Machine Learning for Human Movement Classification Based on Kinect Skeleton Data

SAPIENZA
UNIVERSITÀ DI ROMA

Candidate Lucian Dorin Crainic
Supervisor Prof. Maurizio Mancini
A.Y. 23/24

What is Kinect Skeleton Data?

MICROSOFT KINECT SENSOR

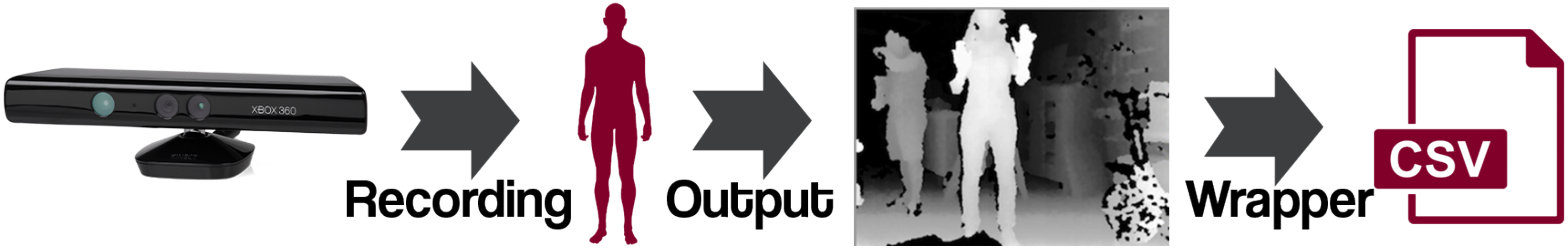
Motion sensing input device developed by Microsoft

- 3 types of data: color images, 3D depth images, bone information

- PyKinect2 library abstracts complex functionalities of the sensor like Skeletal tracking

KINECT SENSOR TO SKELETON DATA

Data collection process

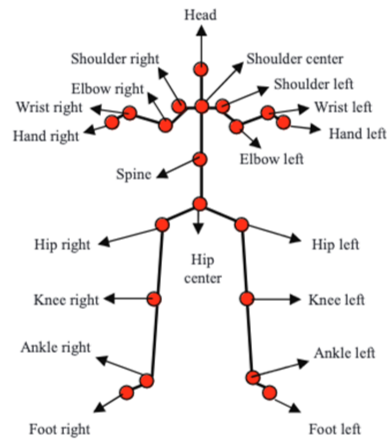


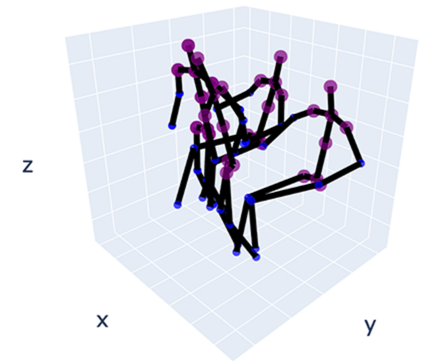
- Patient performs movement in front of the Kinect
- Position and orientation data in the 3D axis is provided by the sensor Skeletal tracking for a set of joints
- Wrapper gets the data and stores it as a multi-dimensional time series in a CSV file

DATA VISUALIZATION

From CSV Data to a 3D plot animation

C	D	E	F	G
Head.state	Head.px	Head.py	Head.pz	Head.rx
2.00	2371.34.00	9561.15.00	11735.38.00	0.00
2.00	2783.25.00	9490.54.00	11698.14.00	0.00
2.00	2958.20.00	9435.11.00	11708.02.00	0.00
2.00	3033.29.00	9422.21.00	11713.33.00	0.00
2.00	3144.00.00	9419.49.00	11742.47.00	0.00
2.00	3249.39.00	9413.10.00	11751.10.00	0.00
2.00	3339.20.00	9410.33.00	11758.34.00	0.00
2.00	3415.19.00	9406.38.00	11768.52.00	0.00
2.00	3483.13.00	9407.19.00	11760.25.00	0.00
2.00	3529.38.00	9401.21.00	11753.20.00	0.00



