Sensor-based Human Locomotion Prediction with Environment Perception and Deep Learning

Zeyu Lu

- Literatures Review
- Preliminary Work
  - Own dataset implementation
    - Logistics regression
    - CNNs
    - RNNs
  - Online dataset implementation
    - Logistics regression
    - CNNs
    - RNNs(LSTM)
  - Preprocess(Minmax, scall, MaxAbs, …)
  - Decoding time-series data into distinguishable images

1- Literatures Review

Review- Deep learning for sensor-based activity recognition

Literatures

Literature	Sensor Modality	Deep Model	Application	Dataset
[2]	Body-worn	SAE	ADL	D03
[3]	Body-worn	RBM	ADL, factory, Parkinson	D02, D06, D14
[6]	Body-worn, ambiemt	RBM	Gesture, ADL, transportation	Self, D01
[10]	Body-worn	CNN	ADL	Self
[11]	Body-worn	CNN	ADL	D06
[12]	Body-worn	DNN	Parkinson	Self
[14]	Body-worn	RNN	ADL	D01, D04, Self
[15]	Object, ambient	DBN	ADL	Self
[16]	Body-worn	CNN	ADL	Self, D01
[17]	Body-worn, object, ambient	RNN	ADL, smart home	D01, D02, D04
[19]	Body-worn	CNN	Factory, health	D02, D13
[18]	Body-worn	CNN	ADL, health	D13
[20]	Body-worn	RBM	Parkinson	Self
[21]	Body-worn, object, ambient	DNN, CNN, RNN	ADL, smart home, gait	D01, D04, D14
[22]	Body-worn	CNN	Gait	Self
[23]	Body-worn, ambient	RBM	ADL, smart home	D16
[26]	Body-worn	RNN	ADL	D16
[27]	Body-worn	CNN	ADL	D03, D05, D11
[29]	Ambient	CNN	Respiration	Self
[31]	Ambient	CNN	Hand gesture	Self
[30]	Body-worn	CNN	ADL	Self
[32]	Body-worn, ambient	RBM	ADL, emotion	Self
[33]	Ambient	RBM	ADL	Self
[36]	Body-worn	CNN	ADL	Self
[37]	Object	RBM	Patient resuscitation	Self
[38]	Object	CNN	Patient resuscitation	Self
[39]	Body-worn	SAE	ADL	D03
[40]	Body-worn	CNN, RBM	ADL	Self
[41]	Body-worn	CNN	ADL, gesture	Self
[42]	Body-worn	CNN	ADL, smart home	D01, D02
[43]	Body-worn	RNN	ADL, smart home	D01, D02, D05, D14
[44]	Body-worn	CNN, RNN	ADL, gesture, posture, factory	D01, D02
[46]	Body-worn	CNN	ADL	Self
[47]	Body-worn, object	RBM	ADL, food preparation, factory	D01, D02, D08, D14
[48]	Body-worn	CNN	PAF disease	D17
[51]	Body-worn	RBM	ADL	D19
[52]	Body-worn	CNN	ADL, factory	D02, D06, D14, D15
[53]	Body-worn	CNN	ADL, factory, Parkinson	D02, D06, D14, D15
[54–56]	Body-worn	CNN	ADL	D03
[57]	Body-worn	CNN, RNN, DNN	ADL, sleep	Self
[59]	Ambient	CNN, RNN	Gait	NA
[61]	Body-worn, object, ambient	DNN	ADL	Self
[62]	Body-worn	DNN	ADL	D03
[65]	Body-worn, ambient	CNN	ADL, location	Self
[63]	Object, ambient	SAE	ADL	NA
[66]	Body-worn, object, ambient	CNN	ADL, smart home, gesture	D01, D18
[68]	Body-worn, object	CNN, RNN	Cartrack, ADL	Self, D19
[69]	Body-worn	CNN	ADL	Self
[70]	Body-worn, ambient, object	CNN	ADL, smart home, factory	D01, D02, D10
[71]	Body-worn	DNN	ADL	Self
[72]	Body-worn	RBM	ADL	Self
[73]	Body-worn	DBN	ADL, smart home	D01, D05, D07
[75]	Object	CNN	Medical	Self
[74]	Body-worn	DNN	ADL	Self
[77]	Body-worn	CNN, SAE	ADL	D04
[76]	Body-worn	CNN	ADL, heart failure	D04, D14

ADL: Activities of Daily Life

1

2

Online public dataset

Public HAR datasets (A=accelerometer, G=gyroscope, M=magnetometer, O=object sensor, AM=ambient sensor, ECG=electrocardiograph).

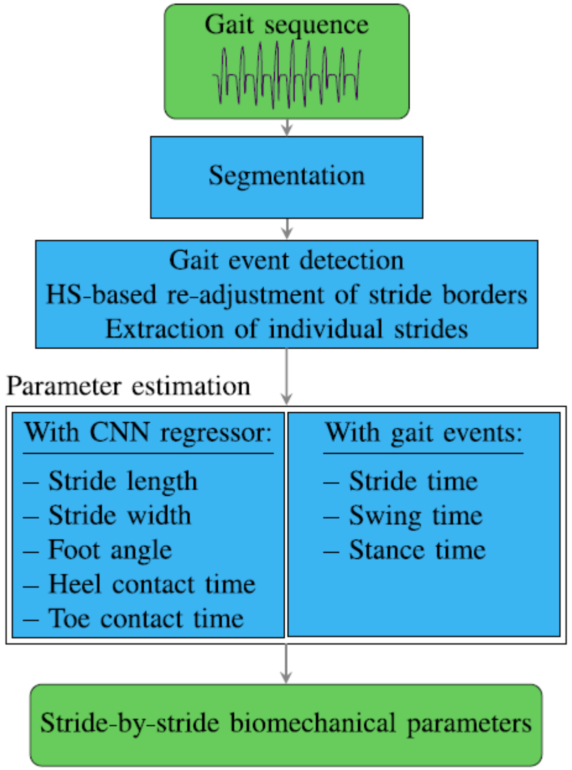
ID	Dataset	Type	#Subject	S. Rate	#Activity	#Sample	Sensor	Reference
D01	OPPORTUNITY	ADL	4	32 Hz	16	701,366	A, G, M, O, AM	[44]
D02	Skoda Checkpoint	Factory	1	96 Hz	10	22,000	A	[47]
D03	UCI Smartphone	ADL	30	50 Hz	6	10,299	A, G	[2]
D04	PAMAP2	ADL	9	100 Hz	18	2,844,868	A, G, M	[76]
D05	USC-HAD	ADL	14	100 Hz	12	2,520,000	A, G	[27]
D06	WISDM	ADL	29	20 Hz	6	1,098,207	A	[3]
D07	DSADS	ADL	8	25 Hz	19	1,140,000	A, G, M	[73]
D08	Ambient kitchen	Food preparation	20	40 Hz	2	55,000	O	[47]
D09	Darmstadt Daily Routines	ADL	1	100 Hz	35	24,000	A	[47]
D10	Actitracker	ADL	36	20 Hz	6	2,980,765	A	[70]
D11	SHO	ADL	10	50 Hz	7	630,000	A, G, M	[27]
D12	BIDMC	Heart failure	15	125 Hz	2	>20,000	ECG	[76]
D13	MHEALTH	ADL	10	50 Hz	12	16,740	A, C, G	[18]
D14	Daphnet Gait	Gait	10	64 Hz	2	1,917,887	A	[21]
D15	ActiveMiles	ADL	10	50–200 Hz	7	4,390,726	A	[53]
D16	HASC	ADL	1	200 Hz	13	NA	A	[23]
D17	PAF	PAF	48	128 Hz	2	230,400	EEG	[48]
D18	ActRecTut	Gesture	2	32 Hz	12	102,613	A, G	[66]
D19	Heterogeneous	ADL	9	100–200 Hz	6	43,930,257	A, G	[68]

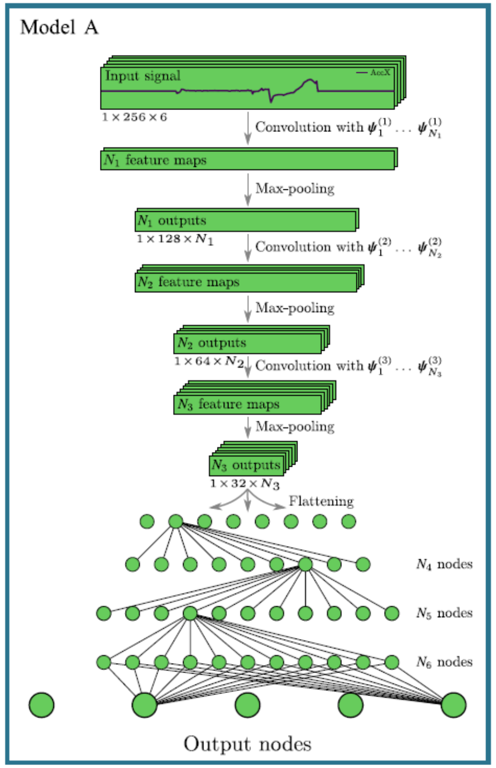
Wang, J., Chen, Y., Hao, S., Peng, X. & Hu, L. Deep learning for sensor-based activity recognition: A Survey. *Pattern Recognit. Lett.* (2018). doi:10.1016/j.patrec.2018.02.010

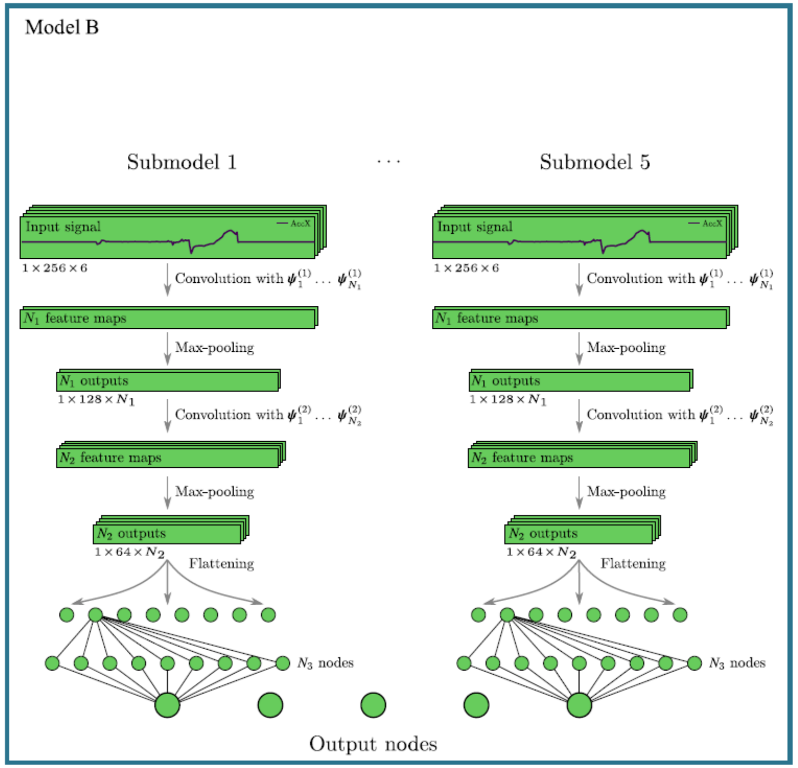
Sensor-Based Gait Parameter Extraction With Deep Convolutional Neural Networks - IEEE Journals & Magazine. Available at: https://ieeexplore-ieee-org.libproxy1.nus.edu.sg/document/7778173. (Accessed: 9th October 2018)

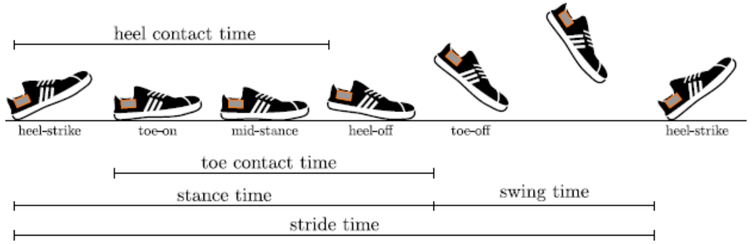
1

Recognition	Methods	Data source	Acc
Gait Parameter	CNNs + Gait events	Self	Above 90%

