324692	0.587896	0.430932	0.427737	0.238996	0.398572	0.360221	0.431248	0.450003
3	0.519976	0.220978	0.561004	0.324771	0.587888	0.430484	0.427177	0.238982	0.399045	0.360056	0.431357	0.450035
4	0.519641	0.220327	0.561005	0.324715	0.587849	0.429838	0.426450	0.238465	0.399071	0.359646	0.431337	0.449875
...	...	...	...	...	...	...	...	...	...	...	...	...
216	0.563641	0.236492	0.586436	0.349675	0.607477	0.454112	0.472563	0.259879	0.446740	0.379400	0.493142	0.409897
217	0.563113	0.236411	0.585787	0.349656	0.606615	0.454063	0.471777	0.259667	0.443565	0.378949	0.492005	0.409692
218	0.562811	0.237155	0.585182	0.350393	0.605353	0.455114	0.471410	0.259908	0.441698	0.378746	0.490389	0.410255
219	0.562370	0.237476	0.584346	0.350403	0.603952	0.455070	0.471352	0.260873	0.440675	0.378865	0.489559	0.410918
220	0.561875	0.237699	0.584007	0.350325	0.603098	0.454936	0.471019	0.260865	0.440466	0.378809	0.489088	0.410901

Coordinates collected by MediaPipe

	LS_X	LE_X	LW_X	RS_X	RE_X	RW_X	LS_Y	LE_Y	LW_Y	RS_Y	RE_Y	RW_Y	OUTCOME
0	2.426571	2.536138	2.178487	1.451712	1.243757	1.466065	0.066028	0.119412	0.153275	0.070711	0.118968	0.189475	0
1	0.334790	0.331785	0.548624	0.266962	0.496314	0.266788	0.010301	0.015554	0.018216	0.014559	0.015740	0.017298	1
2	0.509808	0.534023	0.514044	0.785275	0.885417	1.020383	0.081428	0.036071	0.037133	0.081172	0.035701	0.037438	0
3	0.356345	0.353147	0.545111	0.284150	0.518656	0.283964	0.008986	0.012648	0.014563	0.013645	0.014373	0.018353	1
4	0.902177	1.034640	1.423496	1.193646	1.194152	1.496157	0.014408	0.040289	0.072358	0.013542	0.039334	0.068512	1
5	0.992157	1.012567	1.038031	0.805104	0.714616	0.770867	0.051395	0.066337	0.080624	0.052760	0.067703	0.093827	1
6	0.531969	0.499761	0.482927	0.583379	0.490607	0.518248	0.077786	0.036400	0.033931	0.077084	0.031786	0.029259	0
7	0.333295	0.528231	0.518588	0.528938	0.523715	0.531379	0.012152	0.014830	0.055720	0.012472	0.015329	0.024645	1
8	0.601409	0.610943	0.702389	0.612848	0.717985	0.772389	0.107898	0.057345	0.115148	0.108311	0.059157	0.104653	0
9	0.411324	0.410520	0.599543	0.403763	0.492968	0.605097	0.006805	0.011029	0.037178	0.006890	0.009839	0.013890	1
10	0.645971	0.652521	0.829691	0.387301	0.549229	0.514954	0.082479	0.114960	0.124617	0.077141	0.101319	0.072328	1
11	0.627068	0.898360	1.169192	0.615274	0.738676	0.885804	0.115614	0.058868	0.080073	0.121917	0.048374	0.051448	0
12	0.708063	0.665803	0.574431	0.646385	0.612081	0.645683	0.029291	0.019909	0.014410	0.029165	0.019099	0.015283	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