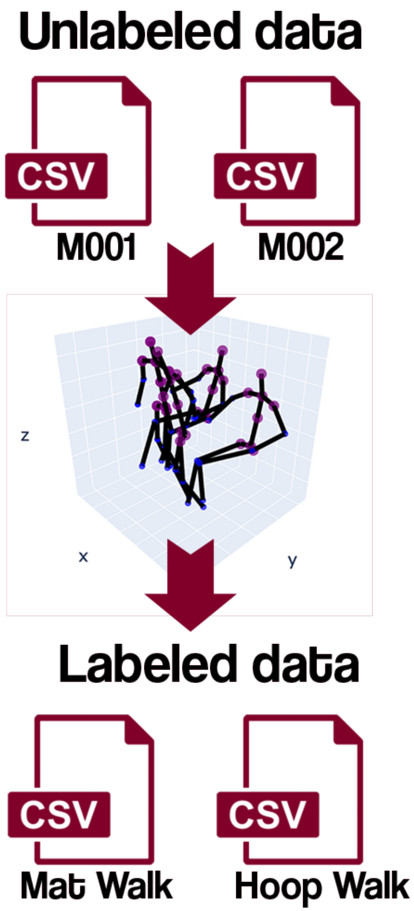


- Every row in the csv file represents a frame in time of the skeleton joints
- Use joints data to create connections between them to build a skeleton animation using Plotly library

MOVEMENTS ANALYSIS

Movements used in the classification task

Movements	
Reach overhead	Chair to Chair
Cross-Reach Left	Cross-Reach Right
Reach Forward	Hoop Walk
Right Leg Stand	Left Leg Stand
Mat Walk	Tug Walk