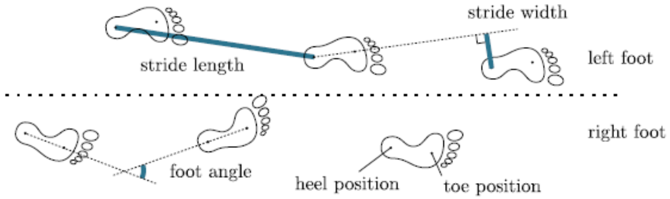








(a)

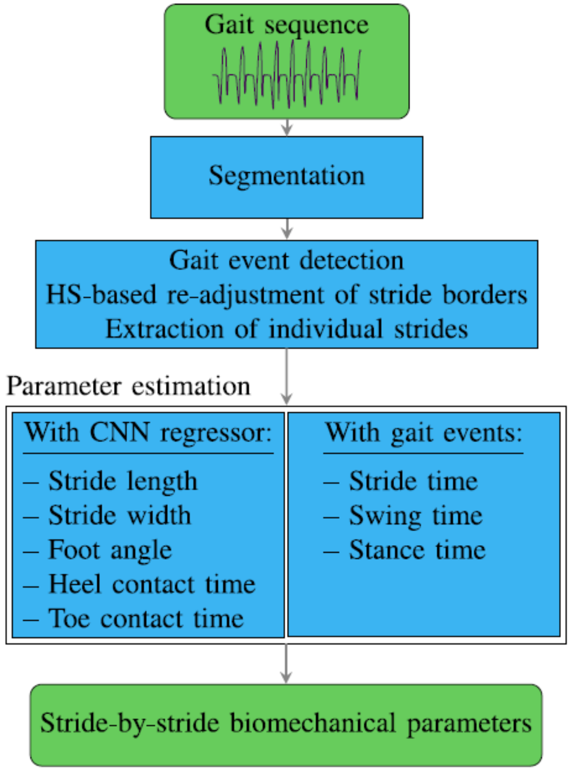


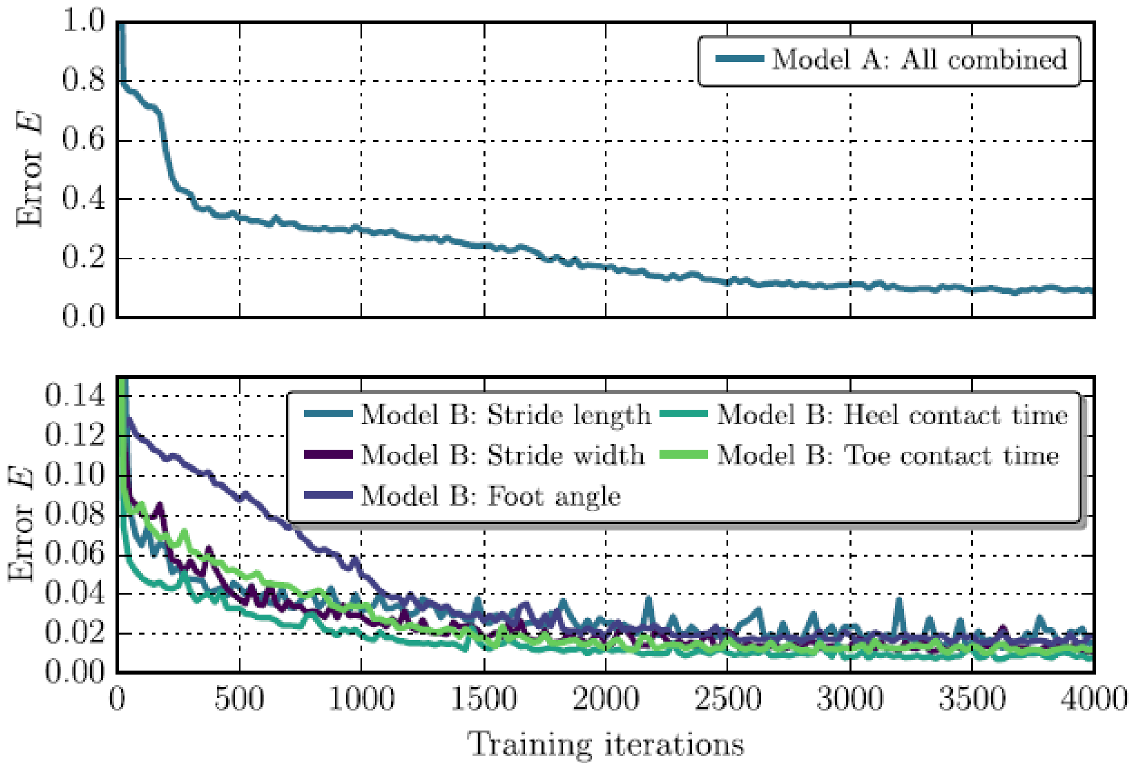
(b)

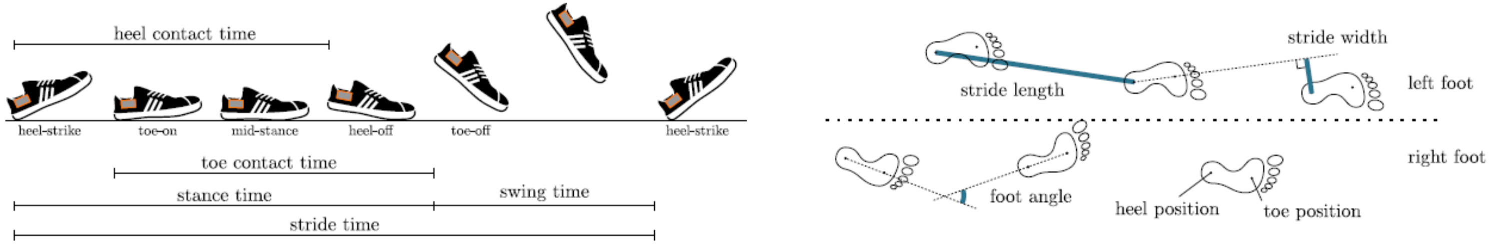
Sensor-Based Gait Parameter Extraction With Deep Convolutional Neural Networks - IEEE Journals & Magazine. Available at: https://ieeexplore-ieee-org.libproxy1.nus.edu.sg/document/7778173. (Accessed: 9th October 2018)

1

Recognition	Methods	Data source	Acc
Gait Parameter	CNNs + Gait events	Self	Above 90%

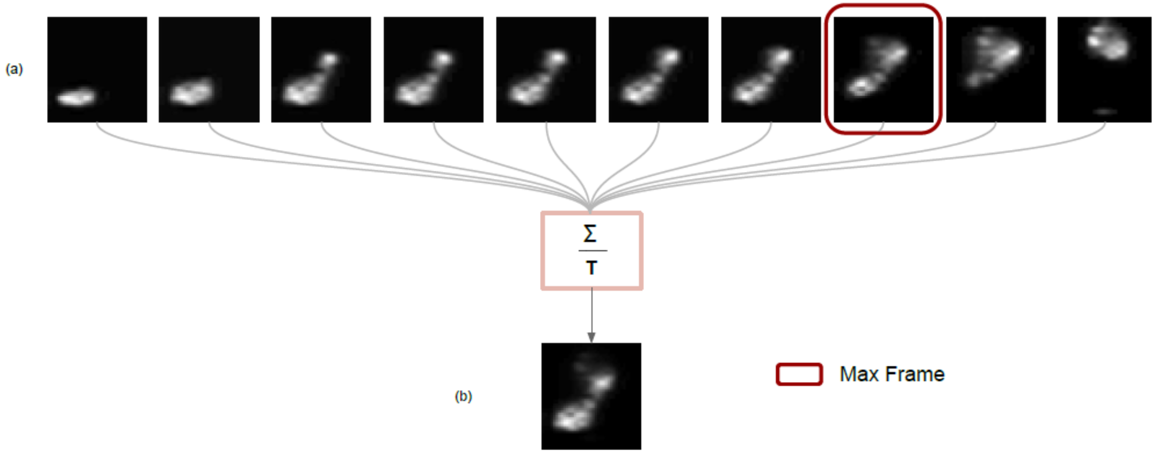






(a) (b)

2 M. S. Singh, V. Pondenkandath, B. Zhou, P. Lukowicz, and M. Liwicki, "Transforming Sensor Data to the Image Domain for Deep Learning - an Application to Footstep Detection," *arXiv:1701.01077 [cs]*, Jan. 2017.

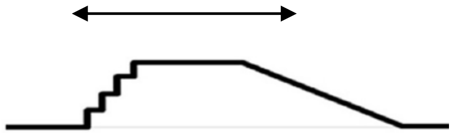


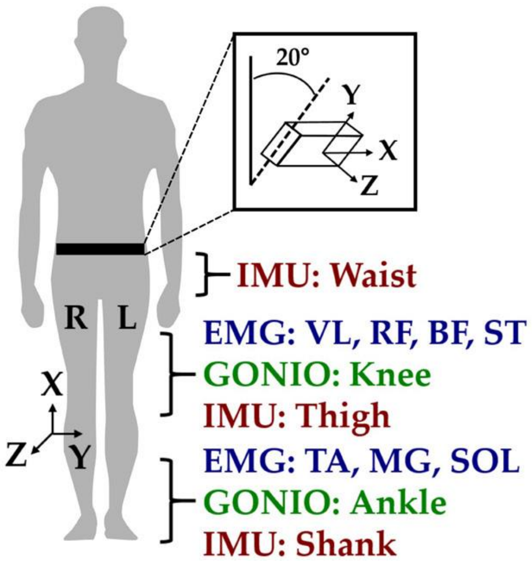
Recognition	Methods	Data source	Acc
Person Identity	CNN+RNN(Transfer learaning-AlexNet)	Self	Below 90%

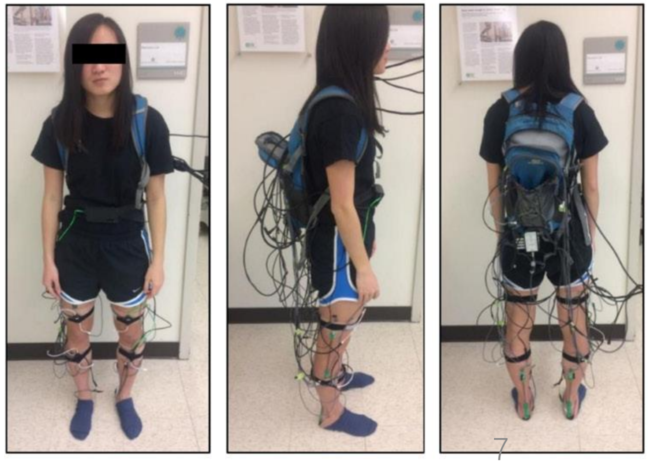
"Frontiers | Benchmark **Datasets** for Bilateral Lower-Limb Neuromechanical Signals from Wearable Sensors during Unassisted Locomotion in Able-Bodied Individuals | Robotics and AI," **2018**.

The dataset contains bilateral EMG and joint and limb kinematics recorded from wearable sensors for **10 able-bodied** individuals as they freely transitioned between **sitting, standing, and several walking-related activities [level ground, stair ascent (SA)/stair descent (SD), and ramp ascent (RA)/ramp descent (RD)]**.