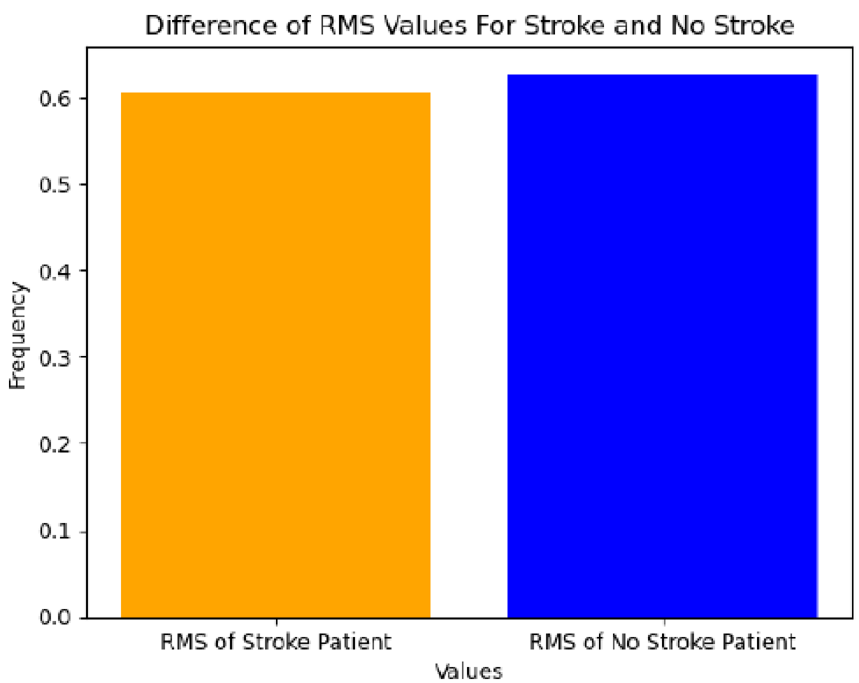
Dataset of standard deviation values of each of pose points

3. Data Analysis



Comparison of Standard Deviation of Different parts

- We have observed that individuals without stroke have had **higher magnitude** of **standard deviation** values than those with stroke.
- This result may be explained by the fact that non-stroke patients have higher range of motion in their hands and legs and are able to carry out a wider variety of actions.