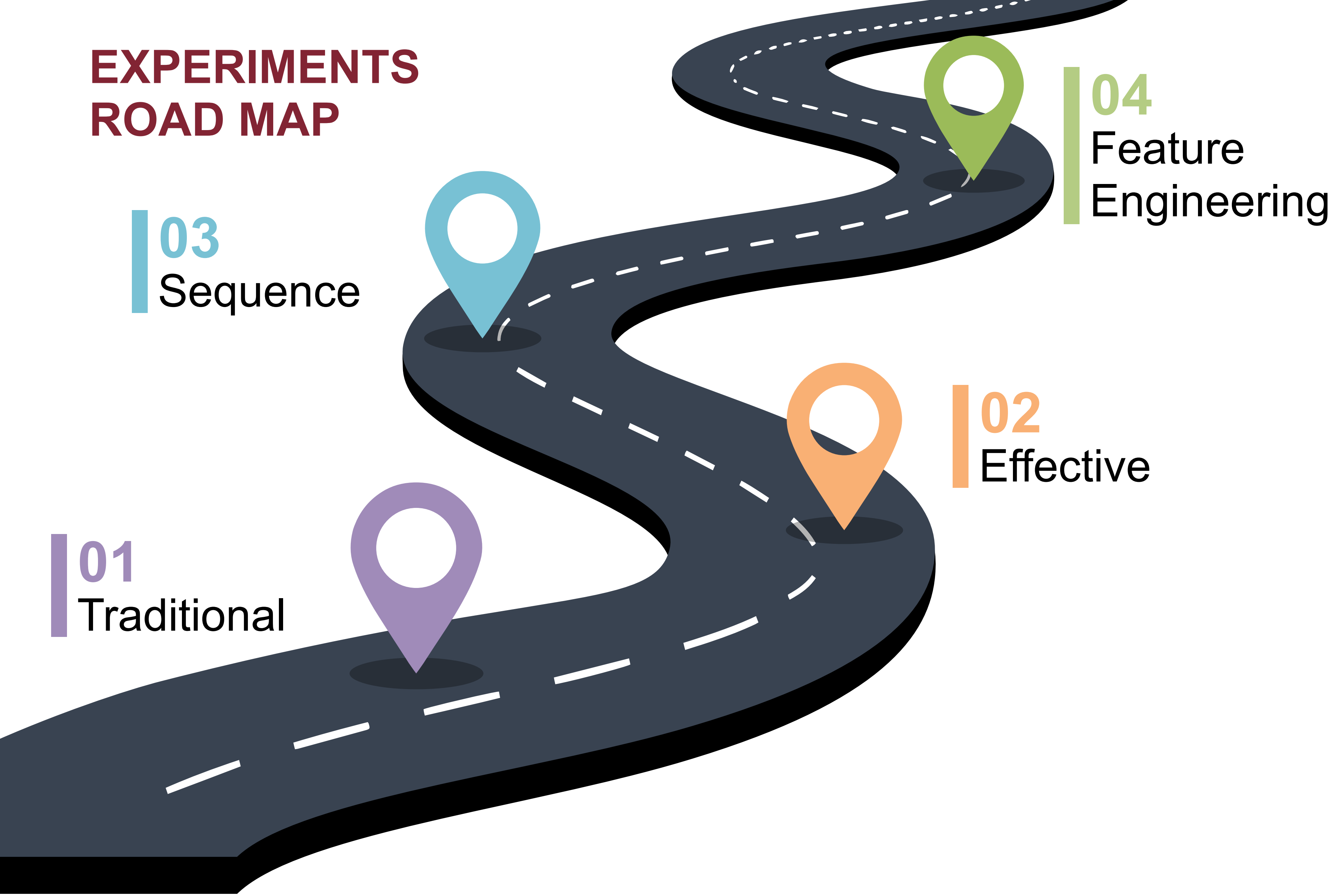
MACHINE LEARNING MODELS

Used in the experimental phase of this thesis

Models	
Support-Vector Machines	Gaussian Naive Bayes
Random Forests	Gradient Boosting
Logistic Regression	Linear-Discriminant Analysis
Multi-Layer Perceptron	K-Nearest Neighbors
Ada Boost	Decision Trees

- **10** Machine Learning models selected
- Implemented using **Scikit-Learn** library
- Selection based on *popularity* and *performance*

EXPERIMENTS ROAD MAP



01 TRADITIONAL APPROACH

Literature approach for effective training and testing split

70% Training 30% Testing

Random Forests

Overfitting ?

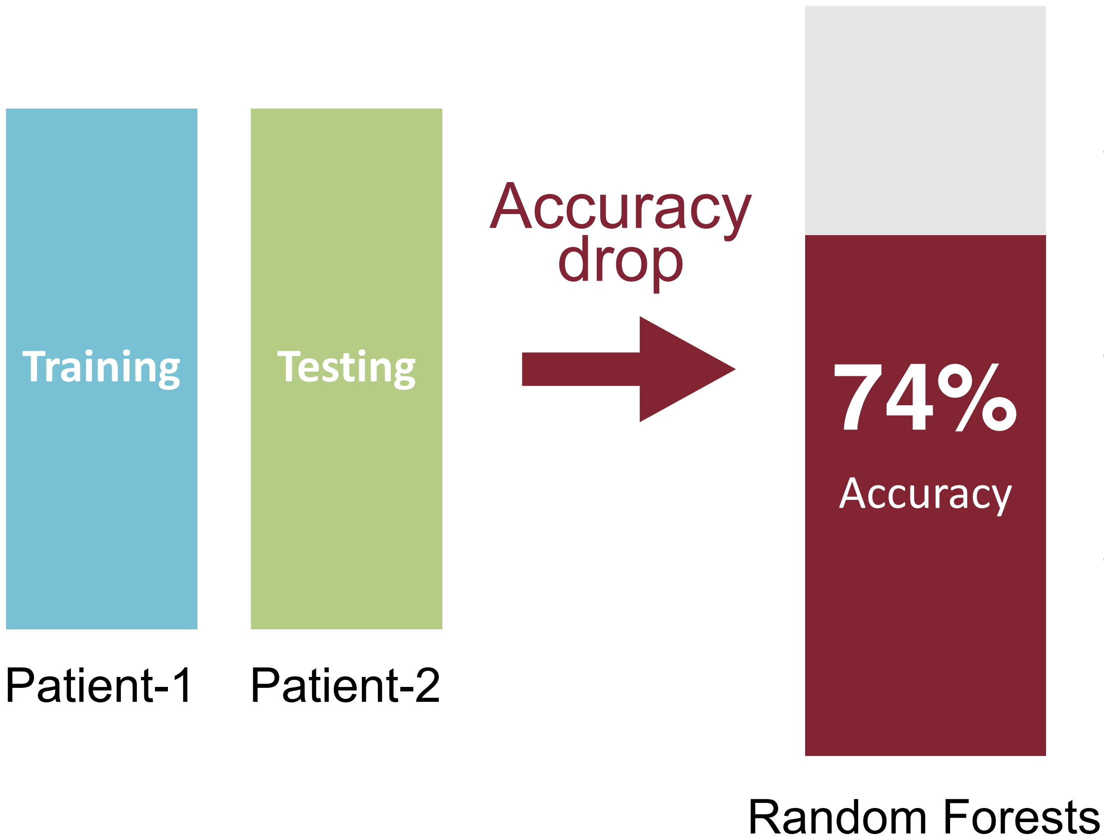
YES!