25 repetition

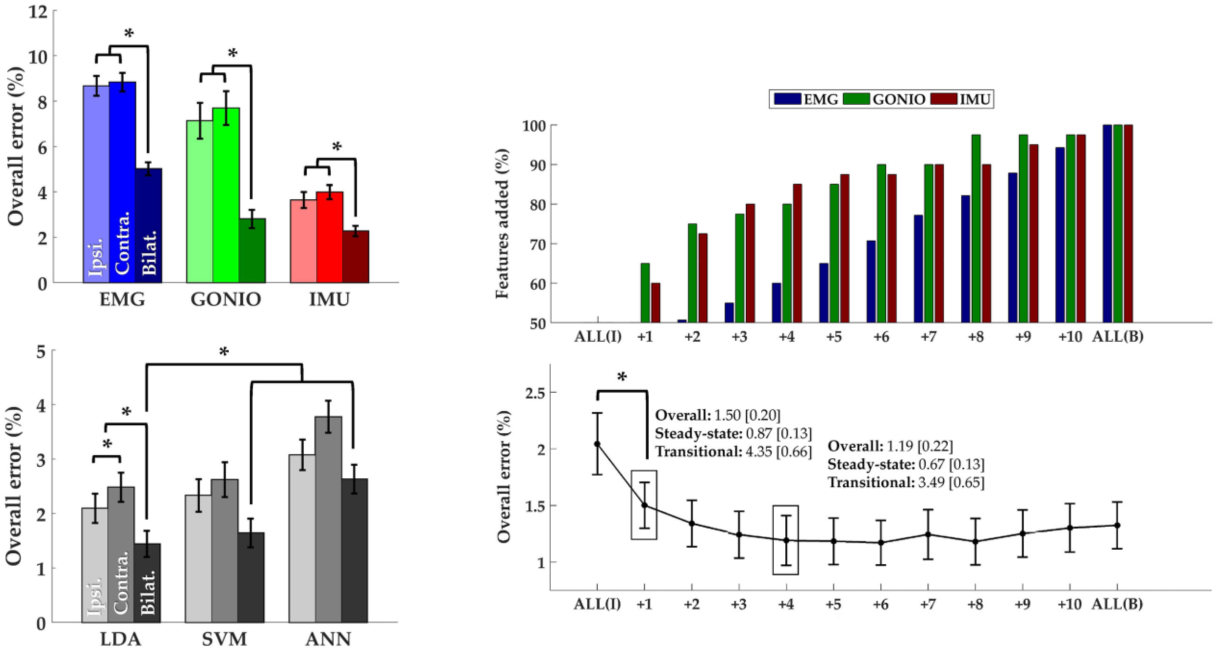


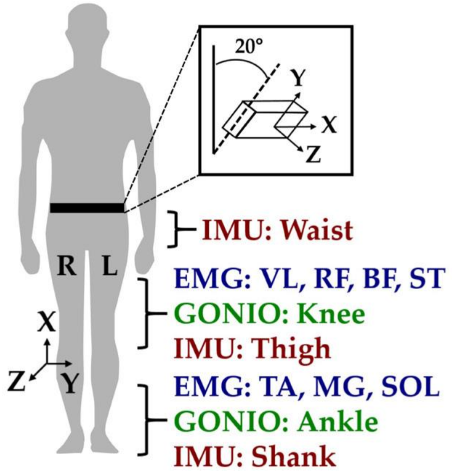


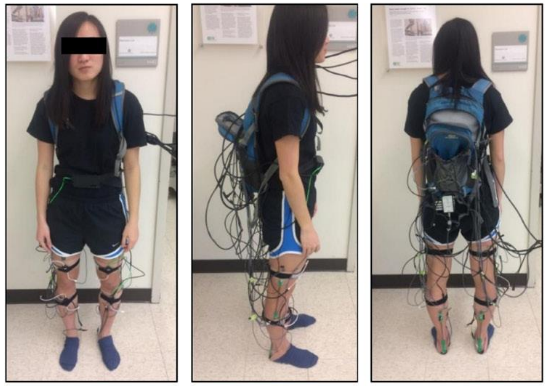


B. Hu, E. Rouse, and L. Hargrove, "Fusion of Bilateral Lower-Limb Neuromechanical Signals Improves Prediction of Locomotor Activities," *Front. Robot. AI*, vol. 5, 2018.

Only one contralateral sensor significantly **reduced** overall, **extraneous sensors** may not only be **redundant** but also detrimental for intent recognition because they contribute to model **overfitting**



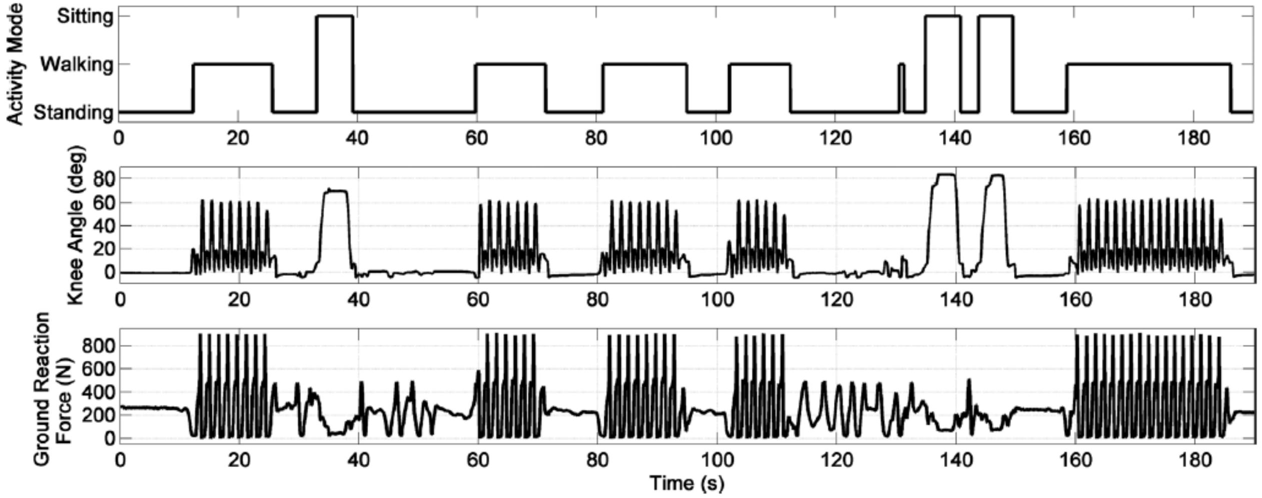




Recognition	Methods	Data source	Acc	Sensors
Contralateral effect	SVM/ANN/LDA	Self	Above 90%	IMU, EMG, Gonio

linear discriminant analysis (LDA)

H. A. Varol, F. Sup, and M. Goldfarb*, “Multiclass **Real-Time** Intent Recognition of a Powered Lower Limb Prosthesis,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 3, pp. 542–551, Mar. 2010.



Recognition	Methods	Data source	Acc	Sensors
Sitting, walking, standing**(*)**	GMM	Self	Above 90%	Ground force, Goniometer

Gaussian mixture models(GMMs)

Y. Massalin, M. Abdrakhmanova, and H. A. Varol, “User-Independent Intent Recognition for Lower Limb Prostheses Using Depth Sensing,” *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 8, pp. 1759–1770, Aug. 2018.

Recognition	Methods	Data source	Acc	Sensors
SD, SA, walking, standing, running(*)	SVM + Voting Filter	Self	Above 90%	Depth sensing

J. Kang, J. Kim, S. Lee, and M. Sohn, “Transition activity recognition using fuzzy logic and overlapped sliding window-based convolutional neural networks,” *J Supercomput*, Jul. 2018.

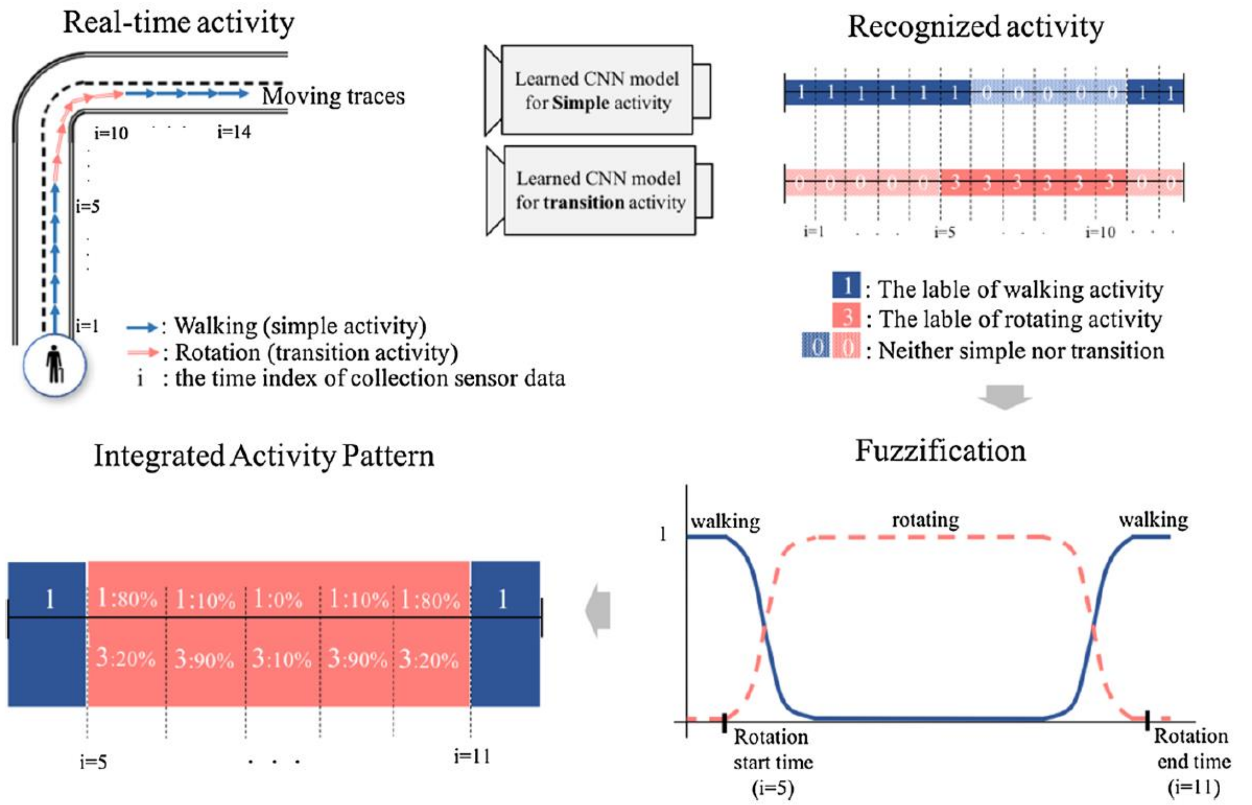
Modes: turn to left or right, stand up, and travel down the stairs (Transition activities)
walking, running, and standing(simple activities)

Methods: CNN + fuzzy logic

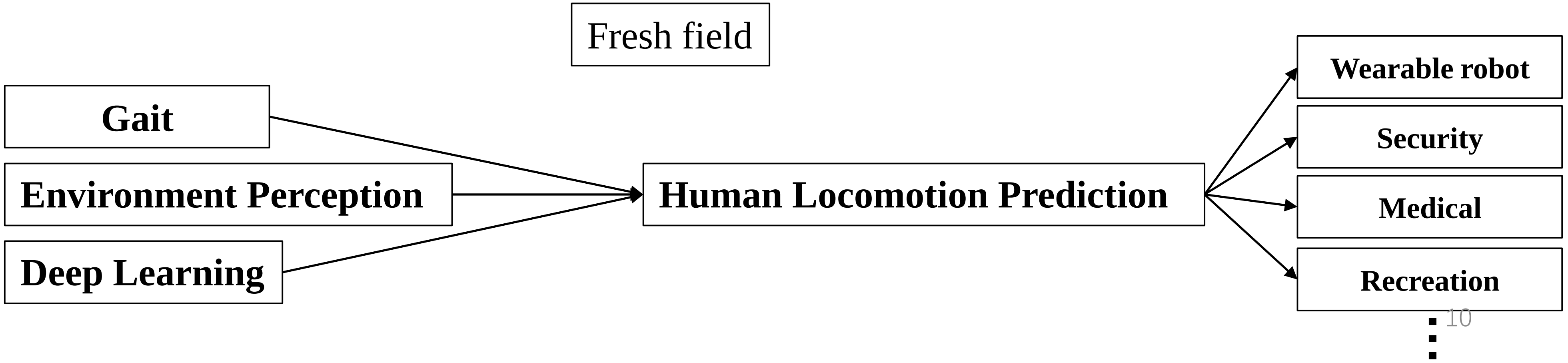
Data source: **smartphone (*)**

Acc: Above 95% for the simple activities
Above 95% for the transition activities

most researchers are concerned only with the recognition of simple activities like walking, running, sitting, and/or standing