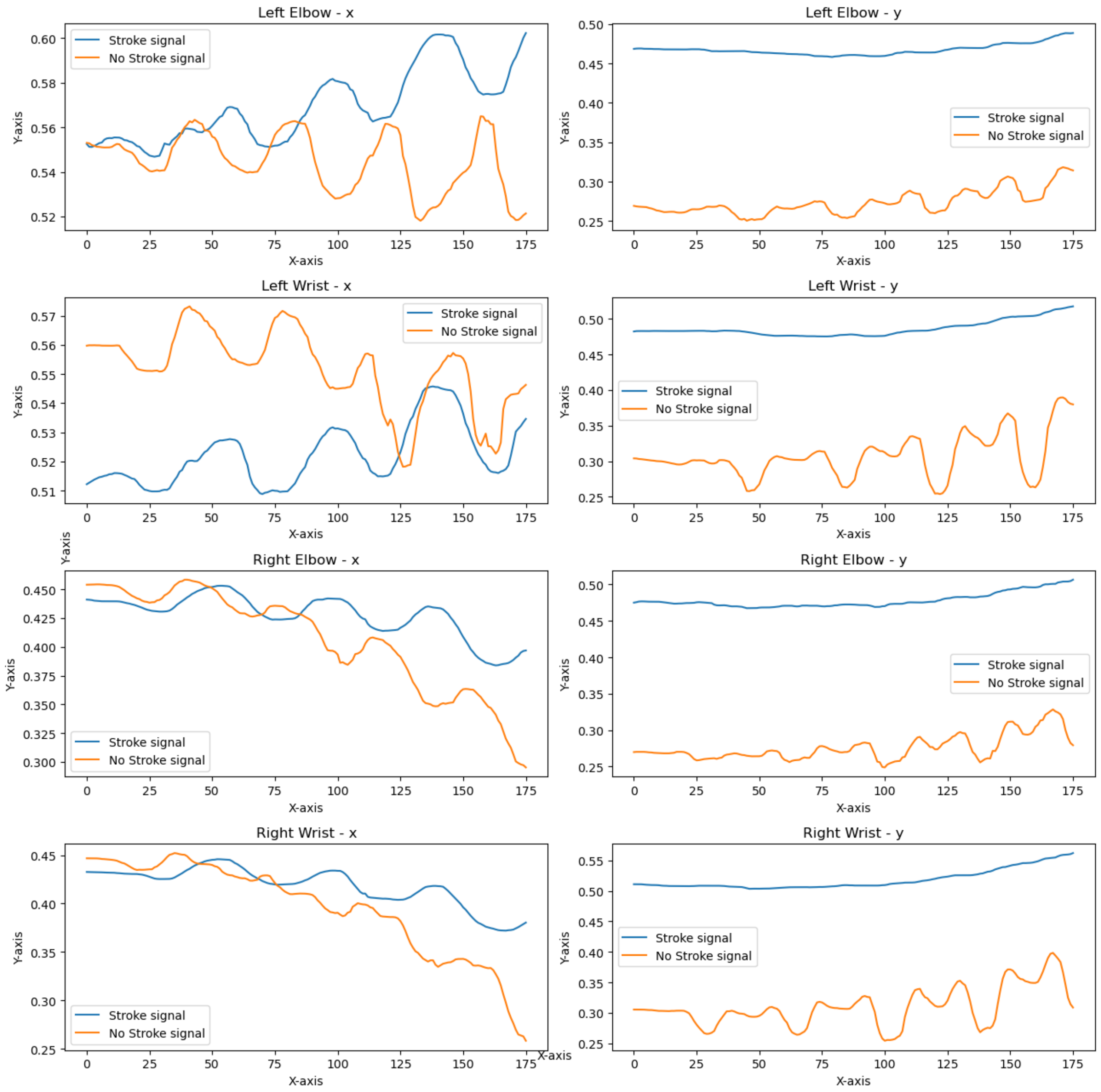


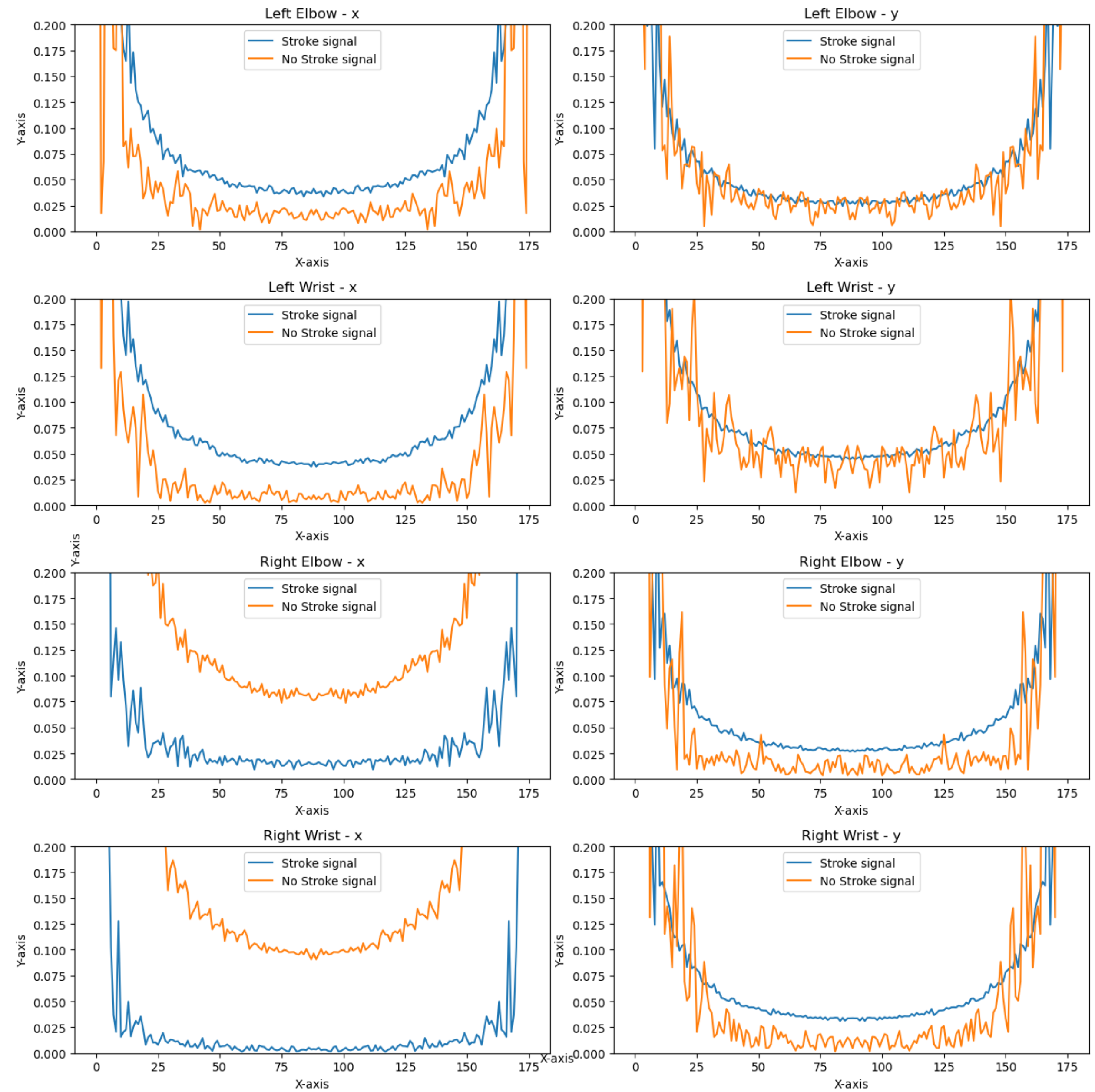

```
Stroke RMS =  0.604952901143718 No Stroke RMS=  0.6270678568782155
```

