04/07/2020

TRAINIT:

DETECTION AND CLASSIFICATION OF WRONG PLAYED STROKES IN TABLE TENNIS.

In collaboration with Al Ahly Club and Egyptian Table Tennis Federation.

GROUP MEMBERS:

HABIBA HEGAZY, MOHAMED ABDELSALAM, MOUSTAFA HUSSIEN AND SEIF ELMOSALAMY

SUPERVISED BY:

DR. AYMAN NABIL, DR. AYMAN EZZAT

TEACHING ASSISTANT:

YOUMNA IBRAHIM

Introduction(1/2)

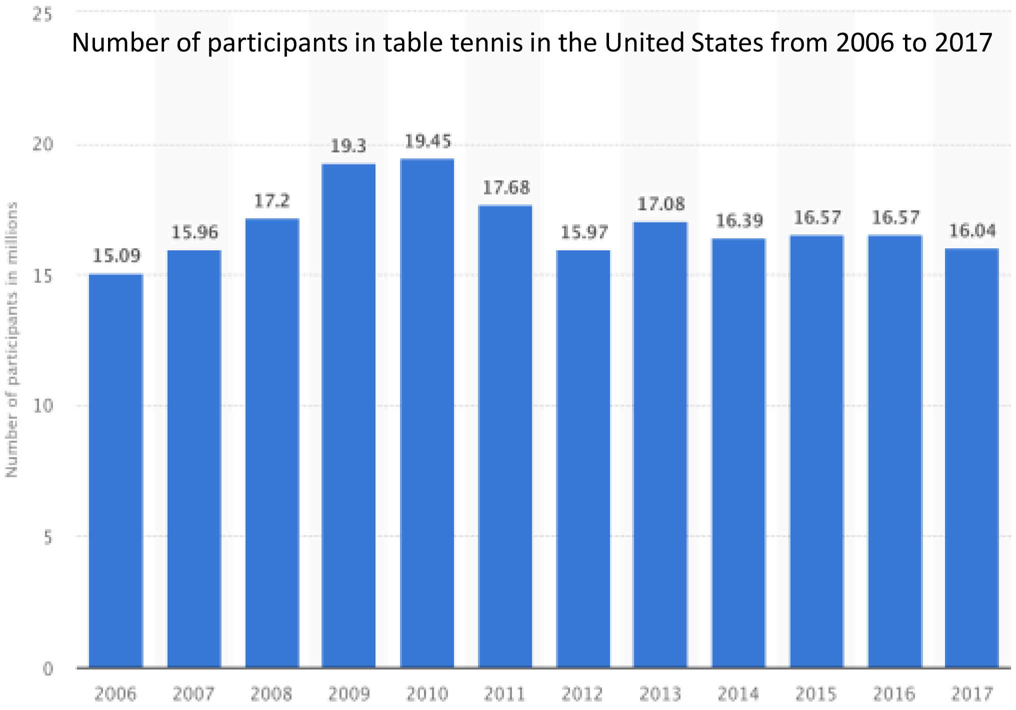
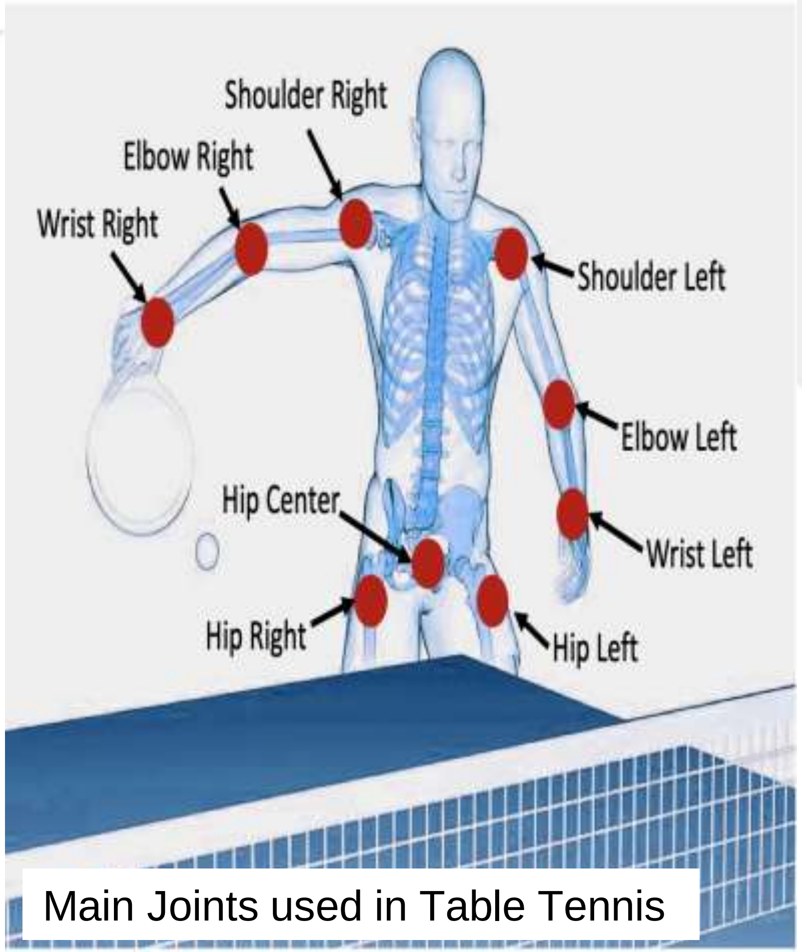


Table Tennis became popular to reach 16 million players.


Backhand


Forehand



Main Joints used in Table Tennis

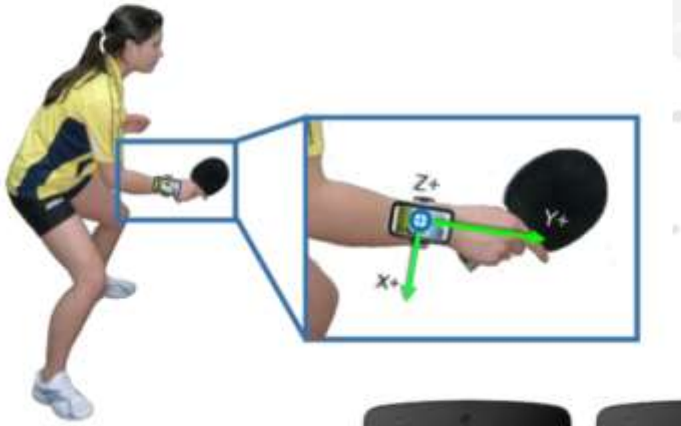
Introduction(2/2) - Common Mistakes

Stroke started with extended elbow

Wrong waist movement

Related Work (1/2): *Average accuracy stroke detection and classification*

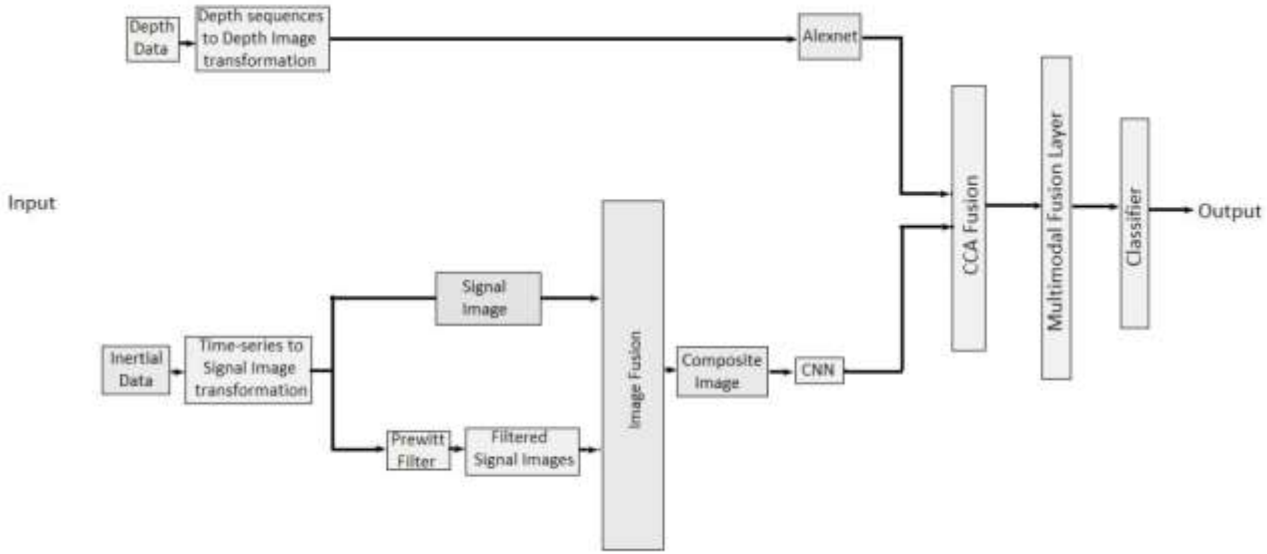
- **Device used:** mobile device.
- The system detects and classifies tennis strokes: **forehand and backhand**.
- **Algorithm used:** Decision Tree
- Average accuracy 69.63% and 77.21%
- Detection the wrist movement.
- Online Feedback.



Waraporn Viyanon, Vimvipa Kosasaeng, Sittichai Chatchawal, and Abhirat Komonpetch. 2016. SwingPong: analysis and suggestion based on motion data from mobile sensors for table tennis strokes using decision tree. In *Proceedings of the 2016 International Conference on Intelligent Information Processing* (**ACM**).

Related Work (2/2): *Sensor fusion*

- Aim: Human Action Recognition.
- **Sensors used:** Kinect and Internal sensors.
- Dataset: Berkeley MHAD.
- **Fusion techniques:** Image fusion, CCA fusion, Multi-model layer fusion.
- **Algorithm used:** CNN and SVM.
- The technique increased the classification accuracy to 98%