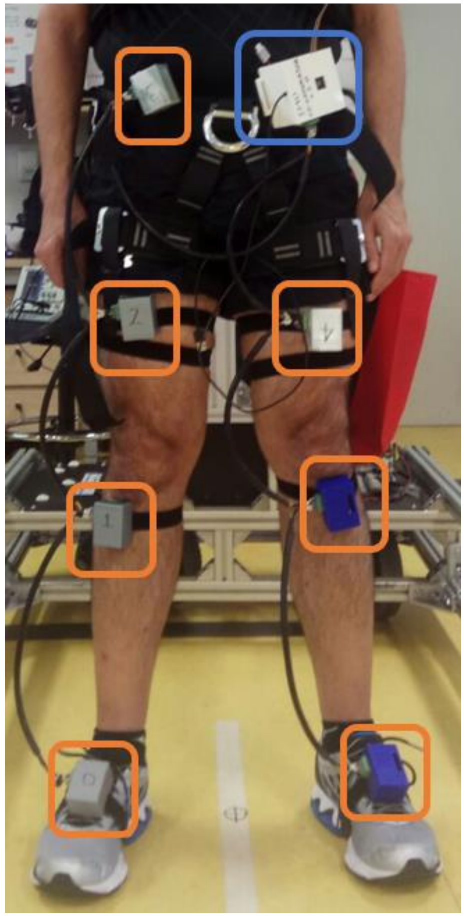


Conclusion



2- Preliminary Work

Setup



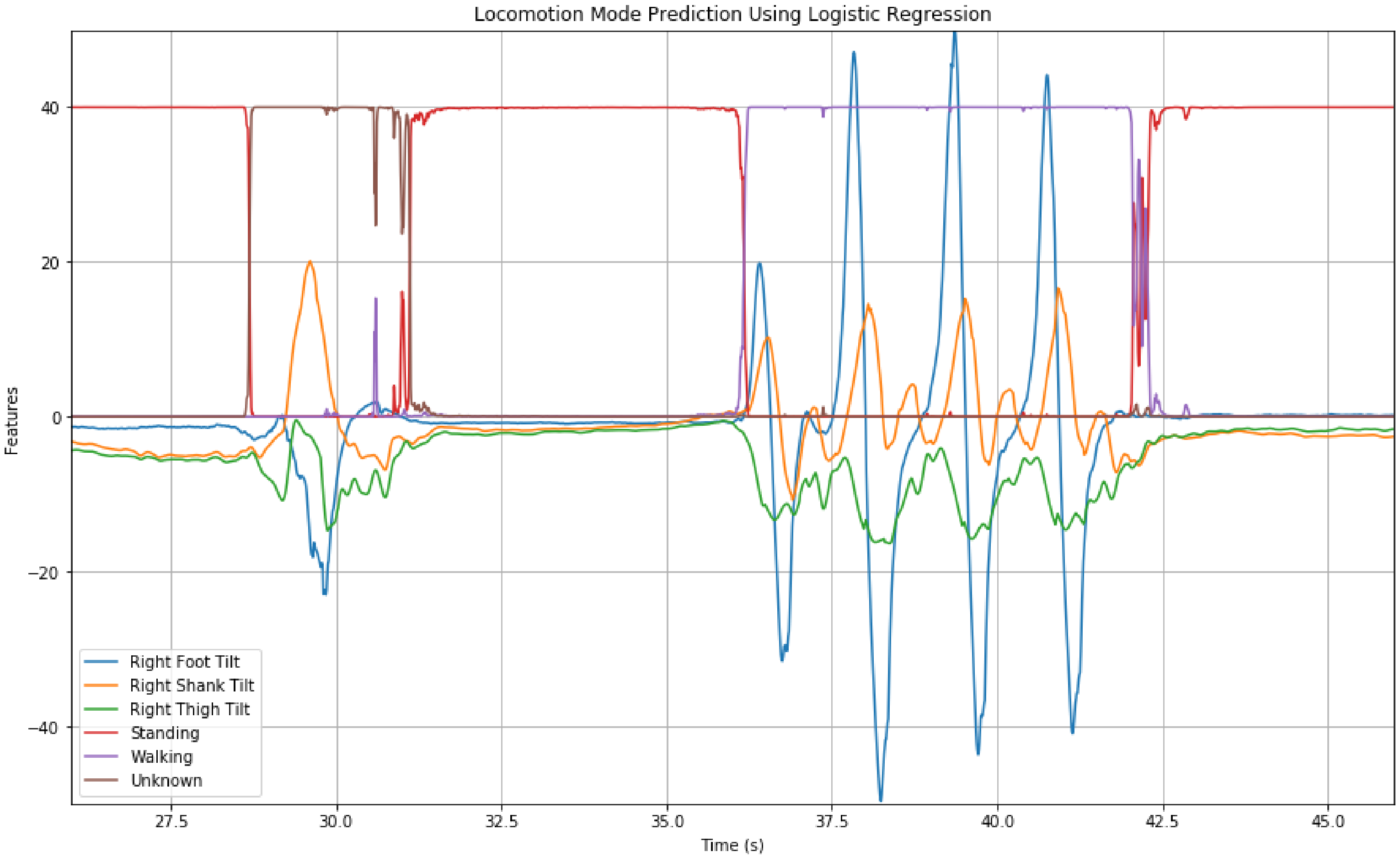
Inertial Sensors

Datalogger

Own Dataset

Modes: Sitting & Standing (~10 mins)
Walking & Standing (~5mins)
Samples: 100 Hz

Logistic Regression (validation_split = 0.33)

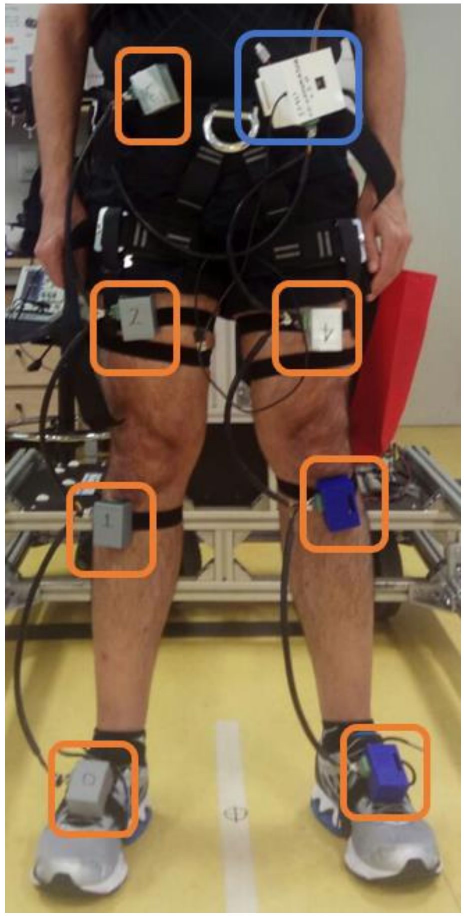


After 30 epochs: train_loss: 0.0564 - train_acc: 0.9798
val_loss: 0.2840 - val_acc: 0.9084

Own Dataset

Modes: Sitting & Standing (~10 mins)
Walking & Standing (~5mins)
Samples: 100 Hz

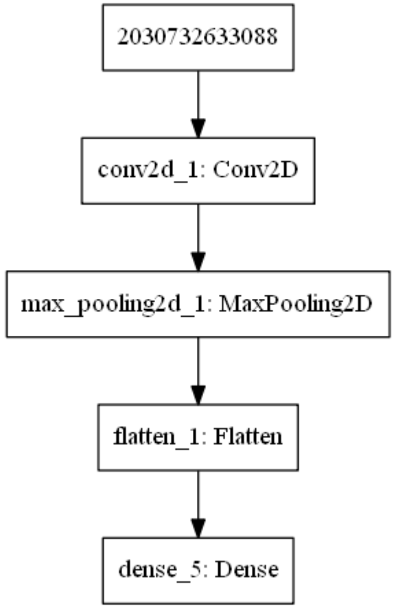
Setup

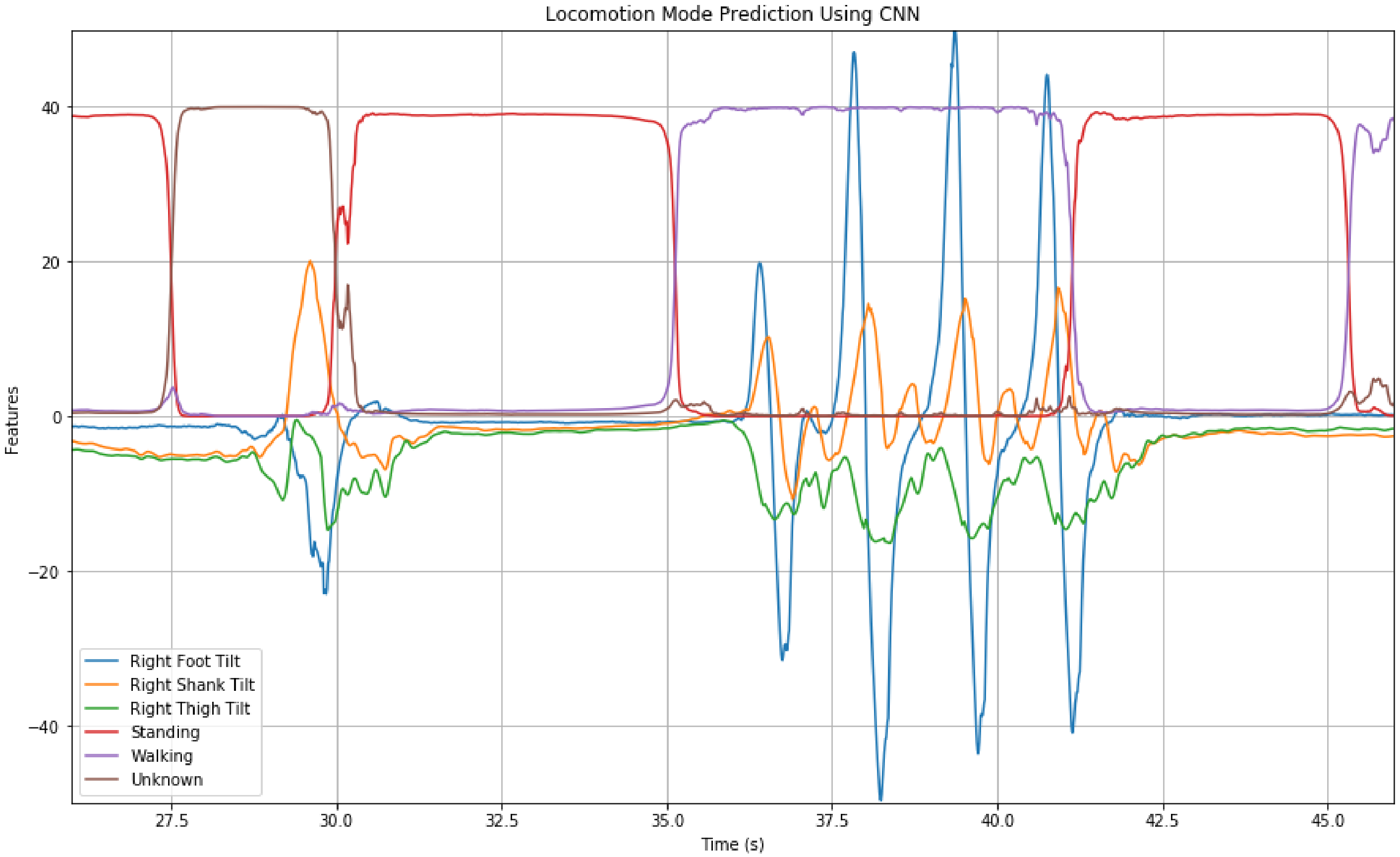


Inertial Sensors

Datalogger

CNNs (validation_split = 0.33, window size: 120)



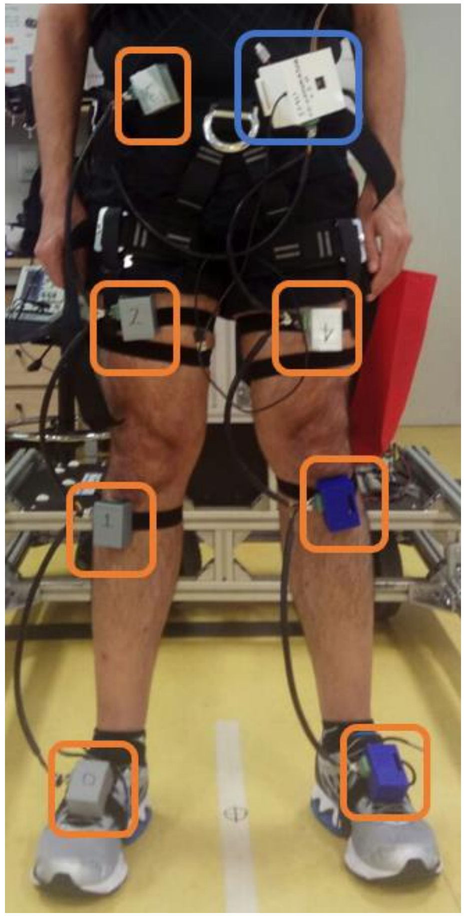


After 1 epoch: train_loss: 0.2219 – train_acc: 0.9356
val_loss: 0.1073 - val_acc: 0.9631