Comparison of RMS value of stroke and no stroke person

- Similarly, the **RMS value** was found to be **higher** in individuals without stroke compared to those with stroke.
- This is because non-stroke patients can perform larger and more diverse movements, resulting in a greater range of motion and higher amplitude of movements unlike stroke patients.

Time and Frequency domain analysis





Time domain analysis

Frequency domain analysis

4. Model Training and Validation

SVM

```
# Evaluate the accuracy of the SVM classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy percentage using svm classifier:", accuracy*100)
```

```
Accuracy percentage using svm classifier: 69.3333
```

- **Standard deviation** dataset was fed to an SVM model.
- **Accuracy** provided by the model is **69.33%**

LSTM

```
Epoch 796/800
3/3 [==============================] - 0s 34ms/step - loss: 0.0154 - accuracy: 1.0000 - val_loss: 0.4987 - val_accuracy: 0.75
00
Epoch 797/800
3/3 [==============================] - 0s 32ms/step - loss: 0.0181 - accuracy: 1.0000 - val_loss: 0.5427 - val_accuracy: 0.75
00
Epoch 798/800
3/3 [==============================] - 0s 33ms/step - loss: 0.0153 - accuracy: 1.0000 - val_loss: 0.6403 - val_accuracy: 0.75
00
Epoch 799/800
3/3 [==============================] - 0s 31ms/step - loss: 0.0196 - accuracy: 1.0000 - val_loss: 0.4194 - val_accuracy: 0.75
00
Epoch 800/800
3/3 [==============================] - 0s 32ms/step - loss: 0.0146 - accuracy: 1.0000 - val_loss: 0.5884 - val_accuracy: 0.75
00
```

- Model was trained with **800 epoch**
- **Validation accuracy** provided by the model was **75%**

CNN 12 Channel 1D

```
Epoch 995/1000
3/3 [==============================] - 0s 54ms/step - loss: 5.1337e-04 - accuracy: 1.0000 - val_loss: 0.4710 - val_accuracy:
0.8667
Epoch 996/1000
3/3 [==============================] - 0s 51ms/step - loss: 5.1751e-04 - accuracy: 1.0000 - val_loss: 0.4675 - val_accuracy:
0.8667
Epoch 997/1000
3/3 [==============================] - 0s 46ms/step - loss: 5.1778e-04 - accuracy: 1.0000 - val_loss: 0.4399 - val_accuracy:
0.8667
Epoch 998/1000
3/3 [==============================] - 0s 53ms/step - loss: 5.3571e-04 - accuracy: 1.0000 - val_loss: 0.4375 - val_accuracy:
0.8667
Epoch 999/1000
3/3 [==============================] - 0s 54ms/step - loss: 5.3324e-04 - accuracy: 1.0000 - val_loss: 0.4672 - val_accuracy:
0.8667
Epoch 1000/1000
3/3 [==============================] - 0s 51ms/step - loss: 5.0318e-04 - accuracy: 1.0000 - val_loss: 0.5327 - val_accuracy:
0.8667
```

- Model was trained with **1000 epoch**
- **Validation accuracy** provided by the model was **86.7%**

CNN 12 Channel 1D – validation results

```python
from sklearn.metrics import f1_score

# Assuming you have y_true (true labels) and y_pred (predicted labels)
f1 = f1_score(y_test, y_pred)
print("F1 Score for 12 Channel 1D CNN Model:", f1)