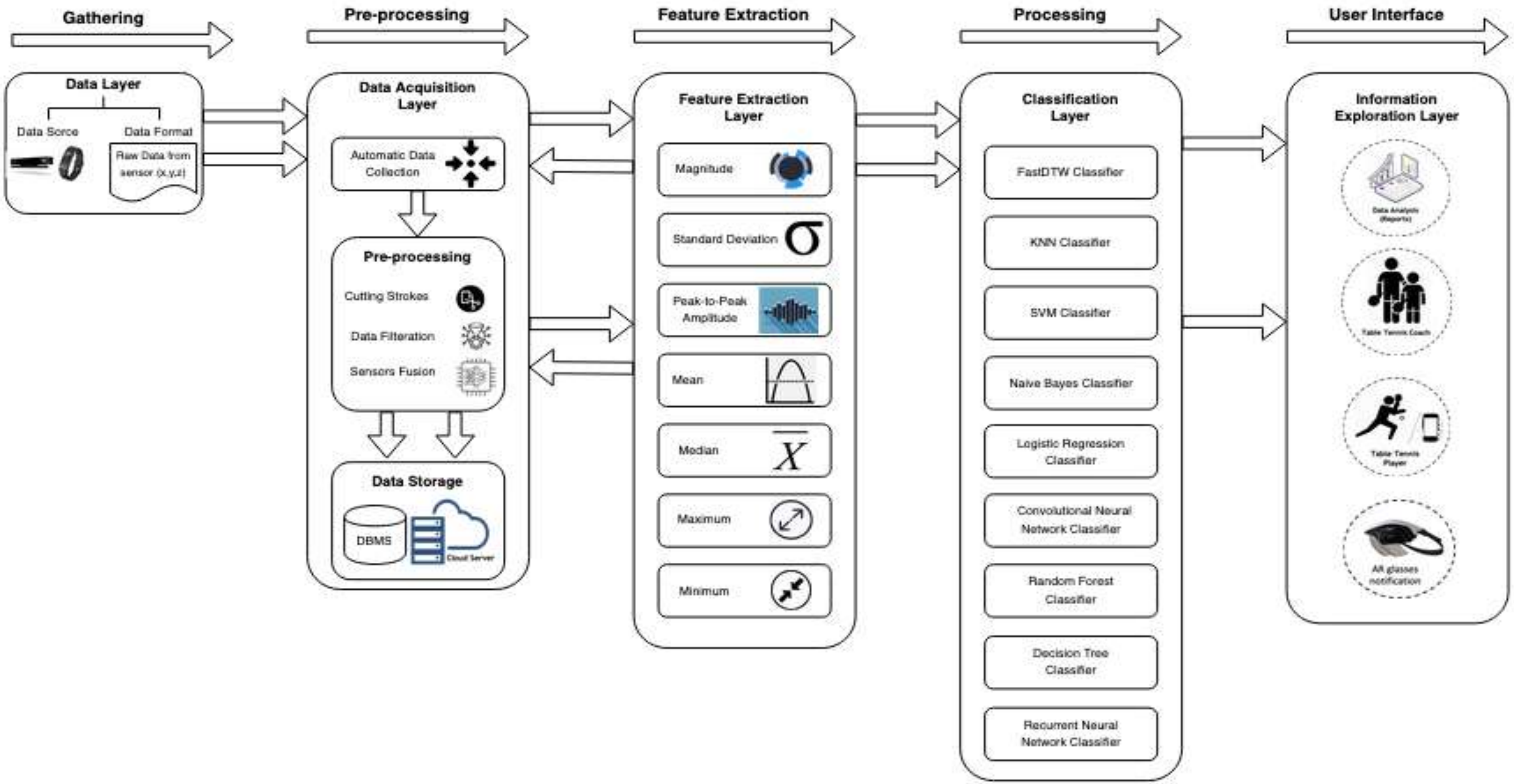


Ahmad, Zeeshan & Khan, Naimul. (2019). Human Action Recognition Using Deep Multilevel Multimodal (M2) Fusion of Depth and Inertial Sensors. **IEEE** Sensors Journal. PP. 1-1. 10.1109/JSEN.2019.2947446.

Problem definition

Enhance the classification accuracy and provide online real-time feedback for enhancing the player stroke style by monitoring different body joints.

System Overview



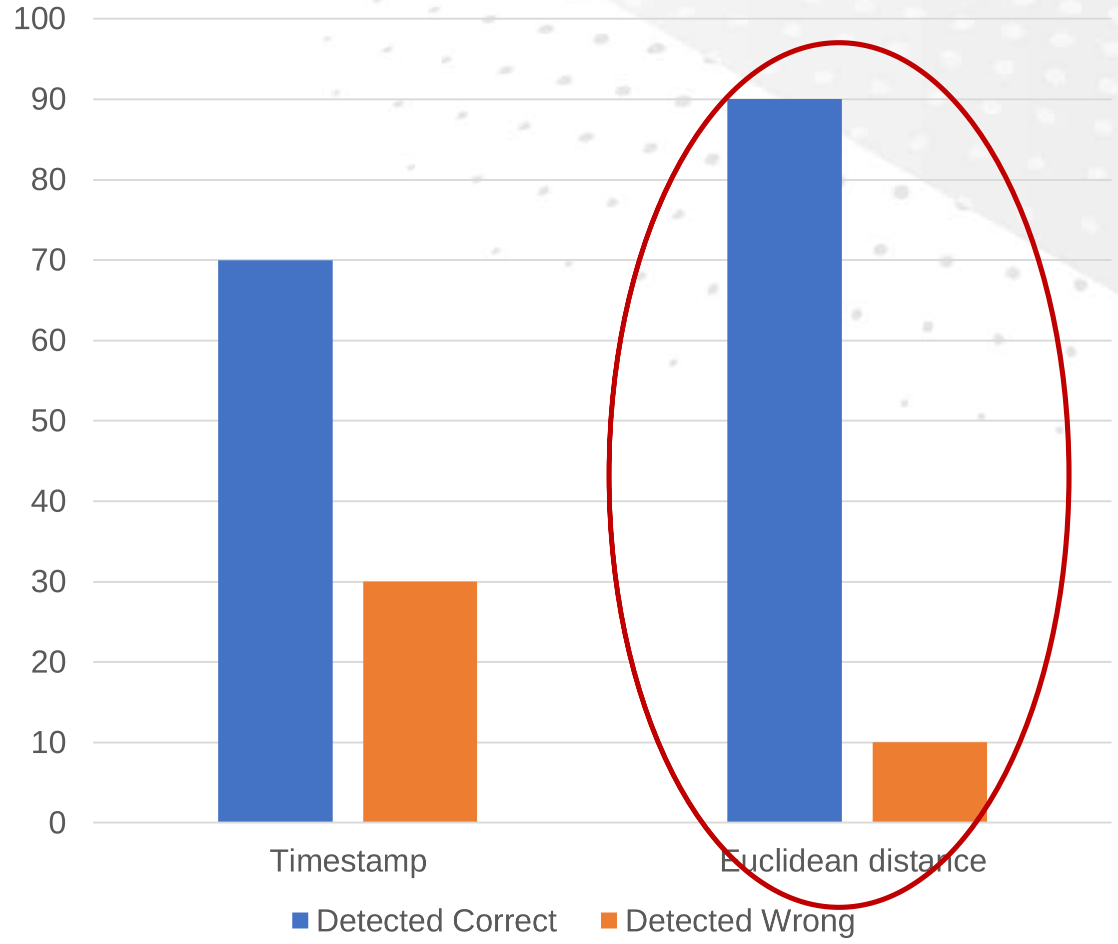
Pre-processing – stroke segmentation

Timestamp based

- Stroke was cut before or after the stroke finishes.

Euclidean distance based

- Neglected unwanted movements and achieved more accuracy than timestamp by 20%

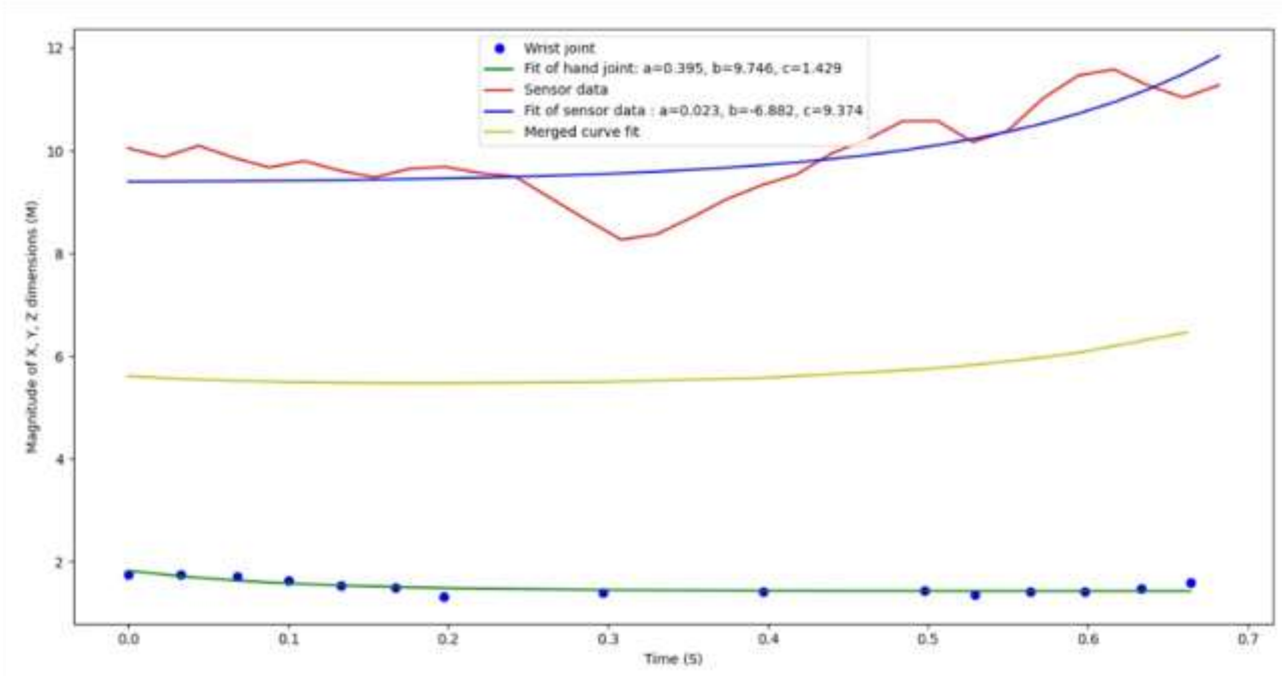


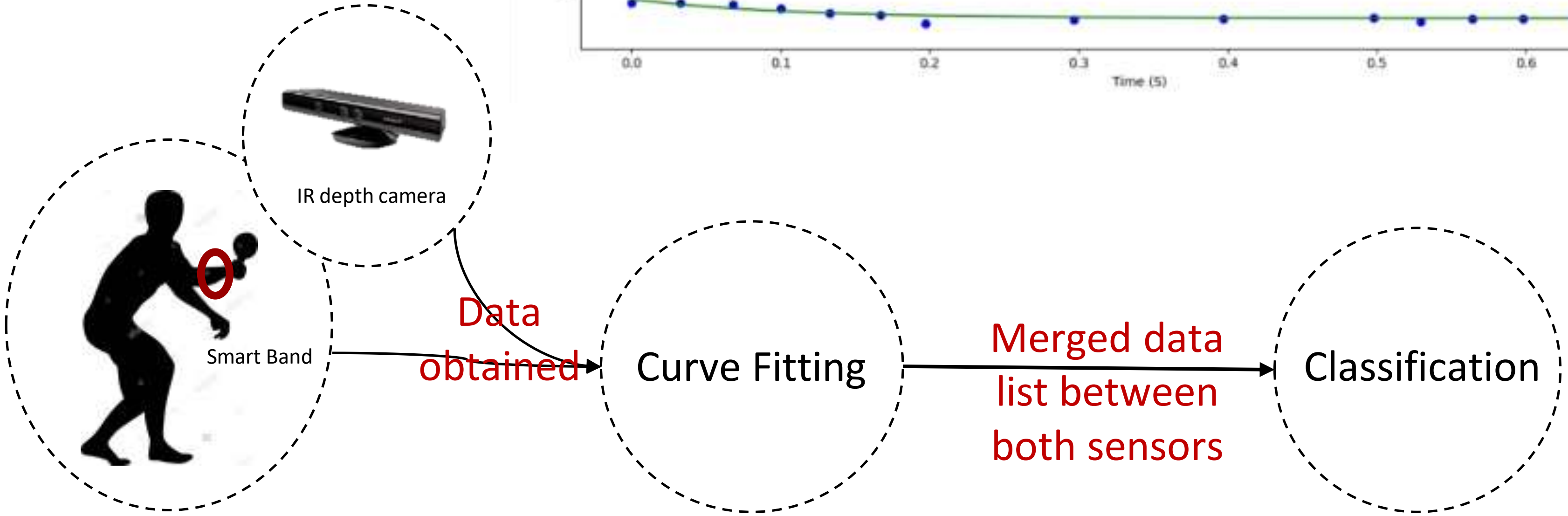
Pre-processing – Sensor fusion

Curve fitting

- Kinect, Accelerometer and Gyroscope data synchronized and merged.