Patient-1

- Data is split without any *costraints*
- Patient's data in **both** training sets !
- **Patient ID** needs to be considered when splitting the data

02 EFFECTIVE APPROACH

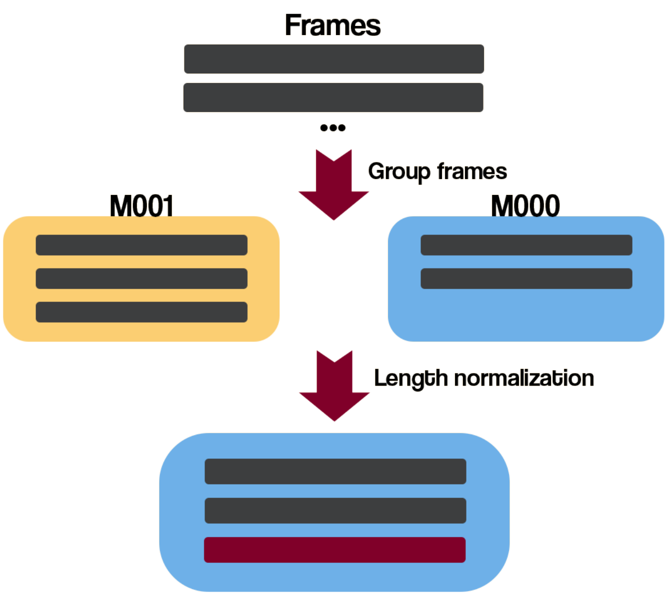
Data splitting with Patient ID set as a costraint



- Patient is present in **one set** only
- Models do not overfit anymore
- How can we help the models to *differentiate* better ?

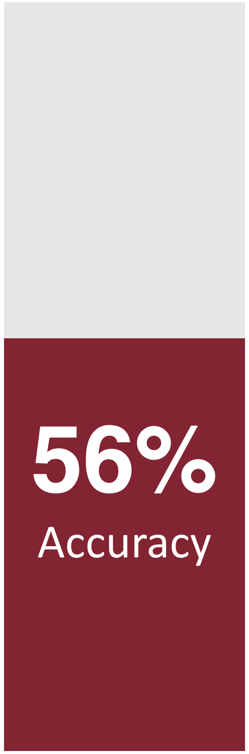
03 SEQUENCE APPROACH

Frames grouped based on movement type



PROBLEMS

- Different lengths
- Normalize lengths using feature aggregation
- Loss of context