```

```
F1 Score for 12 Channel 1D CNN Model: 0.888888888888889
```

F1 Score - 0.88

```python
from sklearn.metrics import precision_score

# Assuming you have y_true (true labels) and y_pred (predicted labels)
precision = precision_score(y_true, y_pred)
print("Precision for 12 Channel 1D CNN Model:", precision)

```

```
Precision for 12 Channel 1D CNN Model: 1.0
```

Precision - 0.88

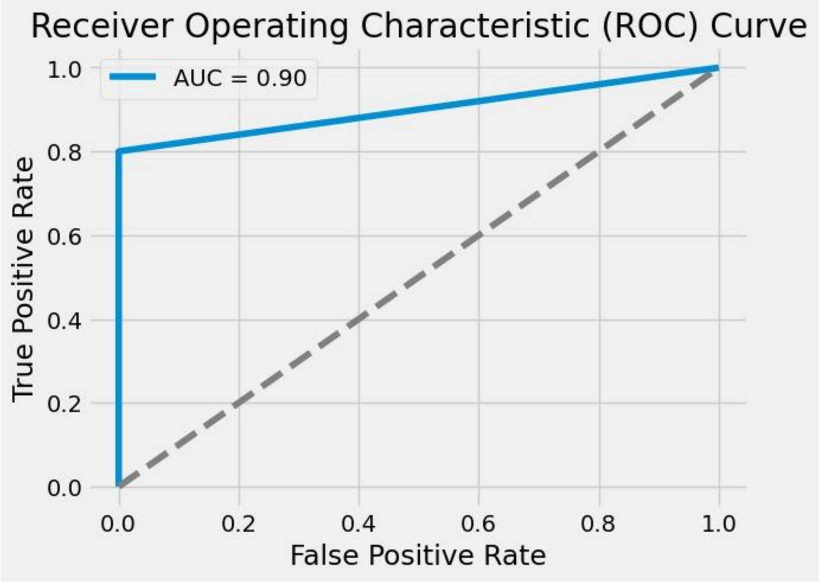
```python
from sklearn.metrics import recall_score

# Assuming you have y_true (true labels) and y_pred (predicted labels)
recall = recall_score(y_test, y_pred)
print("Recall for 12 Channel 1D CNN Model:", recall)

```

```
Recall for 12 Channel 1D CNN Model: 0.8
```

Recall – 0.8



ROC Curve

```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.83      1.00      0.91        10
         1.0       1.00      0.80      0.89        10

    accuracy                           0.90        20
   macro avg       0.92      0.90      0.90        20
weighted avg       0.92      0.90      0.90        20
```

Classification report

```
Confusion Matrix for 12 Channel 1D CNN Model:
[[10  0]
 [ 2  8]]
```