Wrist Joint

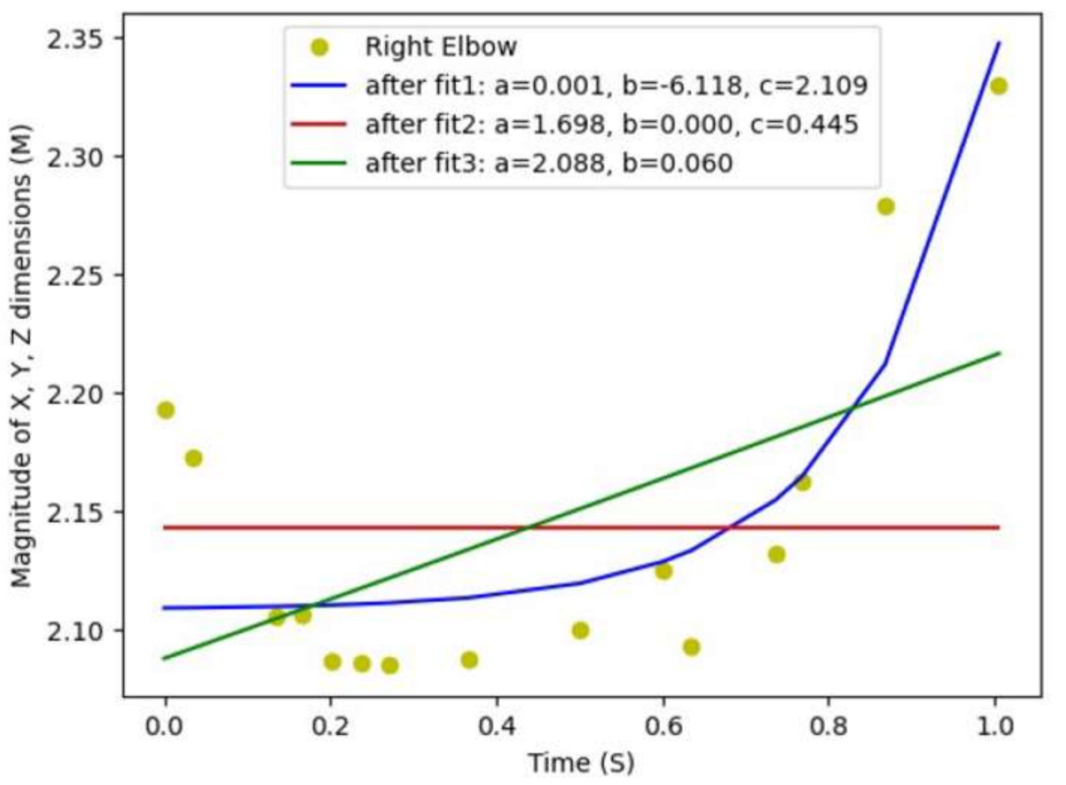




Curve fitting methodologies

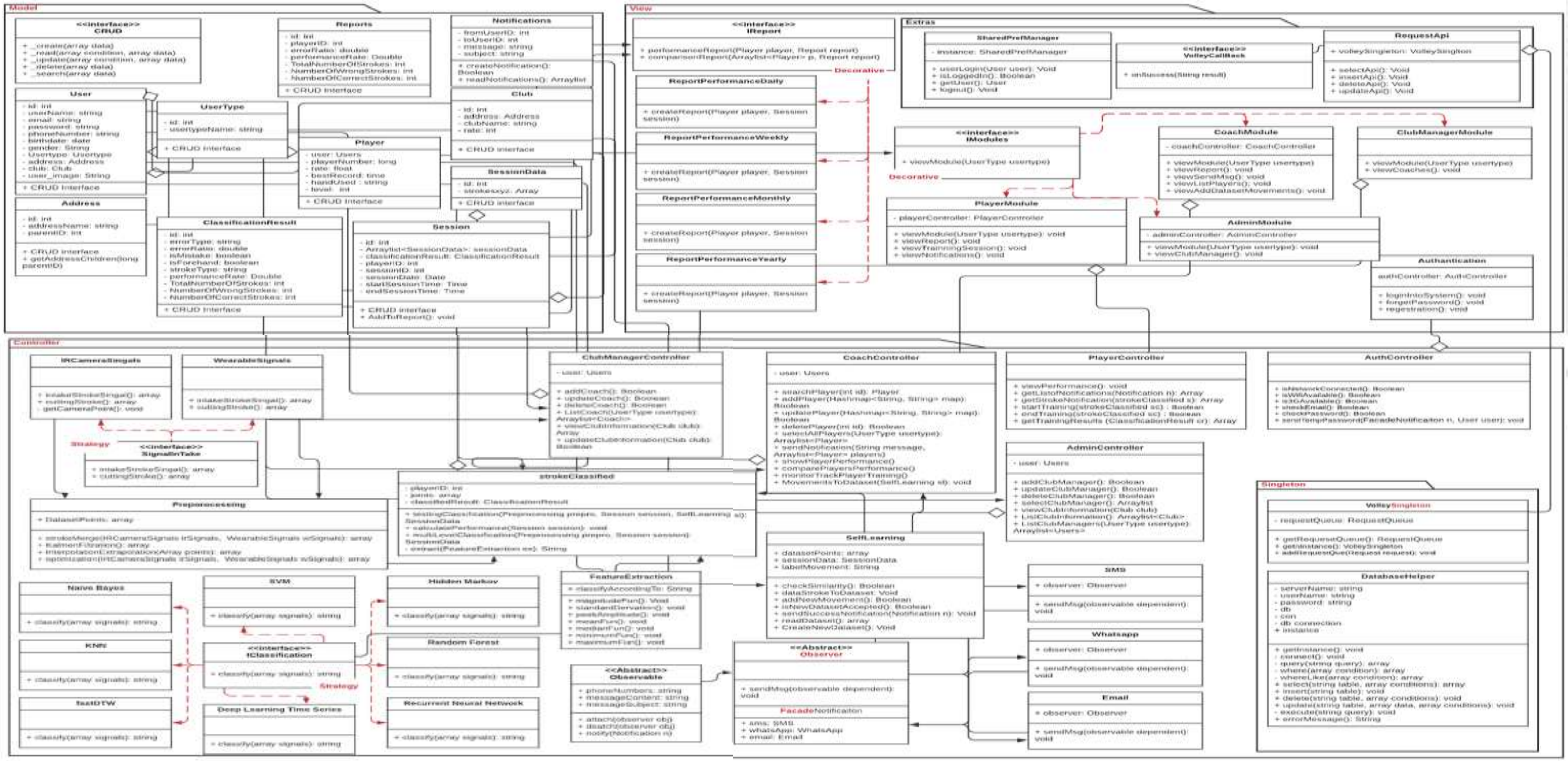
The usage of the curve fitting equations depends on the type of data. According the following graph shows that equation [1] was the most optimal equation used in the system.

	CF Equations	Constraints	Graph Curves
1	$a \times e^{-b \times x} + c$	No constraints	Blue Curve
2		0 < a <= 3.0; 0 < b <= 1.0; 0 < c <= 0.5;	Red Curve
3	$a \times e^{b \times x}$	Initial guesses a = 2.088; b = 0.060	Green Curve



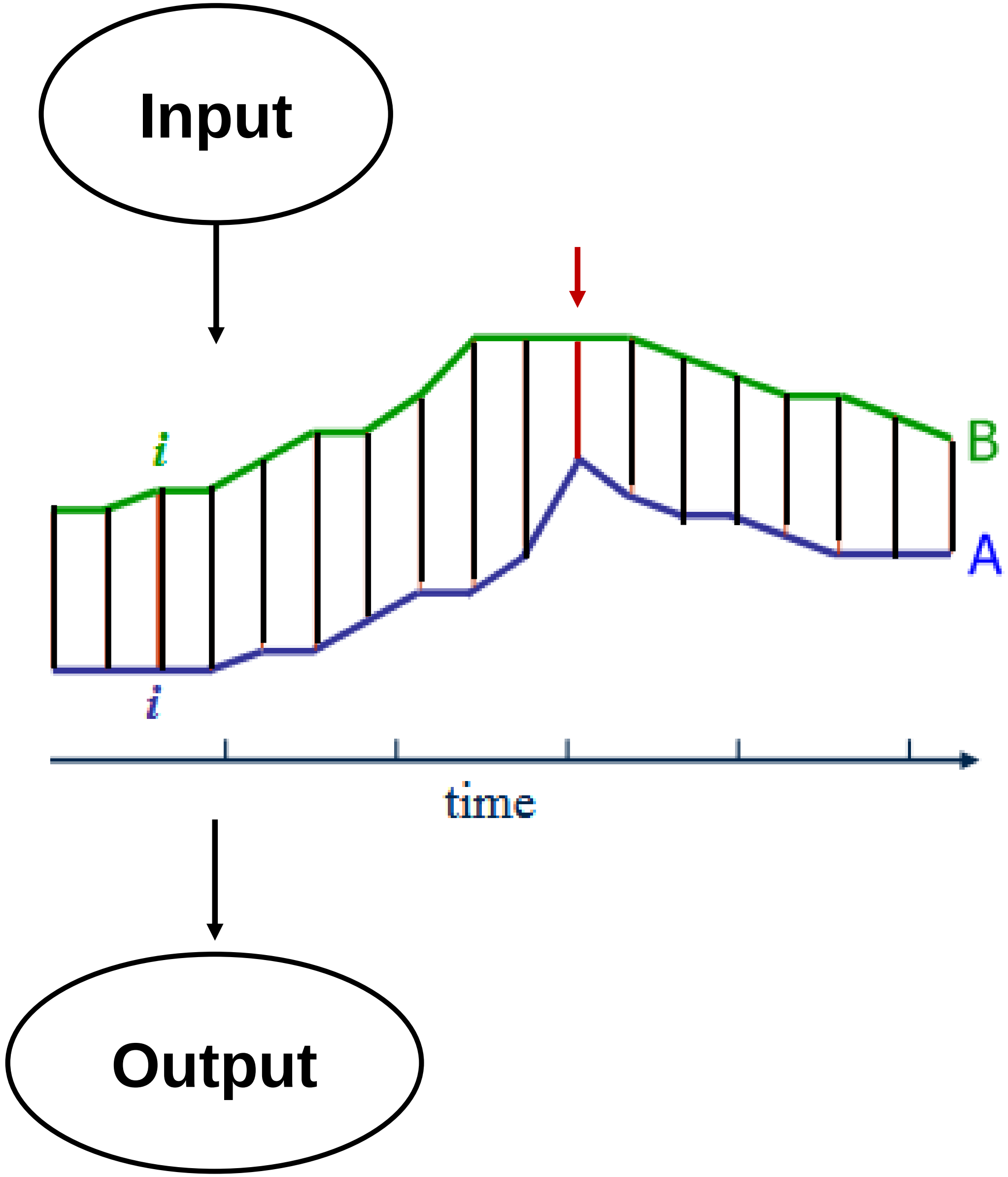
Processing

Misr International University
جامعة مصر الدولية



Through system we used Software engineering design patterns.

Why FastDTW Algorithm?