Own Dataset

Modes: Sitting & Standing (~10 mins)
Walking & Standing (~5mins)
Samples: 100 Hz

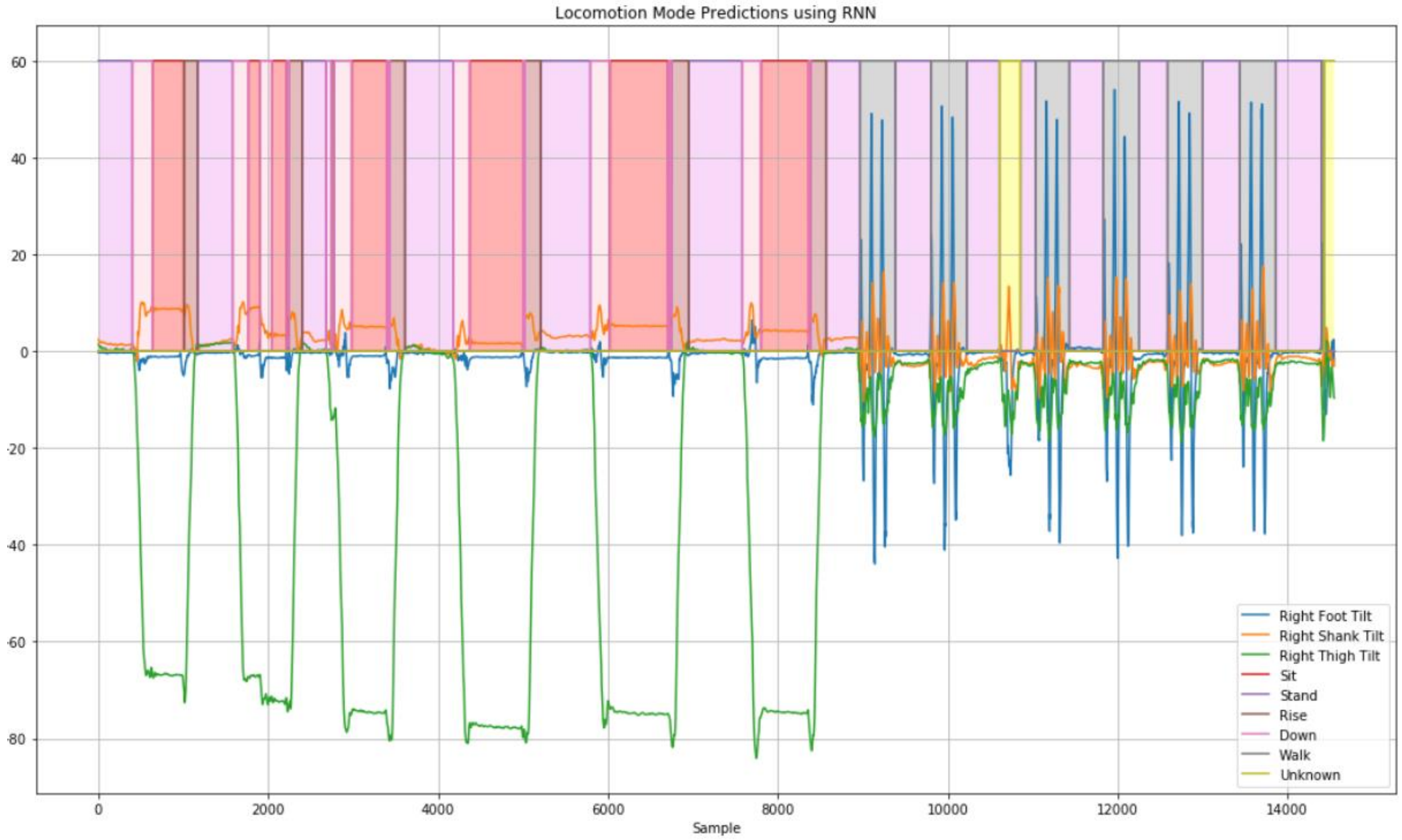
Setup



Inertial Sensors

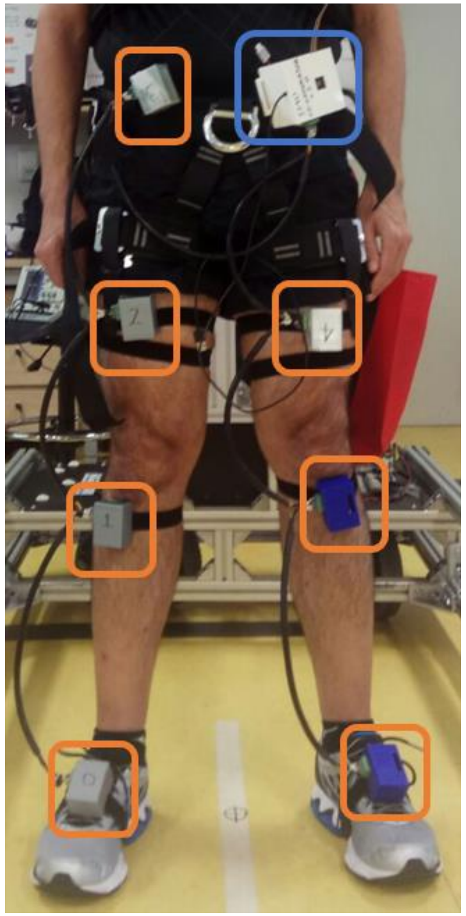
Datalogger

RNNs (validation_split = 0.33, window size: 120)



After 10 epochs: train_loss: 0.2030 – train_acc: 0.9778
val_loss: 0.2817 - val_acc: 0.9558

Setup



Inertial Sensors

Datalogger

Own Dataset

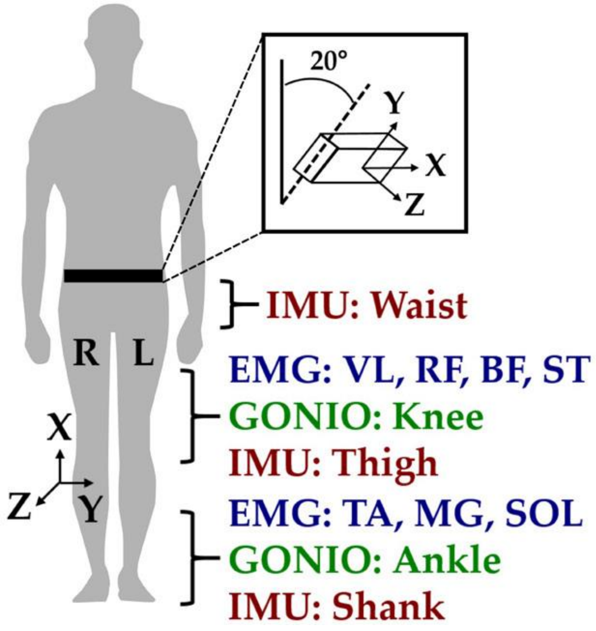
Modes: Sitting & Standing (~10 mins)

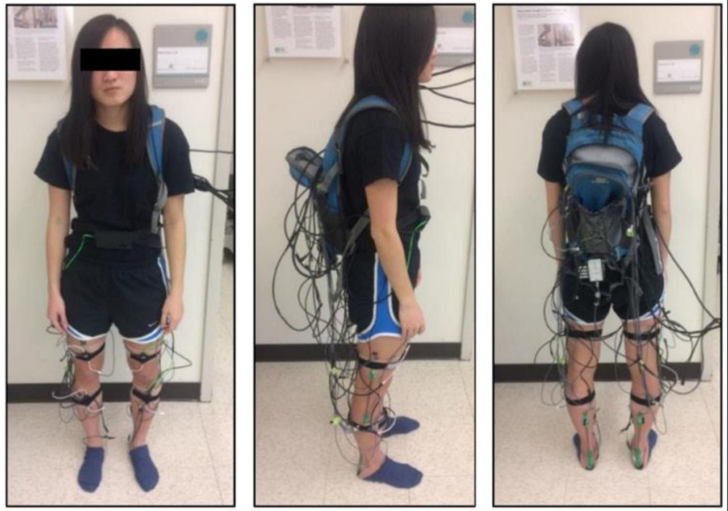
Walking & Standing (~5mins)

Samples: 100 Hz

Model	Epochs	Train_Acc	Val_Acc	Val Split
Logistics Regression	30	0.9787	0.9084	0.33
CNNs	1	0.9356	0.9631	0.33
RNNs	10	0.9778	0.9558	0.33

High accuracy, but it trained and tested on the same dataset.





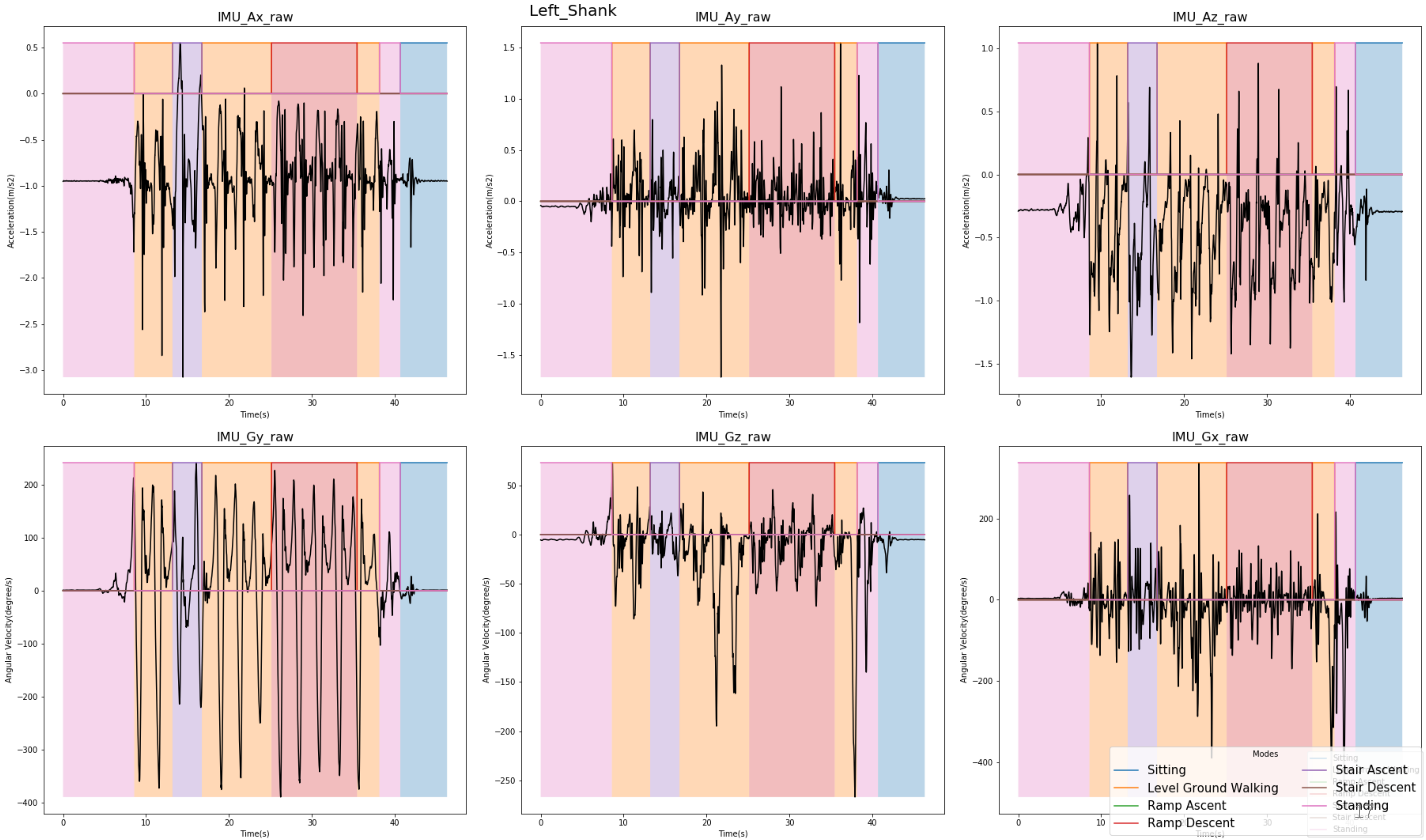
Dataset from Online(2018)