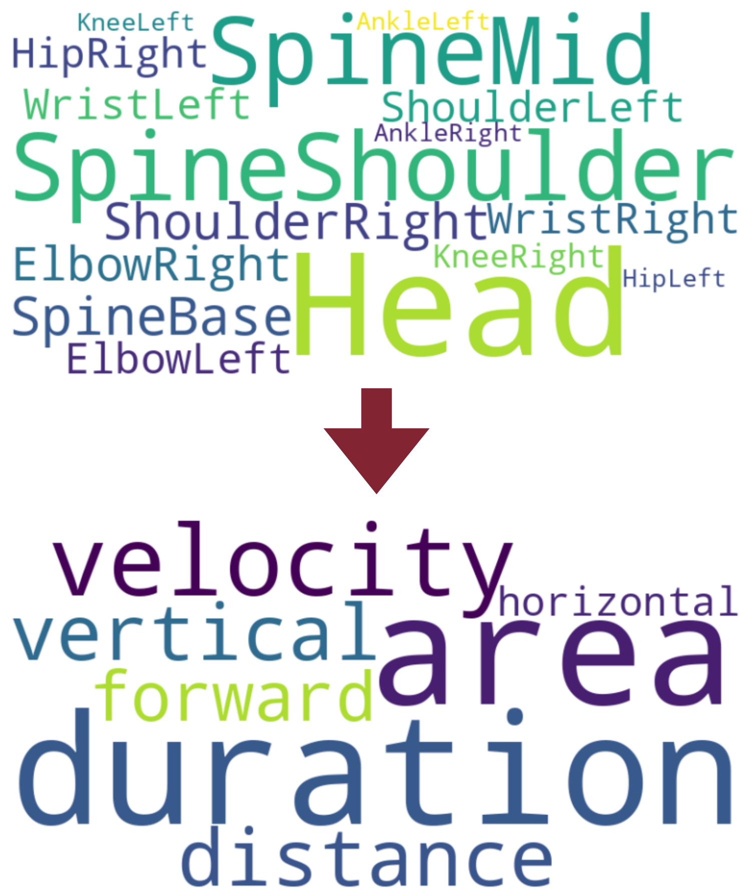
KNNs

04 FEATURE ENGINEERING APPROACH

Transform Kinect Skeleton data to obtain better results

- Calculate *new features* from skeleton data to obtain more informative features

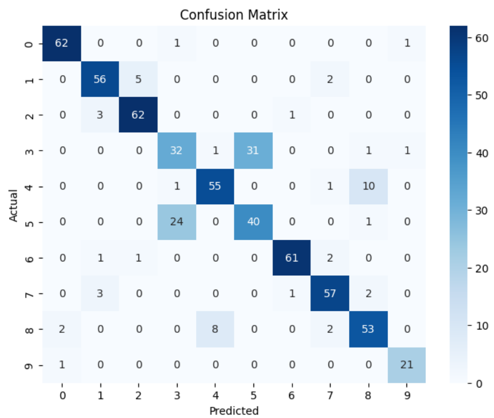
- Reduced features number from **120** to **14**



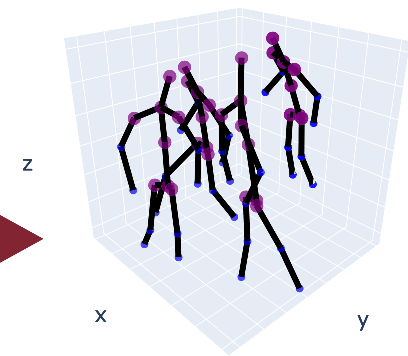
Multi-Layer Perceptron

MOVEMENTS SIMILARITY

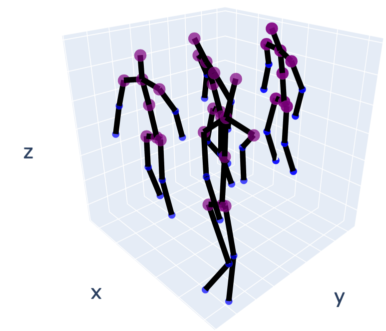
Models struggle to differentiate between two movements



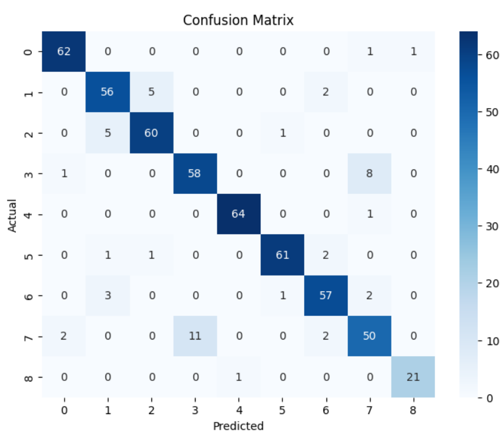
Movements similarity



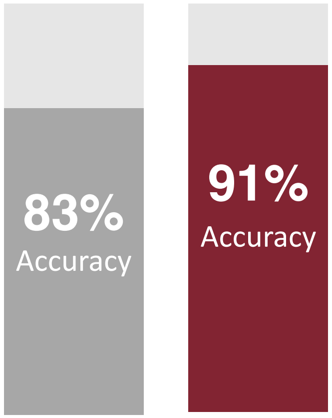
Hoop Walk



Mat Walk



Movement removed



Multi-Layer Perceptron

CONCLUSIONS

Key findings of this thesis

1 Kinect skeleton data

Suitable for a classification task, but it alone does not provide enough information

2 Feature Engineering

Crucial process for increasing classification *accuracy* by creating more informative features

3 Multi-Layer Perceptron

Best performing model with an accuracy score of **0.83**

4 3D visualization

Visually identify movements and label them

FUTURE WORK

Potential new methods to implement

CLASSIFICATION

Explore the possibility of **real-time** prediction

DATA

Collect more data with new movements

MODELS

Implement complex models like *Neural Networks*

Thank you for your attention.