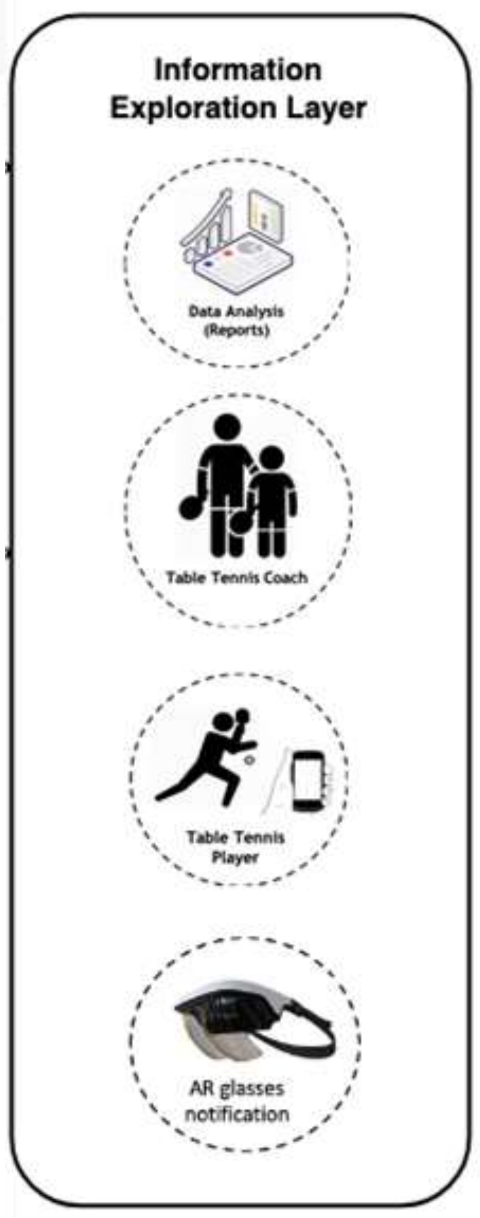


- 2 Arrays:
  - 1- Each array from the dataset.
  - 2- Filtered array of stroke intake.

- The algorithm is able to find the optimal alignment between **the two time series**.
- It **finds the nearest optimal alignment** between the 2 arrays and compare them and get the shortest distance between the 2 waves.
- FastDTW complexity is O(N) time and memory.

Output

- ❑ Coach Module:
    - ❖ Strokes made by player.
    - ❖ Real-time player rating.
    - ❖ Mistakes took place.
- ❑ Player module:
    - ❖ Vibration.
    - ❖ AR Application for data viewing.
- ❑ Reports Module:
    - ❖ Daily, weekly, and monthly.
    - ❖ Player's performance comparison.

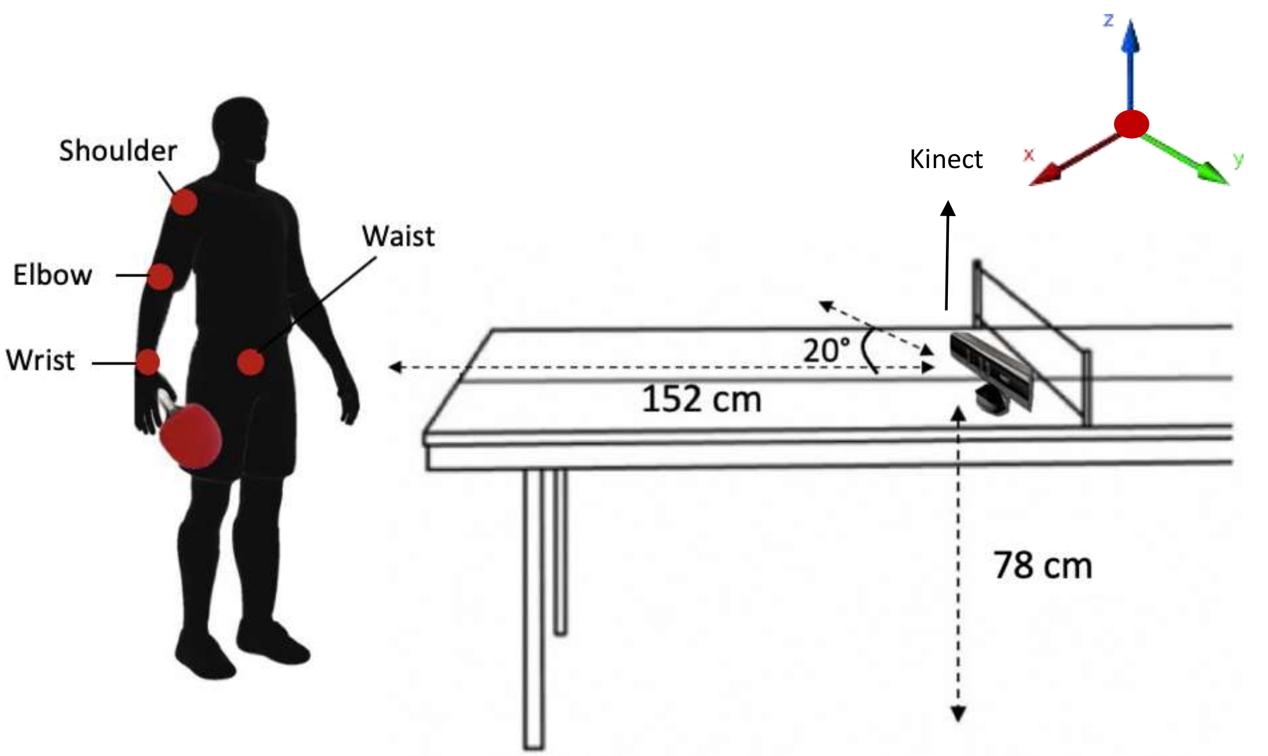



Provide a real-time application.
Notification system acceptable by player's environment.
Challenges
Increase classification accuracy.
Detect stroke mistakes according to different joints.

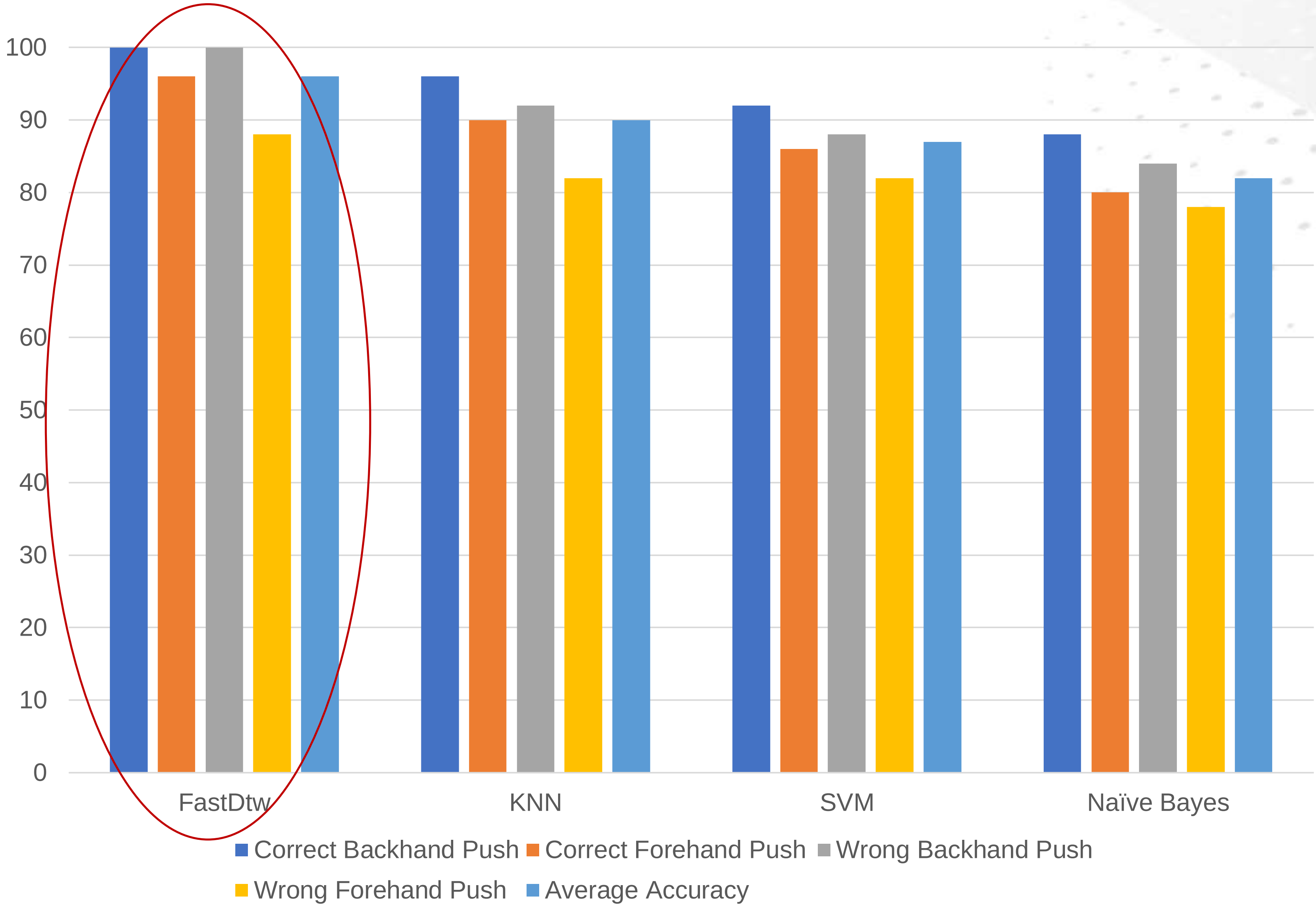
Experiments Setup (1/3) – Usage of Kinect

Experiments objective

- Test different algorithms on Kinect usage.
- Make a user dependent and independent study on the algorithms.
- Over all data collected is 960 strokes from 6 different players.



Experiment 1.1 – Algorithm Comparison on Kinect usage



FastDTW achieved high accuracy 96%

Experiment 1.2 – User dependent and independent

Different classification algorithm comparison on user-dependent.

	FastDTW	KNN	SVM	Naive Bayes
Correct Backhand Push	100%	91.67%	91.67%	83.33%
Correct Forehand Push	100%	91.67%	100%	91.67%
Wrong Backhand Push (Elbow joint)	100%	100%	83.33%	66.66%
Wrong Backhand Push (Wrist joint)	83.33%	83.33%	66.66%	66.66%
Wrong Backhand Push (Shoulder joint)	100%	100%	100%	83.33%
Wrong Backhand Push (Waist joint)	83.33%	66.66%	66.66%	50.00%
Wrong Forehand Push (Elbow joint)	100%	100%	83.33%	83.33%
Wrong Forehand Push (Wrist joint)	83.33%	83.33%	83.33%	66.66%
Wrong Forehand Push (Shoulder joint)	100%	83.33%	83.33%	83.33%
Wrong Forehand Push (Waist joint)	66.66%	66.66%	66.66%	50%
Precision	0.8276	0.7586	0.6970	0.5833
Recall	1.0000	0.9167	0.9583	0.8750
F-Measure	0.9057	0.8302	0.8070	0.7000
Accuracy Average	91.67%	86.67%	82.50%	72.50%

user-dependent classification P-value was 0.015984569 which means there is a difference in the algorithm accuracy, depending on the user.

Different classification algorithm comparison on user-independent.