Modes: ['Sitting','Level Ground Walking','Ramp Ascent','Ramp Descent','Stair Ascent','Stair Descent','Standing']

61 Data channels: ['Right_Shank_Ax', 'Right_Shank_Ay', 'Right_Shank_Az', 'Right_Shank_Gy', 'Right_Shank_Gz', 'Right_Shank_Gx', 'Right_Thigh_Ax', 'Right_Thigh_Ay', 'Right_Thigh_Az', 'Right_Thigh_Gy', 'Right_Thigh_Gz', 'Right_Thigh_Gx', 'Left_Shank_Ax', 'Left_Shank_Ay', 'Left_Shank_Az', 'Left_Shank_Gy', 'Left_Shank_Gz', 'Left_Shank_Gx', 'Left_Thigh_Ax', 'Left_Thigh_Ay', 'Left_Thigh_Az', 'Left_Thigh_Gy', 'Left_Thigh_Gz', 'Left_Thigh_Gx', 'Waist_Ax', 'Waist_Ay', 'Waist_Az', 'Waist_Gy', 'Waist_Gz', 'Waist_Gx', 'Right_TA', 'Right_MG', 'Right_SOL', 'Right_BF', 'Right_ST', 'Right_VL', 'Right_RF', 'Left_TA', 'Left_MG', 'Left_SOL', 'Left_BF', 'Left_ST', 'Left_VL', 'Left_RF', 'Right_Ankle', 'Right_Knee', 'Left_Ankle', 'Left_Knee', 'Right_Ankle_Velocity', 'Right_Knee_Velocity', 'Left_Ankle_Velocity', 'Left_Knee_Velocity', 'Mode', 'Right_Heel_Contact', 'Right_Heel_Contact_Trigger', 'Right_Toe_Off', 'Right_Toe_Off_Trigger', 'Left_Heel_Contact', 'Left_Heel_Contact_Trigger', 'Left_Toe_Off', 'Left_Toe_Off_Trigger']

Implementation Seen in Jupyter Notebook

“Frontiers | Benchmark **Datasets** for Bilateral Lower-Limb Neuromechanical Signals from Wearable Sensors during Unassisted Locomotion in Able-Bodied Individuals | Robotics and AI,” **2018**.


Left_Shank
IMU_Ax_raw
IMU_Ay_raw
IMU_Az_raw
IMU_Gy_raw
IMU_Gz_raw
IMU_Gx_raw
Acceleration(m/s2)
Angular Velocity(degree/s)
Time(s)
Modes
Sitting
Level Ground Walking
Ramp Ascent
Ramp Descent
Stair Ascent
Stair Descent
Standing

Own Dataset

Modes: Sitting & Standing (~10 mins)
Walking & Standing (~5mins)
Samples: 100 Hz

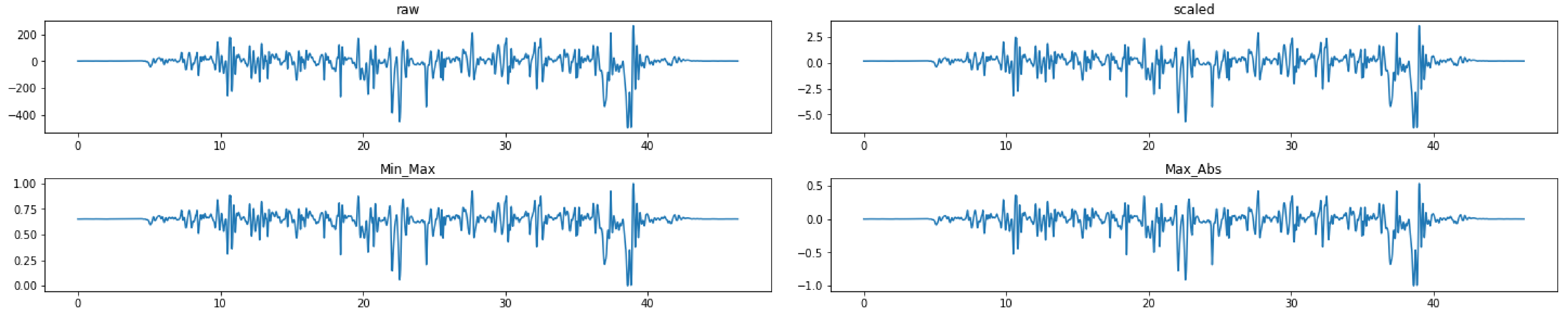
Dataset from Online

Modes: ['Sitting','Level Ground Walking','Ramp Ascent','Ramp Descent','Stair Ascent','Stair Descent','Standing']

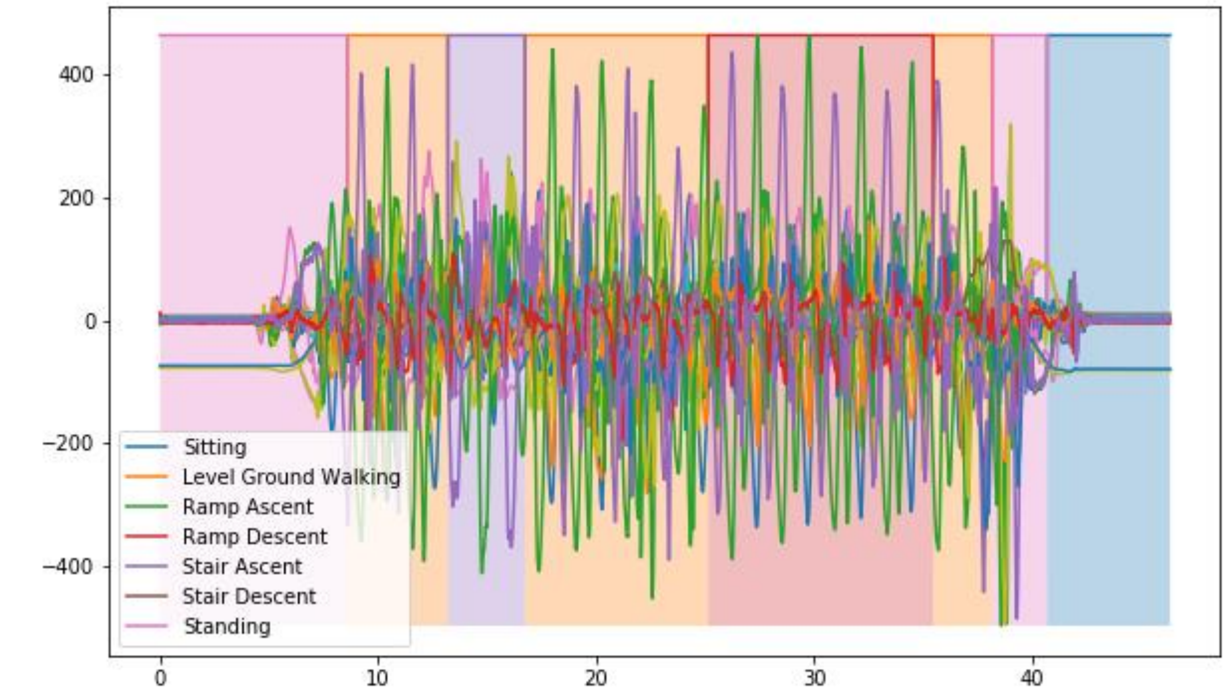
Model		Epochs	Train_Acc	Val_Acc	Val Split	data
Logistics Regression	relu+adam	30	0.9787	0.9084	0.33	Own(15mins)
CNNs		1	0.9356	**0.9631**	0.33	
RNNs		10	0.9778	0.9558	0.33	
Logistics Regression	relu+adam	30	1	0.3873	0.33	Online(**45s**)
	relu+rmsprop		0.9997	0.3868	0.33	
	sigmoid+adam		1	0.3909	0.33	
CNNs (Train on trail'0')	Window size:100(0.2s) Train step size:20	10	0.8279	0.367	Val on '1'	
				0.641	Val on '2'	
				0.395	Val on '3'	

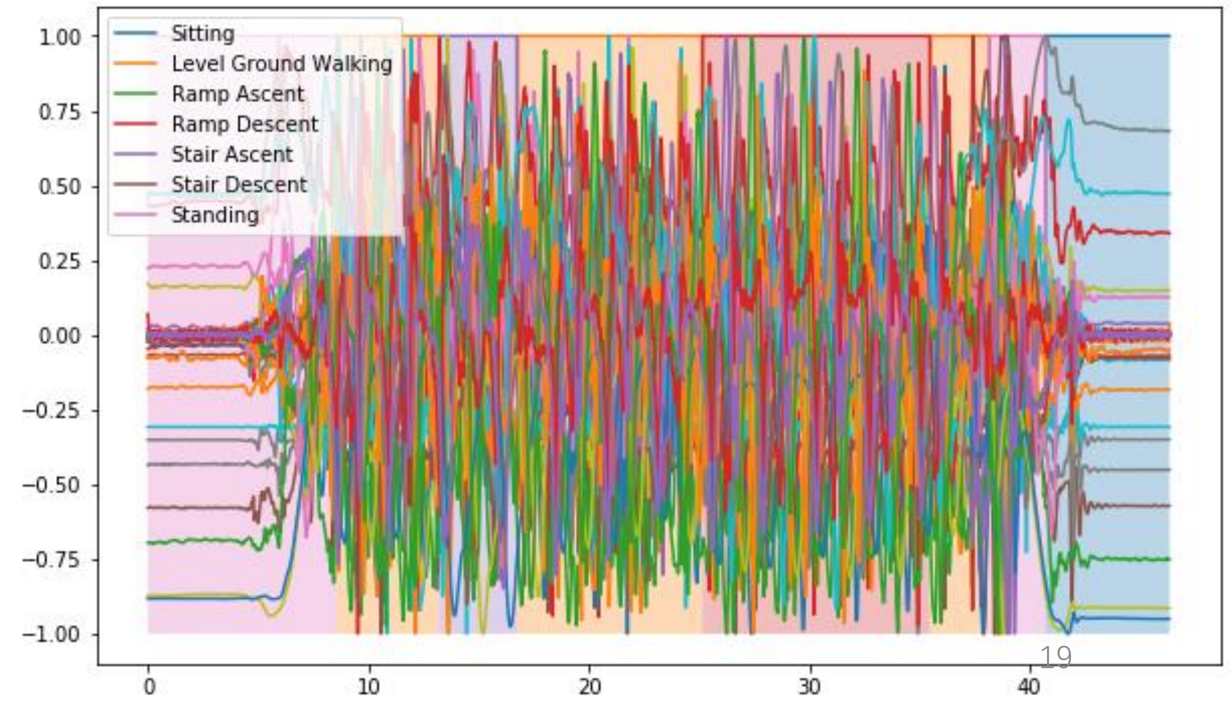
"Frontiers | Benchmark **Datasets** for Bilateral Lower-Limb Neuromechanical Signals from Wearable Sensors during Unassisted Locomotion in Able-Bodied Individuals | Robotics and AI," **2018**.

Preprocess methods: Minmax([0,1]), scall(Mean=0, std=1), Max_Abs([-1,1])…



MaxAbs: keep nagetive features and scale in 1 of absolute value.