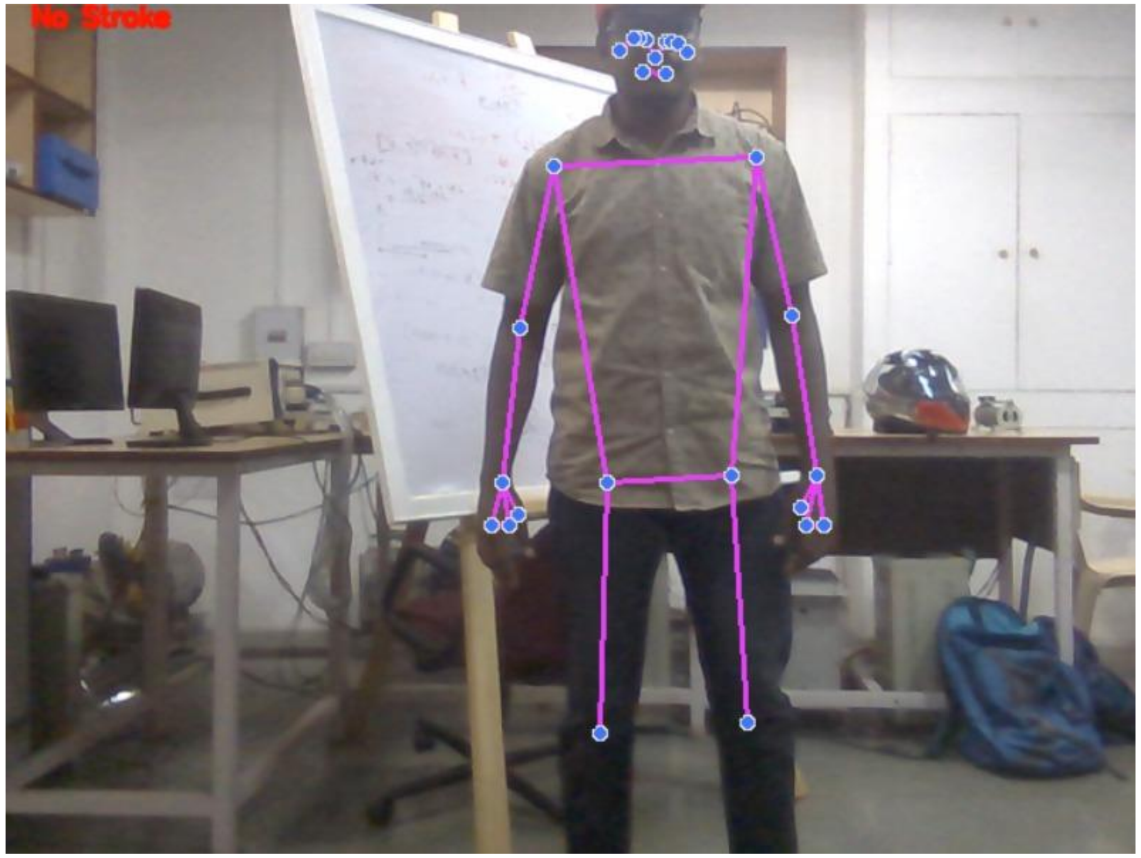
Confusion Matrix

MODEL COMPARISON

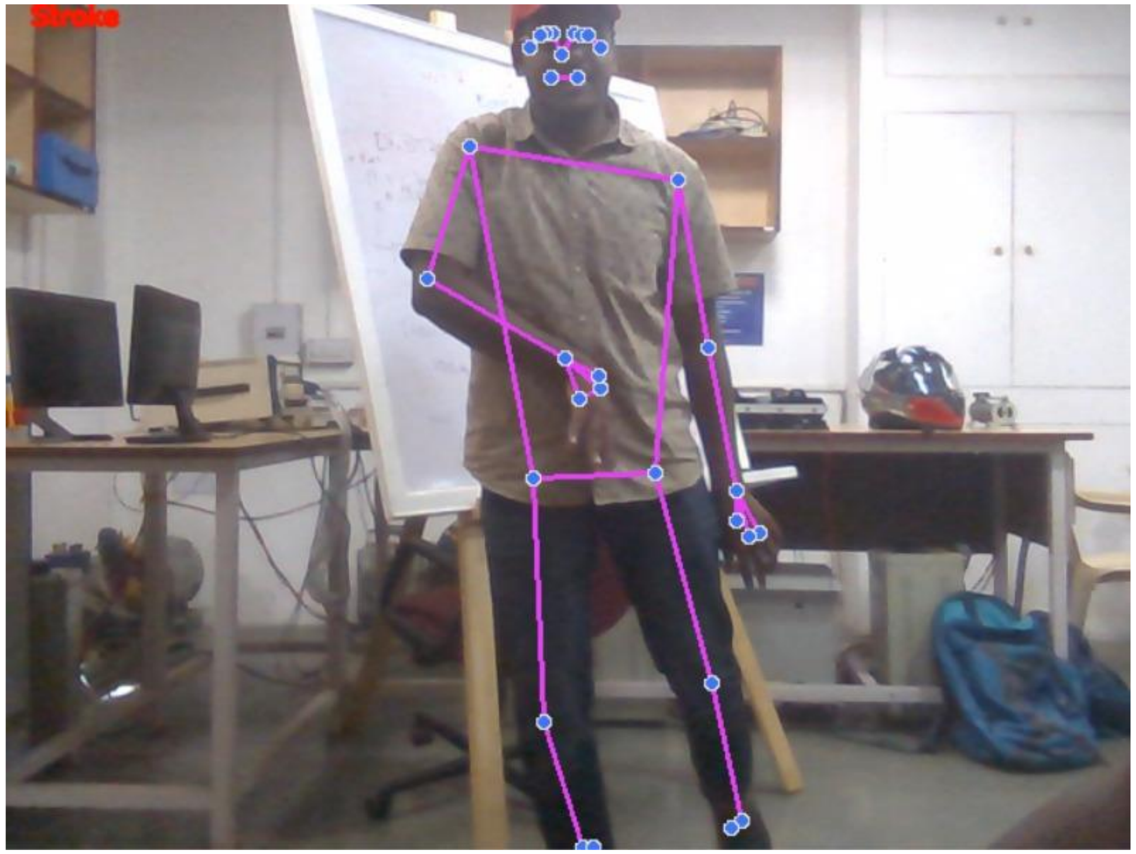
MODEL	VALIDATION SPLIT		
	0.1	**0.2**	**0.3**
SVM	0.63	0.69	0.65
LSTM	0.73	0.75	0.71
CNN	0.82	0.86	0.83