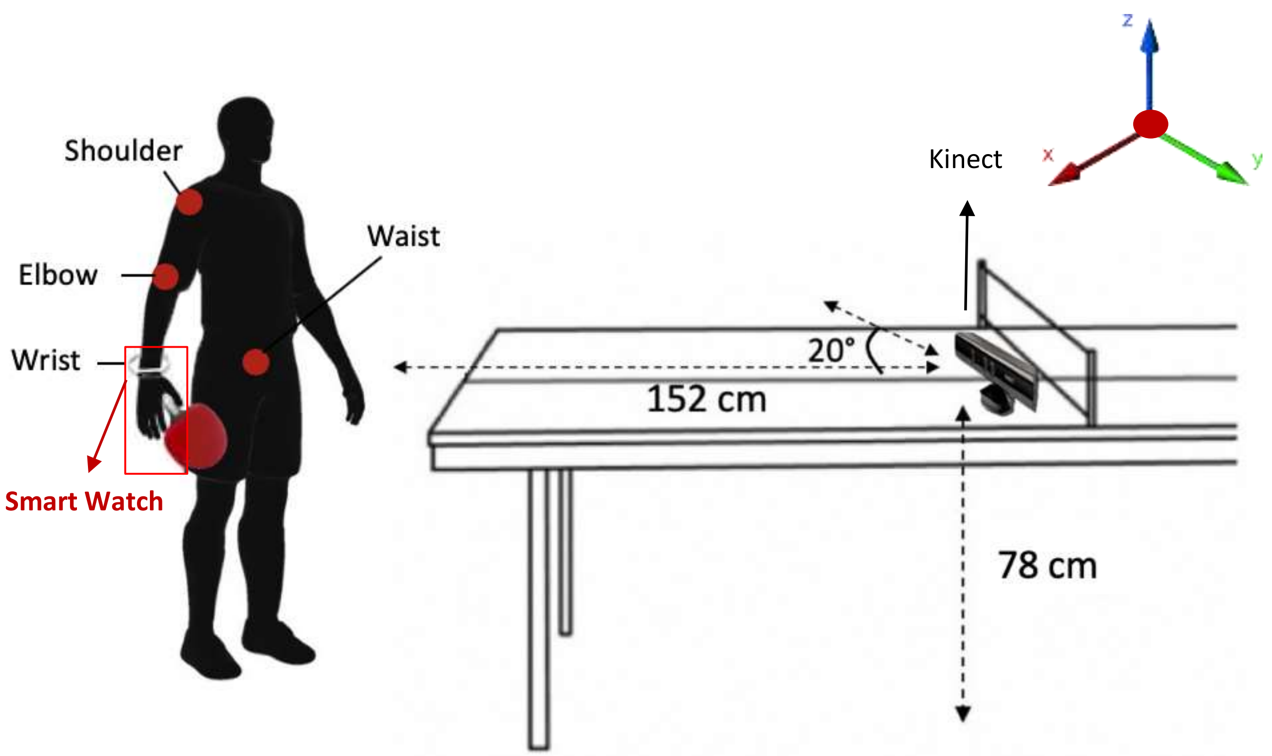
	FastDTW	KNN	SVM	Naive Bayes
correct Backhand Push	100%	96.87%	86.45%	73.95%
correct Forehand Push	97%	93.75%	84.37%	79.16%
wrong Backhand Push (Elbow joint)	99.50%	99.50%	78.50%	76%
wrong Backhand Push (Wrist joint)	85%	85%	76.5%	74.50%
wrong Backhand Push (Shoulder joint)	98%	98%	92%	84%
wrong Backhand Push (Waist joint)	83%	67.50%	60%	51.50%
wrong Forehand Push (Elbow joint)	94.50%	96%	83%	72.50%
wrong Forehand Push (Wrist joint)	79.50%	84%	72.5%	61%
wrong Forehand Push (Shoulder joint)	92.5%	85%	78%	71.50%
wrong Forehand Push (Waist joint)	66%	68.5%	56%	50%
Precision	0.7851	0.7521	0.6357	0.5252
Recall	0.989	0.9479	0.8542	0.7604
F-Measure	0.8756	0.8387	0.7289	0.6213
Accuracy Average	89.50%	87.41%	76.73%	69.31%

user-independent classification, there was a statistically significant difference between groups was determined by one-way ANOVA ($F(3,36) = 6.808490236$, $p = 0.000942206$).

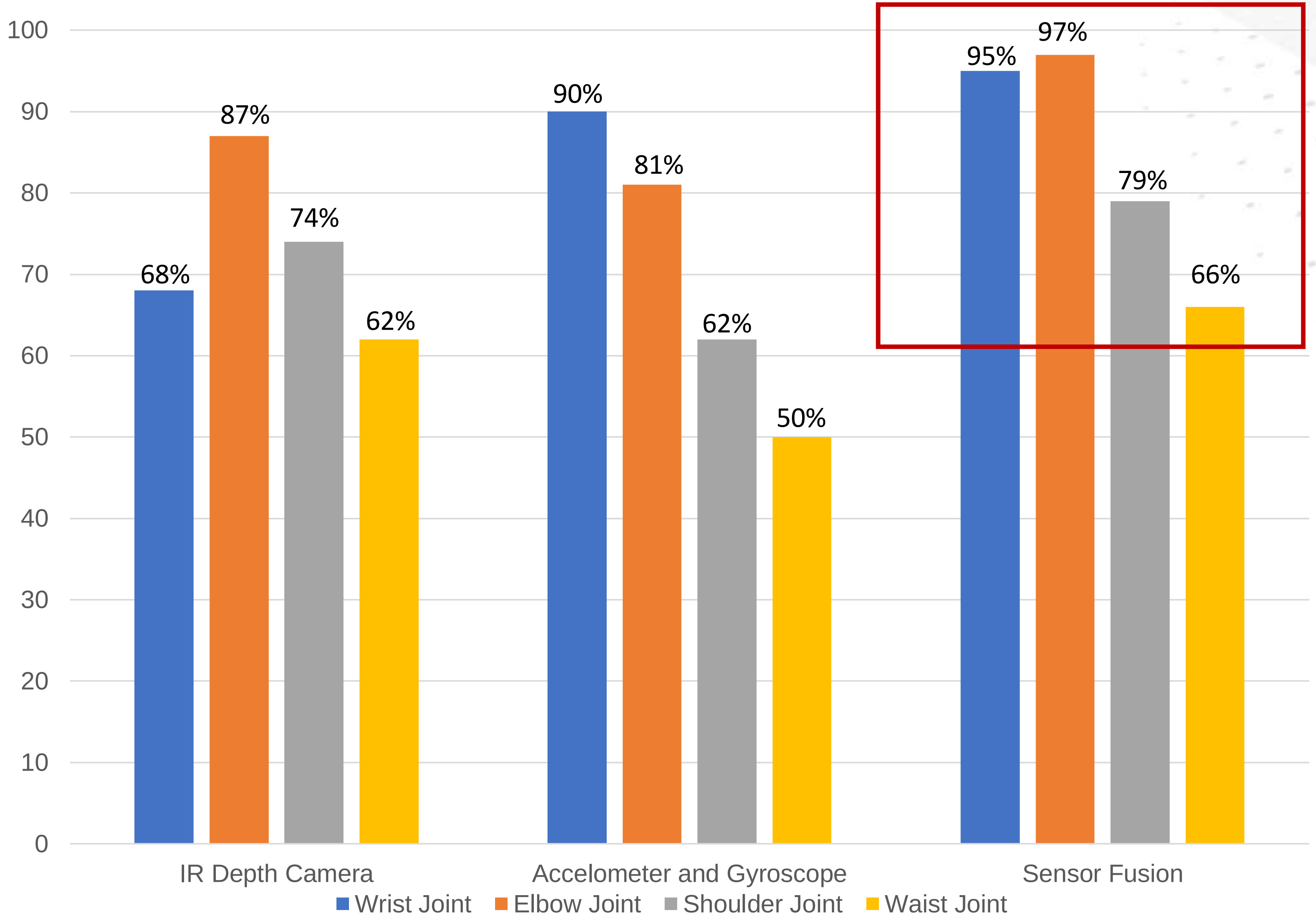
Experiments Setup (2/3) – Sensor Fusion

Experiments objective

- Compare between the usage of different sensors and sensor fusion.
- Test different algorithms on sensor fusion technique.
- Over all data collected is 1000 strokes from 8 different players.



Experiment 2.1 – Sensor fusion & Sensors comparison

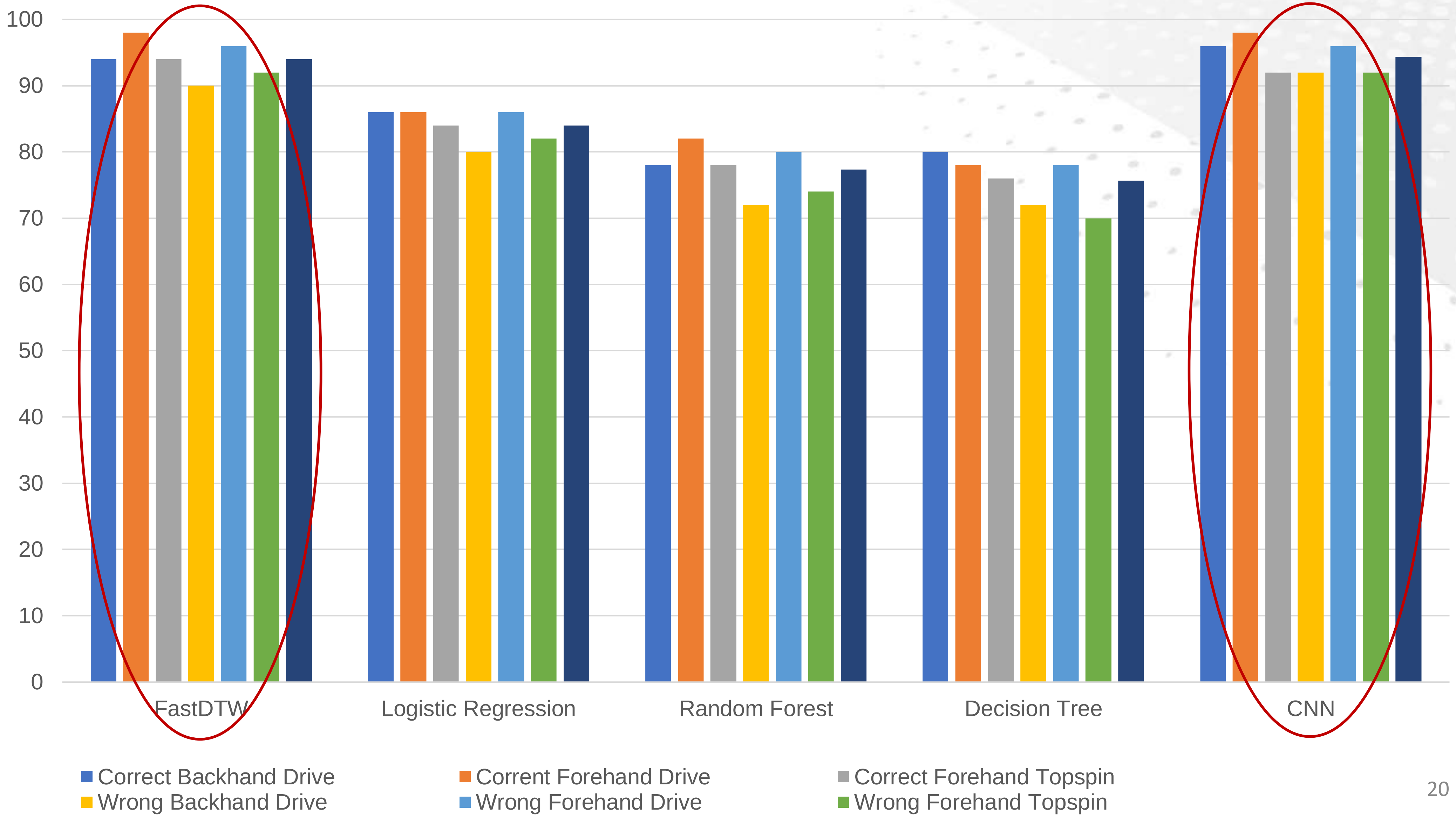


Detection accuracy increased for all joints:

Wrist	by	39.7%
Elbow	by	11.5%
Shoulder	by	6.8%
Waist	by	6.5%

Experiment 2.2 – Classification accuracy on fusion

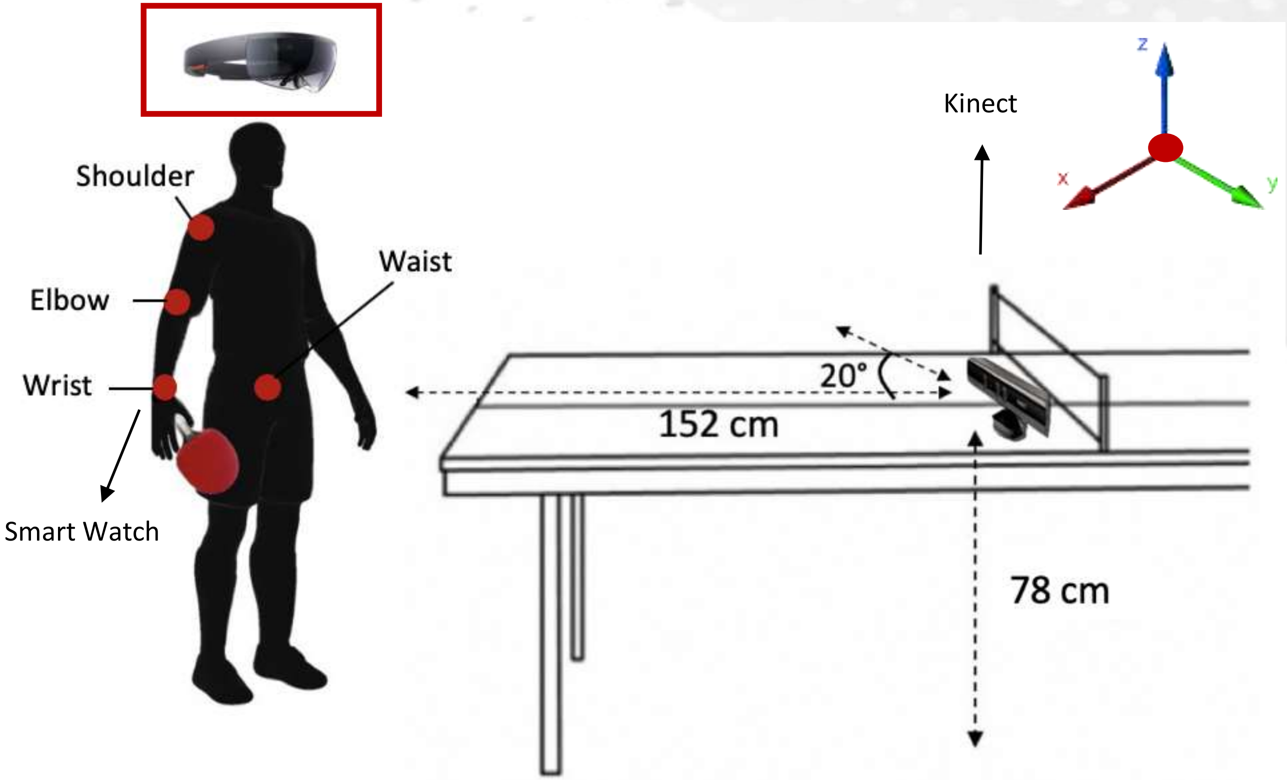
Average Time (Single Stroke)	0.98 sec	1.73 sec	2.08 sec	0.99 sec	2.15 sec



Experiments Setup (3/3) – Usability Study

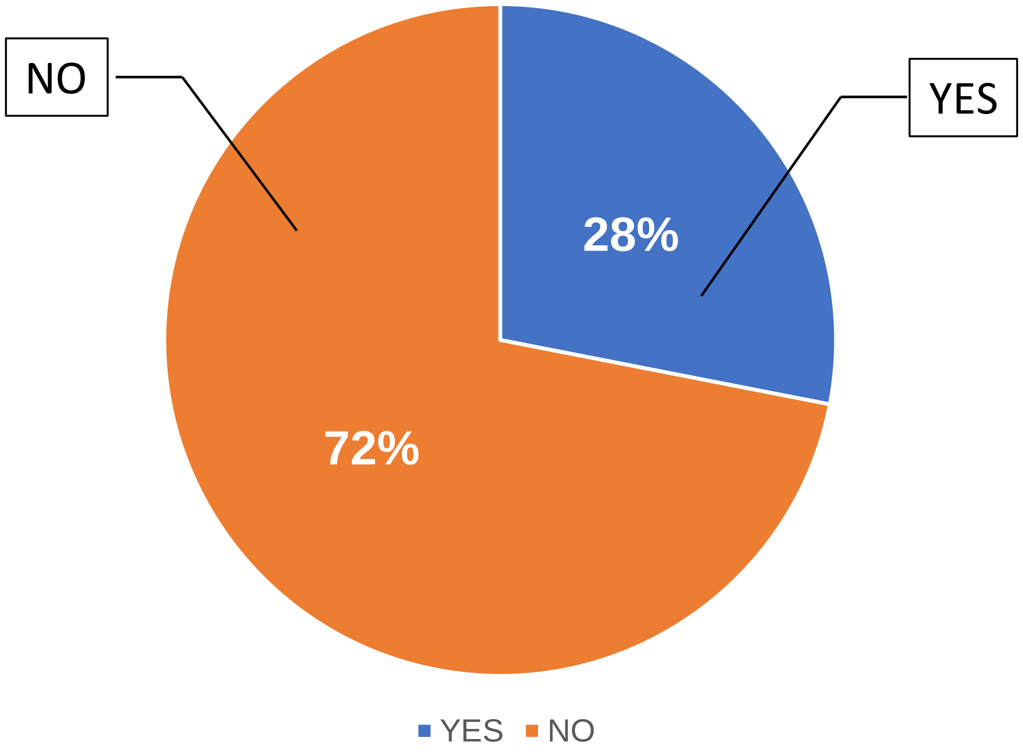
Experiments objective

- Enhance the user feedback by using Augmented Reality.
- Measure the learning style enhancement of the system.
- The system was tested on 50 different players.

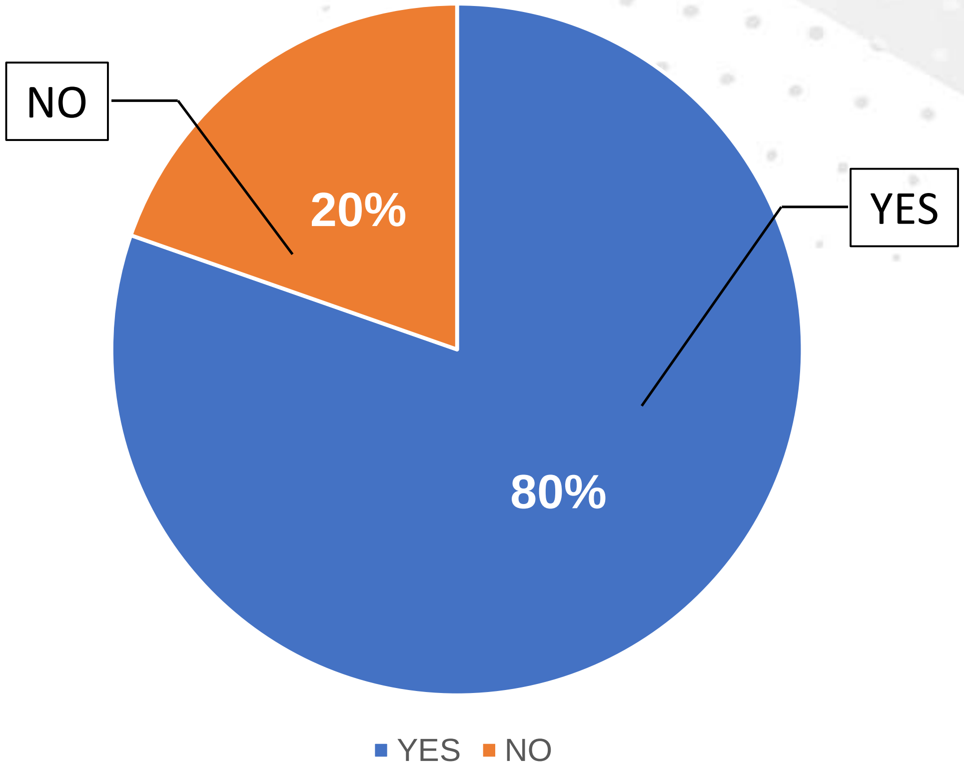


Experiment 3.1 – Usage of Augmented Reality

The usage of AR was reported as highly comfortable 40 of 50 of the players were very satisfied.

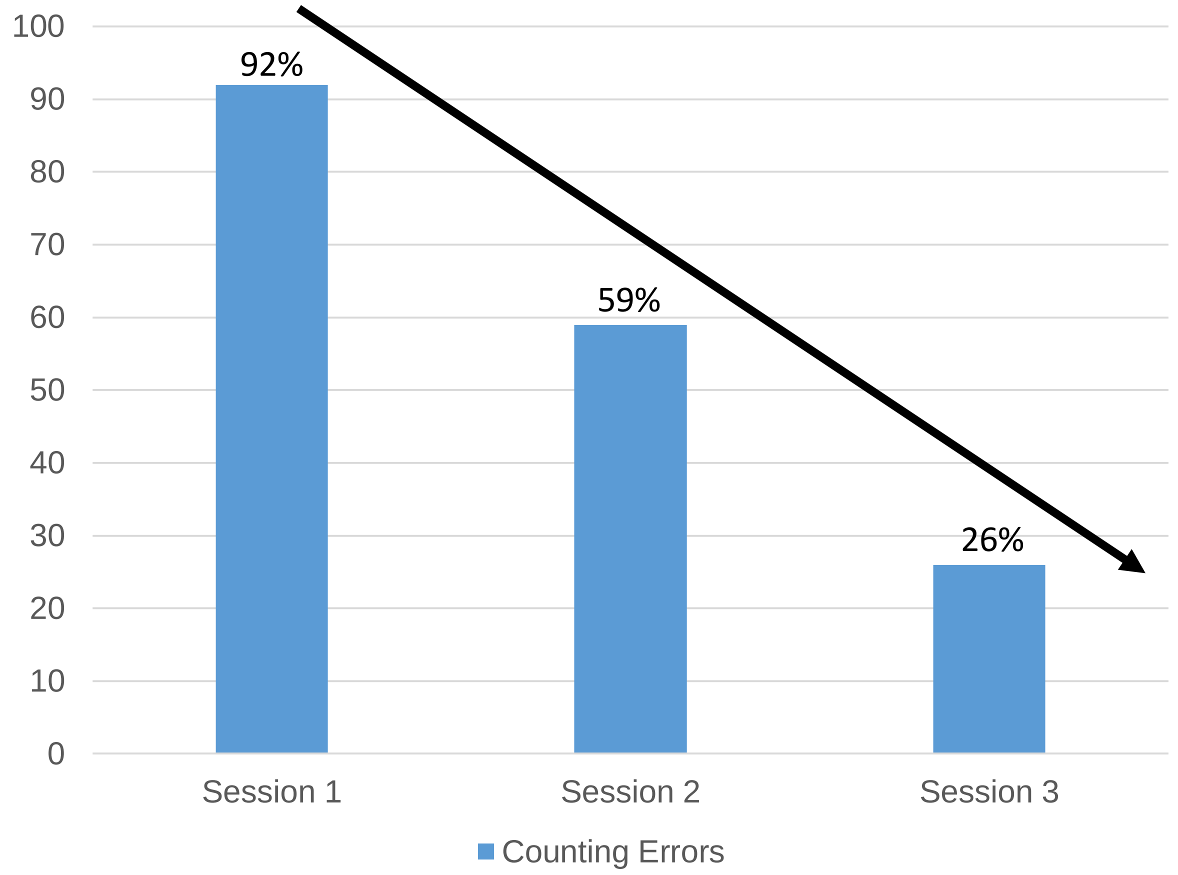


Distractibility Chart for the usage of AR



Comfortability Chart for the usage of AR

Experiment 3.2 – Usability study on Learning Style



Each session contains the average percentage of mistakes done by players on different strokes. The players performance improves through the usage of the system.