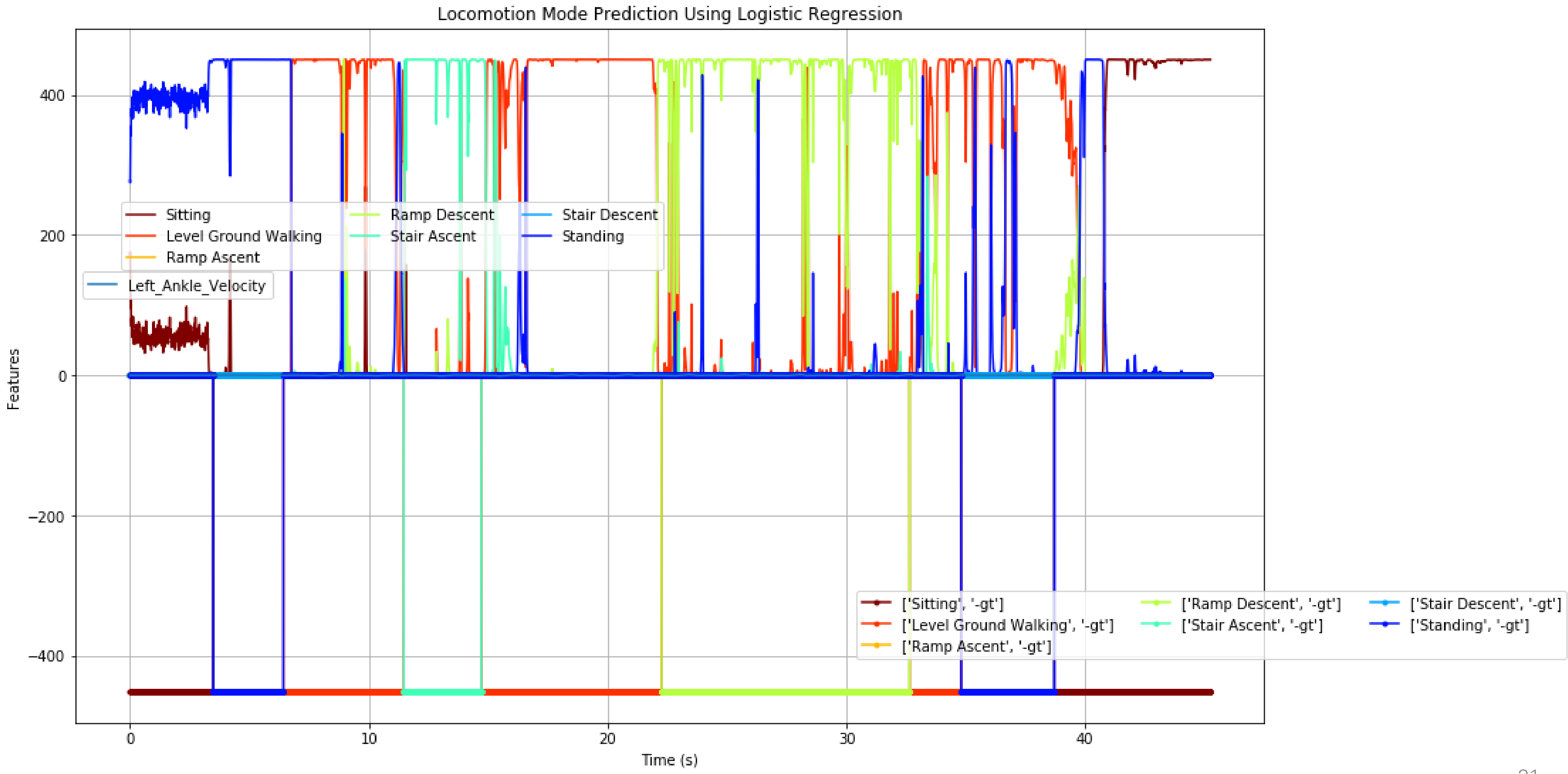




Online dataset train results

Online(45s)								
			Raw			**Max_Abs**		
Model(Train on trail'0')		Epochs	Train_Acc	Val_Acc	Val Split	Train_Acc	Val_Acc	Val Split
Logistics Regression	relu+adam	30	1	0.3873	0.33	1	**0.749**	'2'
	relu+rmsprop		0.9997	0.3868	0.33	1	**0.749**	
	sigmoid+adam		1	0.3909	0.33	0.997	**0.749**	
CNNs	Window size:100(0.2s) Train step size:20	10	0.8279	0.367	Val on '1'	**0.911**	**0.379**	Val on '1'
				0.641	Val on '2'	**0.971**	**0.728**	Val on '2'
				0.395	Val on '3'	**0.909**	0.363	Val on '3'

Online dataset train results

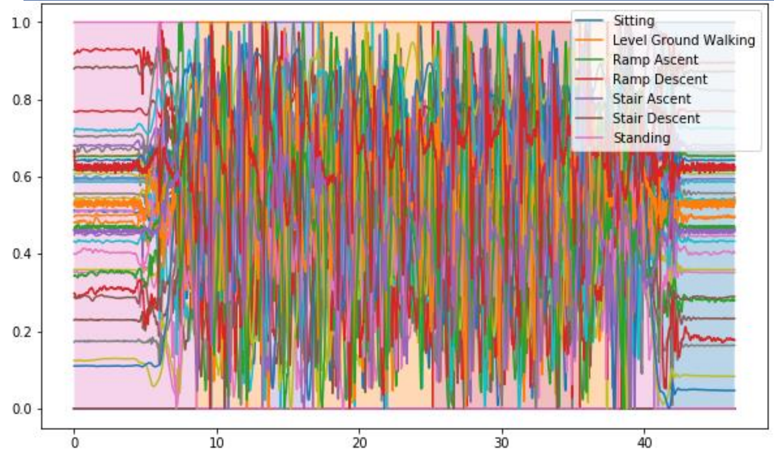


Online data – RNNs(LSTM)

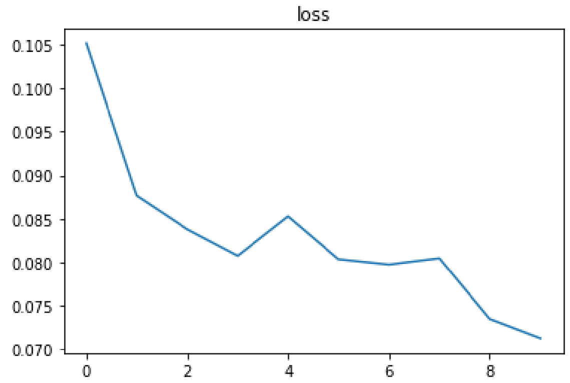
```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_3 (LSTM)                (None, None, 24)          6048
_________________________________________________________________
lstm_4 (LSTM)                (None, 12)                1776
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 91
=================================================================
Total params: 7,915
Trainable params: 7,915
Non-trainable params: 0
```

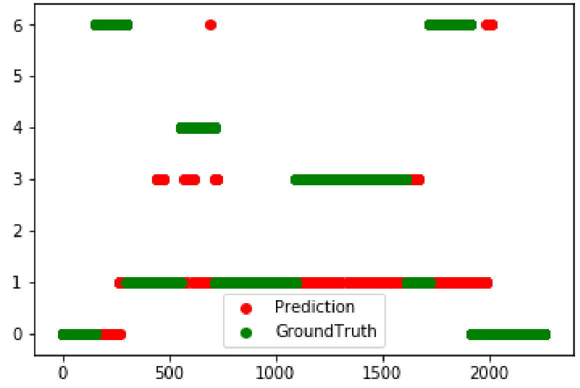
Online(45s)			
	Epochs	Window size	Step size
RNNs (LSTM)	10	200	10
	10	400	10
	100	50	5

MinMax

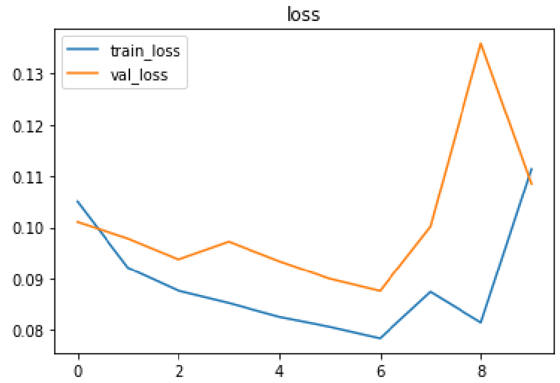


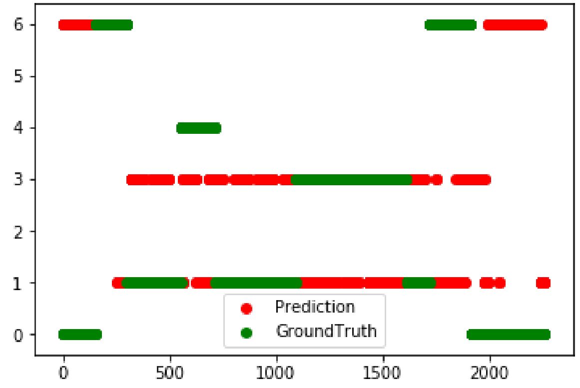
Win_Size-200, Stp_Size-10 (epoch:10)



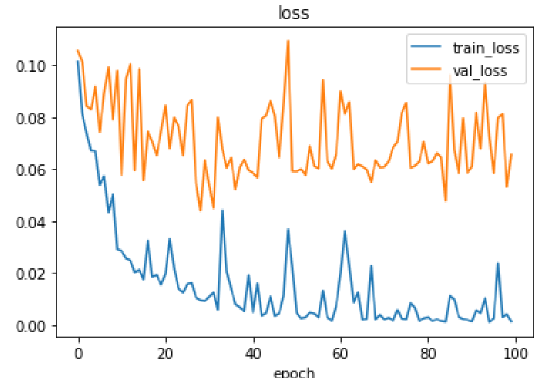


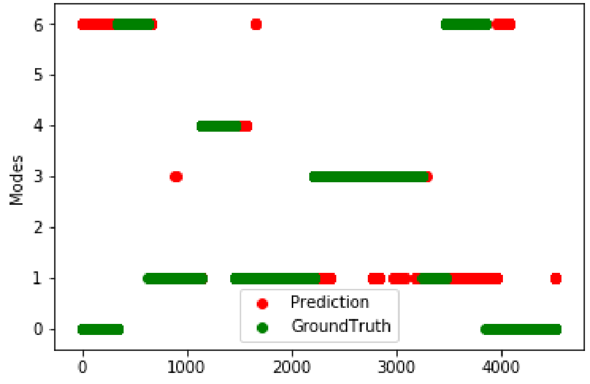
Win_Size-400, Stp_Size-10 (epoch:10)



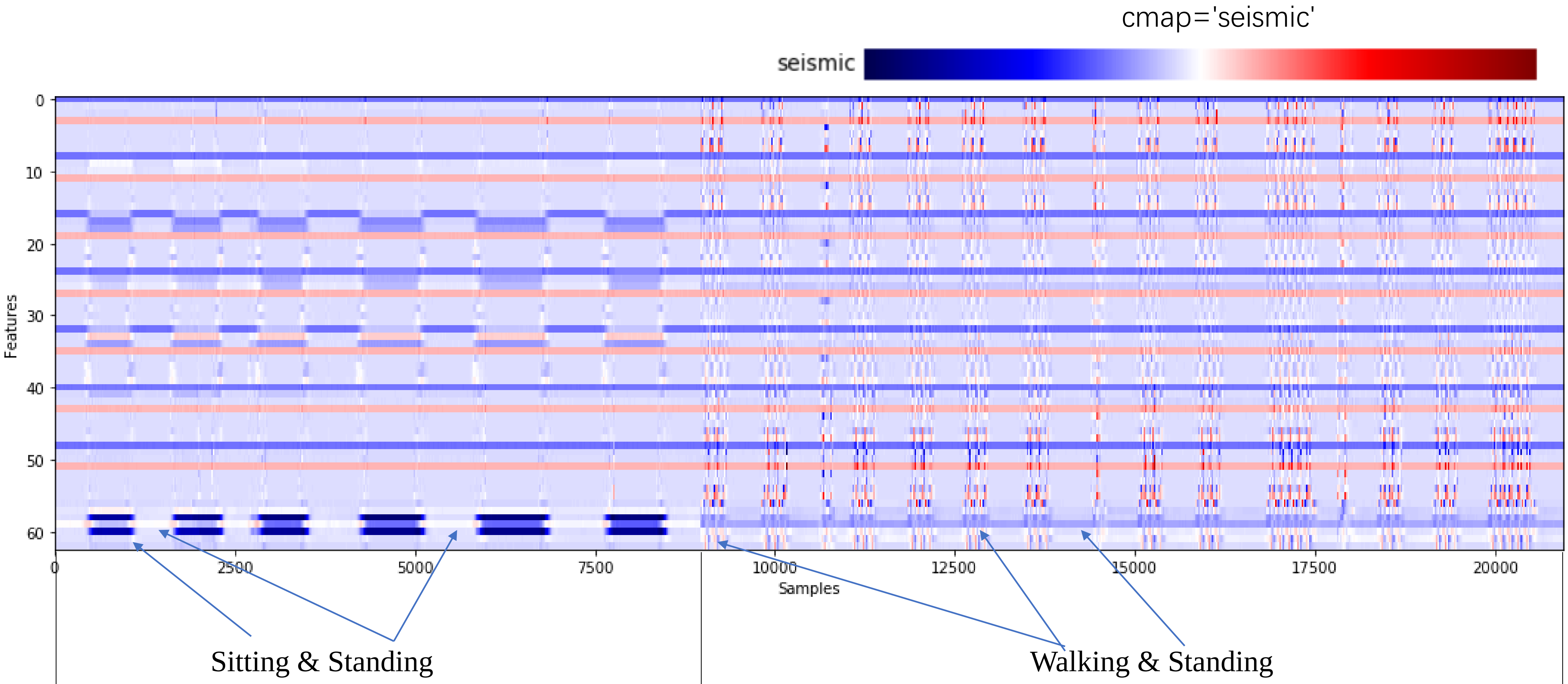


Win_Size-50, Stp_Size-50 (epoch:100)