- We can infer that **CNN** has got the most accuracy of **86%** compared to **SVM** and **LSTM.**
- **CNN 1D** with **12 channels** provide the **best** results for validation split of **0.2**.

Real-time stroke detection



No Stroke detected



Stroke detected

- Developed a **real-time working system** for our model.
- The system makes use of a **pre-trained deep learning model** to classify the pose points of a person as either having a stroke or not.
- This system can be used in rehabilitation centers or hospitals where continuous monitoring of stroke patients is required.

ADVANTAGES AND DISADVANTAGES

Advantages

- Mediapipe provides several additional features, such as face detection and tracking, hand tracking, and object detection.
- Latest advancements in deep learning techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory(LSTM), along with pose estimation techniques, have shown appreciable results.
- This real-time working system we have created can be a noteworthy progress in the field of stroke rehab as it has the capacity to monitor the patients continuously and provide real-time feedback on their movements.

Disadvantages

- Lack of labelled training data for complex human actions, which makes it difficult to train deep learning models
- Variability in human actions, such as variations in clothing, lighting, and camera angles, which can affect the performance of the models.
- The computational complexity of deep learning models can make real-time human action recognition challenging on resource-limited devices.

CONCLUSION AND FUTURE SCOPE

- The commonality and detrimental consequences of stroke make it an issue of utmost prevalence in the healthcare sector. it affects people of all ages, making it a leading cause of disability and death globally.
- Although healthcare has advanced to a significant extent, there are many blatant problems when it comes to diagnosing and treating a stroke patient which must be addressed.
- Have utilized the convolutional neural network (CNN) as well as the Mediapipe library for stroke detection and have solved the impending issues listed above in a non-invasive fashion.
- Have leveraged the Mediapipe framework to detect whether a patient has stroke or not by detecting the semantic key points and evaluating if there is an abnormality in the distance between the respective joint coordinates.
- Have concluded that patients with stroke have limited movement in certain joints or limbs, which results in less variability in their movements
- Non-stroke patients have a higher range of motion in their hands and legs, hence they can perform a wider variety of actions.
- Have successfully created 3 stroke detection models that could be refined and improved further.
- This model could also be used to diagnose other diseases with different training data as well.

REFERENCES

- Han, Yull Kyu, and Young Bok Choi. "Human action recognition based on LSTM model using smartphone sensor." In 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), pp. 748-750. IEEE, 2019.
- Zhang, Pengfei, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. "View adaptive neural networks for high performance skeleton based human action recognition." IEEE transactions on pattern analysis and machine intelligence 41, no. 8 (2019): 1963-1978.
- Orozco, Carlos Ismael, Maria Elena Buemi, and Julio Jacobo Berlles. "Towards an attention mechanism LSTM framework for human action recognition in videos." In 2020 IEEE Congreso Bienal de Argentina (ARGENCON), pp. 1-6. IEEE, 2020.
- Deep, Samundra, and Xi Zheng. "Hybrid model featuring CNN and LSTM architecture for human activity recognition on smartphone sensor data." In 2019 20th international conference on parallel and distributed computing, applications and technologies (PDCAT), pp. 259-264. IEEE, 2019.
- Silsupadol, Patima, Paphawee Prupetkaew, Teerawat Kamnardsiri, and Vipul Lugade. "Smartphone-based assessment of gait during straight walking, turning, and walking speed modulation in laboratory and free-living environments." IEEE journal of biomedical and health informatics 24, no. 4 (2019): 1188-1195.
- Luo, Yue, Jimmy Ren, Zhouxia Wang, Wenxiu Sun, Jinshan Pan, Jianbo Liu, Jiahao Pang, and Liang Lin. "Lstm pose machines." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5207-5215. 2018.