Demo

Video is uploaded

Feedback on system

Video is uploaded

Contribution

1. Created real-time application with the usage of kinect sensor.

Published a paper in The 11th International Conference on Ambient Systems, Networks and Technologies in Poland. Titled **as “Online detection and classification of in-corrected played strokes in table tennis using IR depth camera.”**

ANT 2020

ELSEVIER

2. Added more error types, and increased the classification accuracy.

Published a paper in the 17th International Conference on Mobile Systems and Pervasive Computing in Belgium. Titled by **“IPingPong: A Real-time Performance Analyzer System for Table Tennis Stroke’s Movements.”**

MobiSPC

3. Sensor fusion and measure classification accracy with time responding.

Submitted a paper in INASS Journal. Titled by **“Multi-Sensor Fusion for Online Detection and Classification of Table Tennis Strokes.”**

INASS

4. Supported the system with AR for notification, and made a usability study.

Submitted a paper in JSPAN Journal. Titled by **“Usability Study for a comprehensive table tennis AR based training system with the focus on players' strokes.”**

IASKS

THANK YOU!

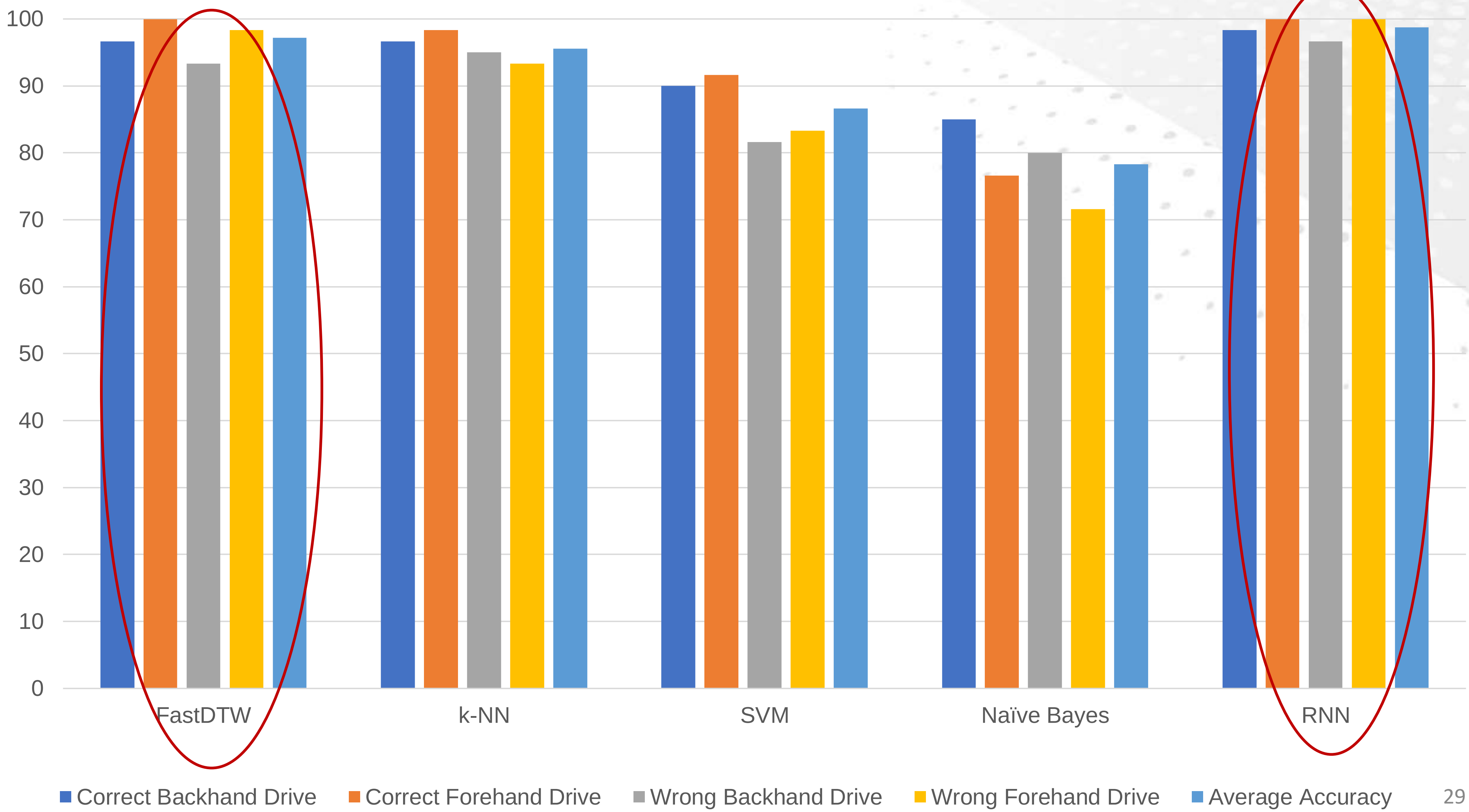
Any Questions?

Future work

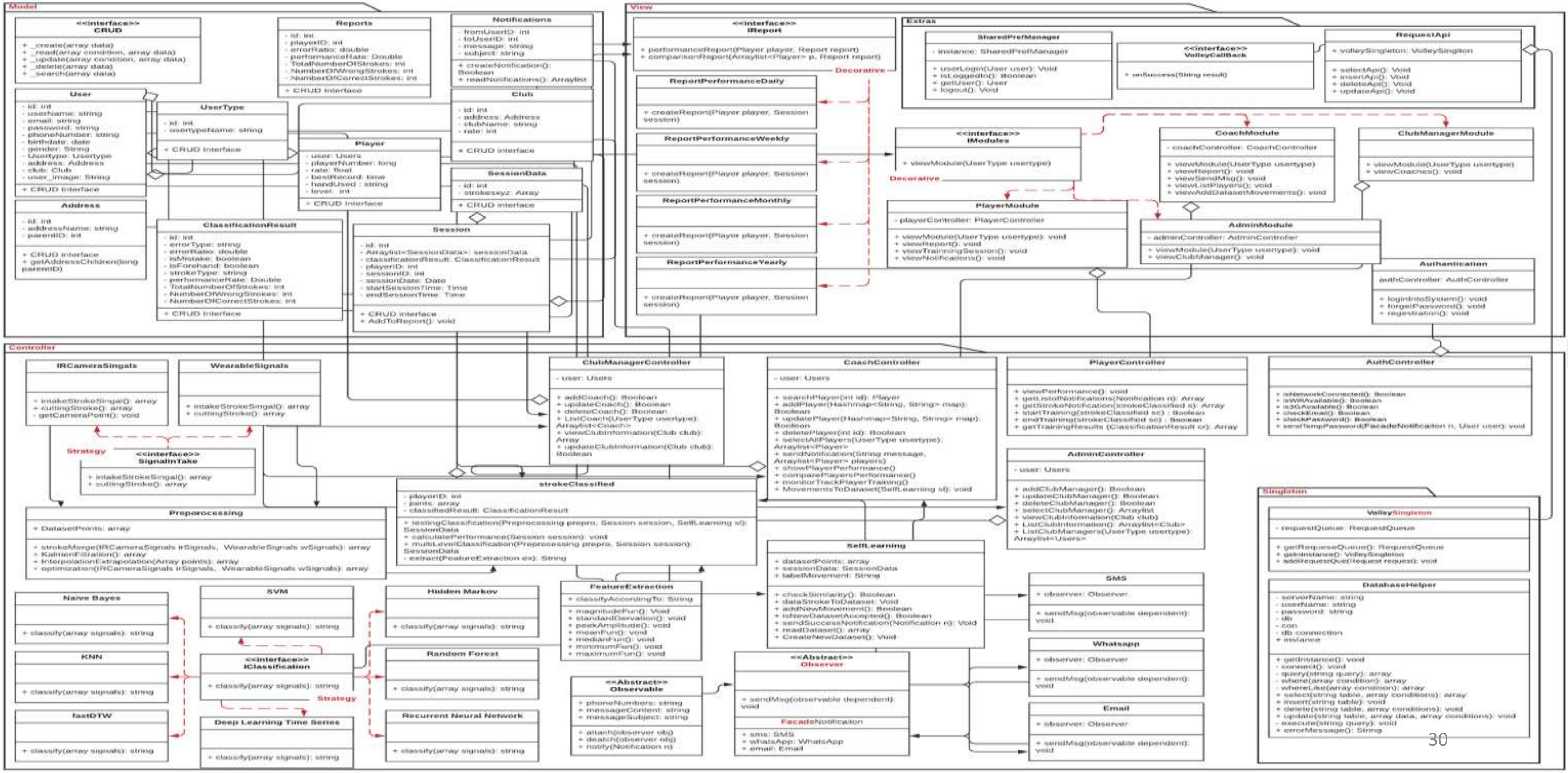
- Increase the size of the dataset to maintain system stability.
- Add an extra sensor to detect the legs' movements of the player.
- Evolve the system to be working on multi-classifier layers.
- Enhance the AR system used from just notification to a full guide AR system.
- Involve the system into a computer training game at home.

Experiment 2.3 – Classification accuracy on fusion

Average Time (Single Stroke)	1.38 sec	4.14 sec	4.89 sec	2.76 sec	6.41 sec



Class Diagram



Design Patterns Used

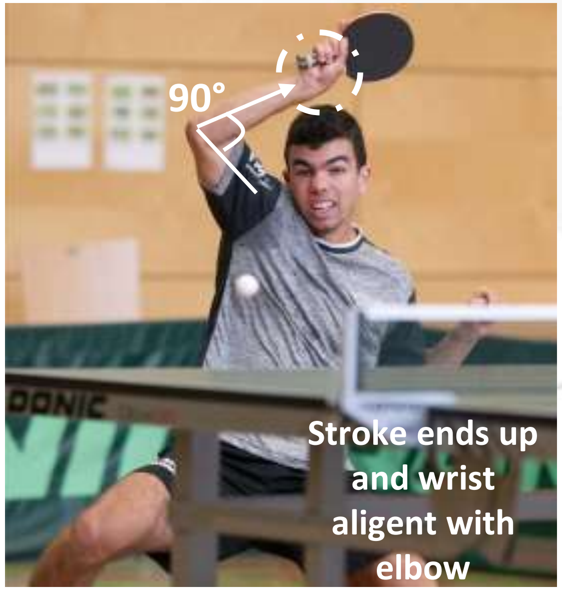
1. Strategy – for the extistance of different algorithms to apply on expriments.
2. Singleton – for database connection.
3. Observer and Facade – for the notification system in the application.
4. Decorative – for the presence of different reporting methods and modules.