Data decoding/pre-process IMU&Gonio data - Image



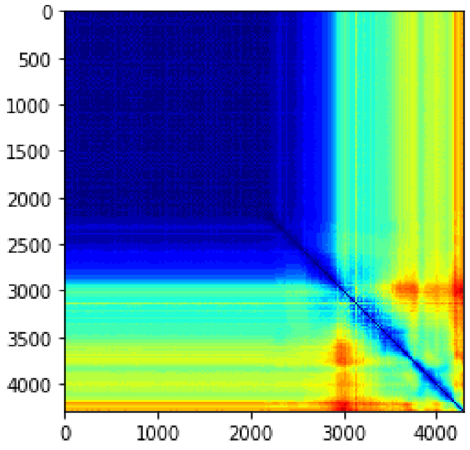
Easily be distinguished by eyes, so does deep learning.

Data decoding/pre-process IMU&Gonio data - Image

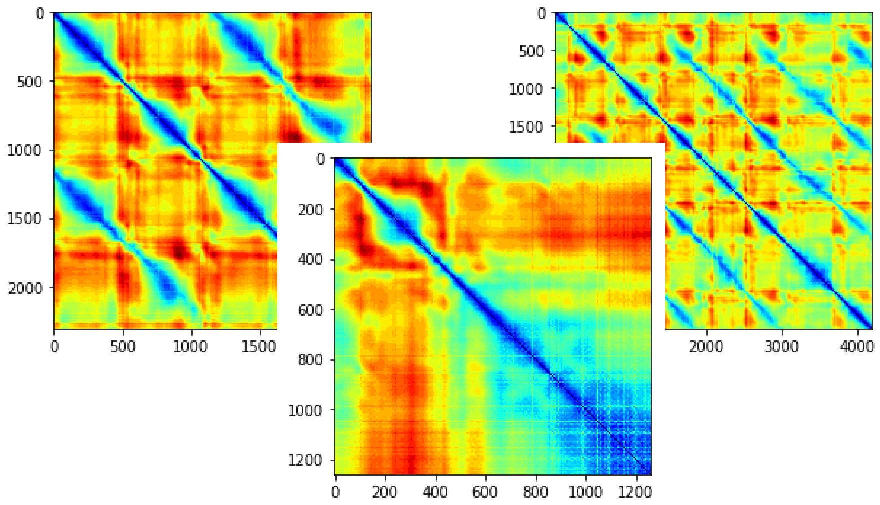
Recurrence Plots

$$RP_{i,j} = \theta(\epsilon - ||q(i) - q(j)||)$$

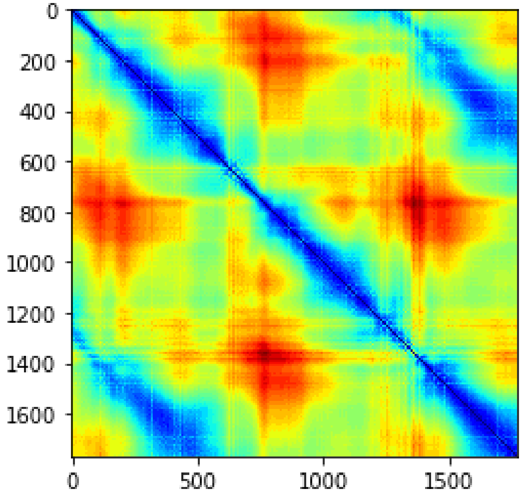
Standing phase



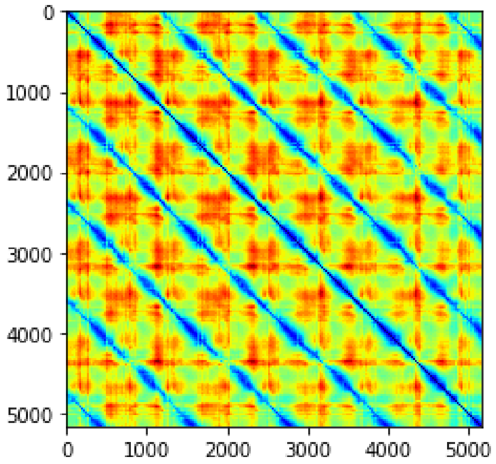
Level ground walking phase



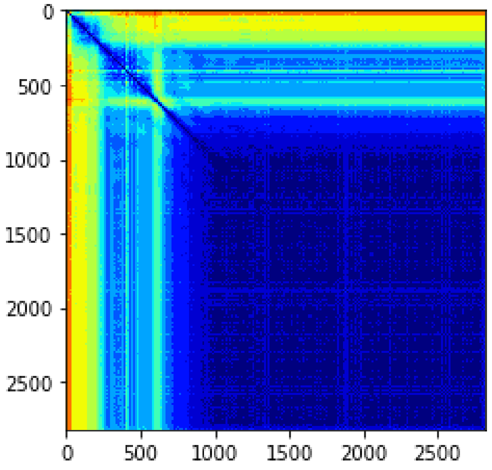
Stair ascent



Ramp descent phase



Sitting phase



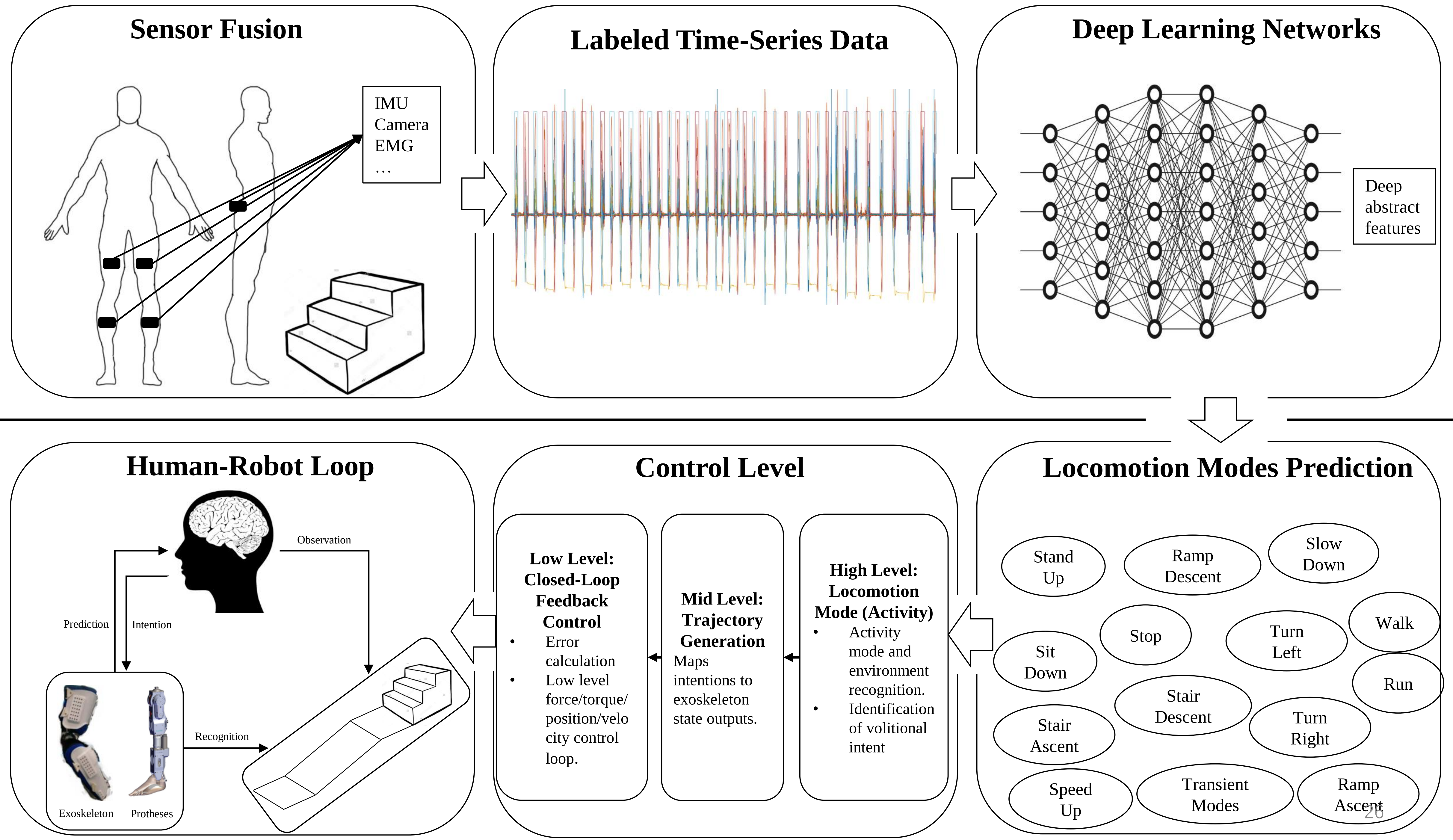
As shown the decoded images of each modes in one trial, we can tell that different modes could be decoded into **a easily distinguishable image**. Then, deep CNNs is capable of discovering deep features in the images and **classify** them. So far, there is only one decode method that has been implemented, but it looks effective. Therefore, **next step** will need to do

1. figure out more decode methods

2. decode all trails of data and input to CNNs, like AlexNet, VGGnet… (Transfer learning)

3. implement RNNs as time-series data being input.(Done)

4. stack all online data and train in CNNs, RNNs, …


Sensor Fusion
IMU
Camera
EMG
…
Labeled Time-Series Data
Deep Learning Networks
Deep abstract features
Locomotion Modes Prediction
Stand Up
Ramp Descent
Slow Down
Sit Down
Stop
Turn Left
Walk
Run
Stair Ascent
Stair Descent
Turn Right
Speed Up
Transient Modes
Ramp Ascent
Control Level
Low Level: Closed-Loop Feedback Control
• Error calculation
• Low level force/torque/position/velocity control loop.
Mid Level: Trajectory Generation
Maps intentions to exoskeleton state outputs.
High Level: Locomotion Mode (Activity)
• Activity mode and environment recognition.
• Identification of volitional intent
Human-Robot Loop
Observation
Prediction
Intention
Recognition
Exoskeleton
Protheses

TimeLine(Plan)

Date	To do	Note
10/08-10/19	Online dataset inspection	Focus on the pre-processing and how many kinds of deep learning networks have been used.
	Paper review	
10/24-11/3	Building deep learning networks	The purchase of computer might be completed then, if not, just training on partial of online data.
	Training model by online dataset	
	Evaluate by our own dataset	
11/05-11/18	Implement the deep learning model on the portable microcomputer	Convert python to C/C++ could be a tough and time-consuming work. Care about the system latency.
	Try real-time locomotion prediction on able subject.	
11/19-12/02	Strategies/Algorithms of utilizing the locomotion prediction model as the high-level control of current exoskeleton	The mid-level and low-level have been completed. Focusing on how to fully work on the classification result.
	Do the experiment controlling exoskeleton to assist human with real-time locomotion prediction	
12/03-12/16	Writing a report/conference paper	Keep touch in the paper submit deadline
	Review