Dataset Screenshot

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
-0.16262	-0.31477	1.085467	-0.18875	-0.34113	1.369497	-0.0878	-0.335	1.56864	-0.30598	-0.38198	1.486046	-0.18827	-0.42695	1.412924	CorrectForehandDrive		
-0.08154	-0.38533	1.177685	-0.14048	-0.20111	1.467108	-0.13438	-0.37417	1.605175	-0.26893	-0.35269	1.372269	-0.23643	-0.42252	1.503543	CorrectForehandDrive		
-0.13059	-0.35375	1.081544	-0.18394	-0.38425	1.368667	-0.07936	-0.43544	1.565205	-0.29302	-0.40681	1.444943	-0.18298	-0.46279	1.438332	CorrectForehandDrive		
-0.14883	-0.31255	1.089852	-0.17869	-0.35905	1.378141	-0.05294	-0.38643	1.565934	-0.28087	-0.41363	1.479733	-0.17197	-0.45408	1.428208	CorrectForehandDrive		
0.125948	-0.42899	1.463241	-0.10762	-0.38425	1.325751	-0.09618	-0.34949	1.552546	-0.25778	-0.37314	1.390336	-0.19565	-0.43453	1.499396	CorrectForehandDrive		
-0.05925	-0.43783	1.103538	-0.10338	-0.40848	1.314653	-0.12082	-0.4087	1.543271	-0.30839	-0.33453	1.301308	-0.23001	-0.40934	1.286395	CorrectForehandDrive		
-0.14781	-0.44841	1.068118	-0.15677	-0.40724	1.281398	-0.13922	-0.42713	1.508694	-0.32725	-0.34092	1.288074	-0.26113	-0.42802	1.323809	CorrectForehandDrive		
0.056279	-0.3202	1.359298	-0.20403	-0.37837	1.310609	-0.13383	-0.31557	1.517041	-0.35135	-0.35251	1.43545	-0.2464	-0.39989	1.348485	CorrectForehandDrive		
-0.11431	-0.6381	1.511584	-0.12564	-0.42457	1.577434	-0.18177	-0.28713	1.548078	-0.31354	-0.35435	1.367083	-0.25923	-0.41137	1.466753	CorrectForehandDrive		
-0.27005	-0.6811	1.599682	-0.17572	-0.49602	1.609134	-0.1788	-0.35016	1.571338	-0.32419	-0.33945	1.32484	-0.27862	-0.40488	1.333853	CorrectForehandDrive		
0.122983	-0.37605	1.410885	-0.10755	-0.52283	1.45191	-0.14717	-0.38618	1.532425	-0.35267	-0.37758	1.427386	-0.23269	-0.42448	1.409952	CorrectForehandDrive		
0.027036	-0.41281	1.337618	-0.02821	-0.39179	1.414909	-0.12443	-0.32508	1.534321	-0.30511	-0.36272	1.415159	-0.19979	-0.41505	1.471714	CorrectForehandDrive		
0.136435	-0.28742	1.383086	-0.05685	-0.4692	1.481712	-0.13066	-0.34321	1.5515	-0.31599	-0.36311	1.401414	-0.22825	-0.41954	1.474572	CorrectForehandDrive		
-0.29102	-0.75351	1.628957	-0.21964	-0.5632	1.582728	-0.19354	-0.35642	1.582377	-0.33238	-0.42699	1.432122	-0.2902	-0.47601	1.451871	CorrectForehandDrive		
-0.3382	-0.78795	1.573456	-0.27503	-0.59133	1.54894	-0.21201	-0.39515	1.532692	-0.34808	-0.42734	1.373006	-0.29028	-0.48459	1.361165	CorrectForehandDrive		
-0.00233	-0.32143	1.338845	-0.25233	-0.42783	1.382353	-0.17561	-0.33003	1.489342	-0.35662	-0.32522	1.326169	-0.29754	-0.38816	1.403928	CorrectForehandDrive		
0.0456	-0.35328	1.360861	-0.08653	-0.45269	1.48871	-0.17028	-0.32509	1.515393	-0.34823	-0.37458	1.355821	-0.27363	-0.43496	1.419081	CorrectForehandDrive		
0.121377	-0.39195	1.602451	0.256362	-0.46998	1.687105	0.107229	-0.3861	1.872711	-0.20174	-0.71029	1.847104	-0.06341	-0.67691	1.807727	WrongForehandDrive		
0.058907	-0.38948	1.616176	0.244018	-0.47996	1.721626	0.078744	-0.38445	1.880657	-0.22521	-0.70126	1.845085	-0.08722	-0.66864	1.810729	WrongForehandDrive		
-0.0283	-0.45782	1.619455	0.172696	-0.4589	1.734566	-0.00113	-0.36849	1.895772	-0.23971	-0.67666	1.843564	-0.1154	-0.64449	1.813255	WrongForehandDrive		
0.002123	-0.53768	1.565617	0.140512	-0.48208	1.735296	0.005544	-0.40871	1.84292	-0.12378	-0.66168	1.861485	0.025802	-0.65638	1.822775	WrongForehandDrive		
-0.01085	-0.48307	1.480367	0.146052	-0.42882	1.702515	0.00623	-0.41158	1.827469	-0.08718	-0.65241	1.855713	0.036053	-0.63817	1.818062	WrongForehandDrive		
-0.03868	-0.3932	1.413388	-0.04532	-0.40638	1.610508	0.006156	-0.41547	1.819761	-0.13702	-0.67108	1.863773	-0.01688	-0.62764	1.793285	WrongForehandDrive		
0.018078	-0.24141	1.291973	0.168189	-0.42113	1.538376	0.072072	-0.42962	1.722779	-0.12567	-0.67428	1.843456	0.047549	-0.61441	1.803311	WrongForehandDrive		
0.017453	-0.18661	1.364016	0.098894	-0.37449	1.568588	0.017169	-0.45334	1.78243	-0.13962	-0.59177	1.821703	0.024097	-0.59198	1.814972	WrongForehandDrive		
0.027691	-0.15317	1.354084	0.099319	-0.37995	1.577206	0.010838	-0.45843	1.786238	-0.14486	-0.58281	1.817366	-0.0026	-0.58052	1.821868	WrongForehandDrive		
-0.13846	-0.33819	1.415487	0.068197	-0.40666	1.45557	-0.07404	-0.33151	1.475892	-0.20766	-0.63737	1.828099	-0.09963	-0.58774	1.718305	WrongForehandDrive		
-0.11218	-0.41775	1.438106	0.102124	-0.52036	1.521312	-0.05803	-0.45218	1.467522	-0.24285	-0.63987	1.817364	-0.12392	-0.6269	1.786433	WrongForehandDrive		
-0.01169	-0.50562	1.439891	0.035585	-0.58242	1.730231	-0.1091	-0.46223	1.816599	-0.33362	-0.61067	1.816503	-0.18434	-0.5964	1.769216	WrongForehandDrive		

Strokes collected for dataset:

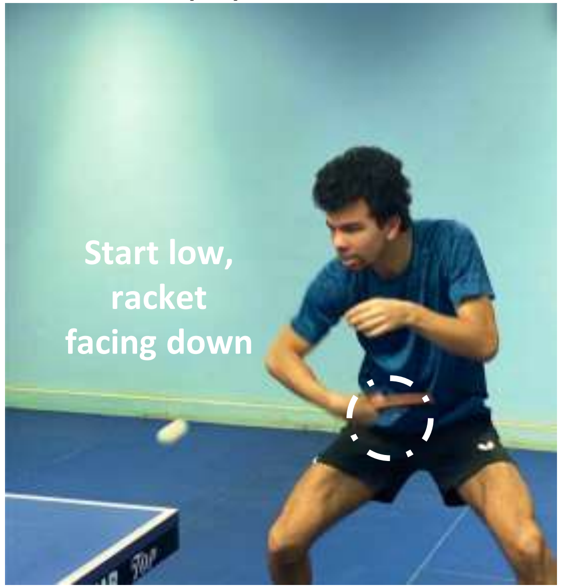
Forehand Drive



Forehand Topspin

Forehand Push

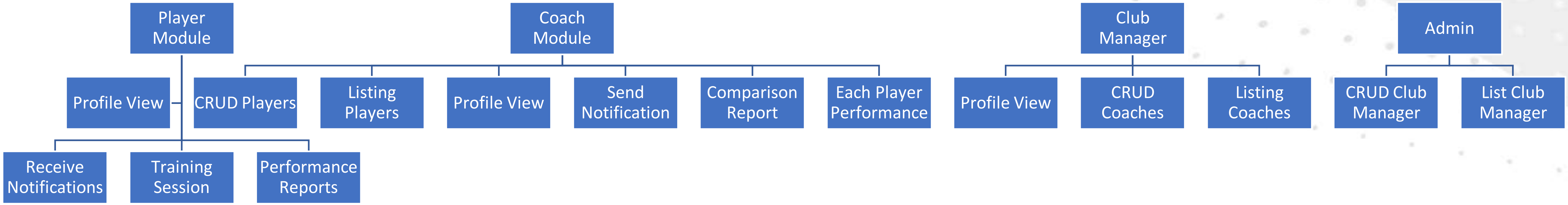
Backhand Drive



Backhand

Backhand Push

Android Application overview



FastDTW Algorithm

Toward accurate dynamic time warping in linear time and space

- They introduced FastDTW, a linear and accurate approximation of dynamic time warping (DTW).
- FastDTW uses a multilevel approach that recursively projects a warp path to a higher resolution and refines it.
- Result: an **average error of 8.6%** with a radius of only 1, and increasing the radius to 20 lowered the error to under 1%.

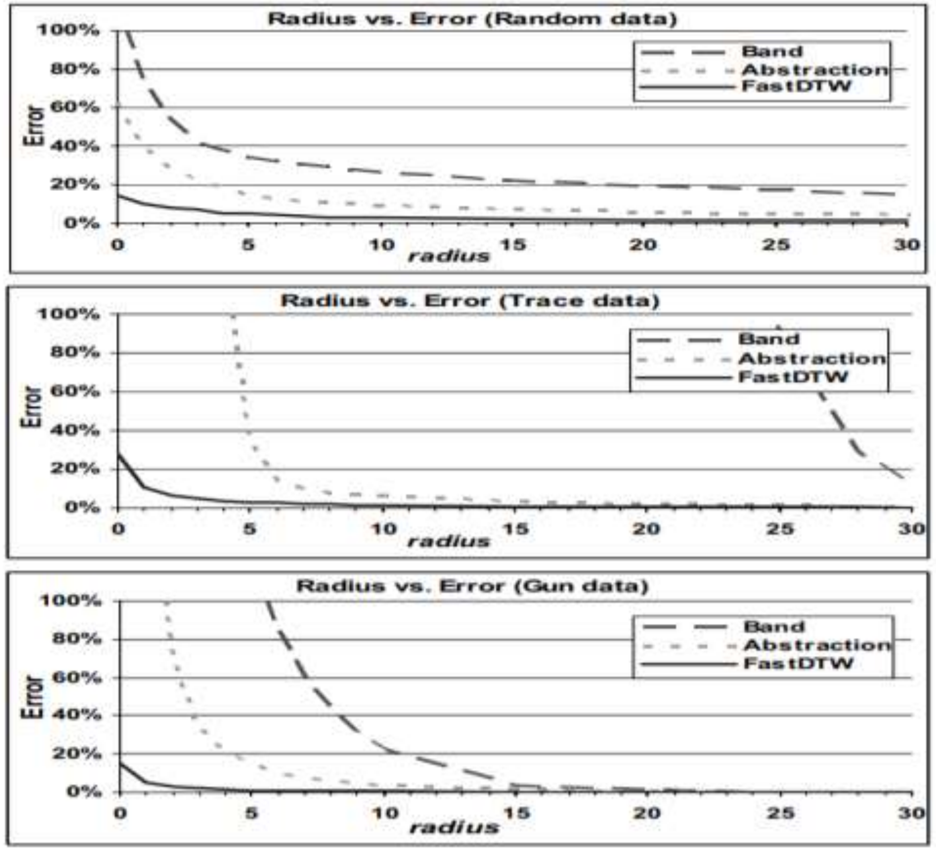


Fig. 9. Accuracy of FastDTW compared to Bands and Abstraction on all three groups of data.

S. Salvador and P. Chan, “Toward accurate dynamic time warping in linear time and space,” Intell. Data Anal., vol. 11, pp. 561–580, 10 2007.

Curve fitting

- This study proposes a curve fitting approach for classification problems.
- Results show that proposed classification approach with optimum values of constants and optimal feature set based on curve fitting has high accuracy rate.

Table 1. The average accuracy rates for the optimization of values of constants of Gaussian function by using proposed method and KNN.

	Iris dataset (%)	Heart dataset (%)	Balance scale dataset (%)
For reference data set by using proposed method	100	85.9	84.6
For validation set by using proposed method	94.6	83.7	93.2
KNN (For validation set)	94.6	55.5	79.2

Table 2. The average classification accuracy rates by using proposed method for stage of determination of optimal feature.

	Iris dataset (%)	Heart dataset (%)	Balance dataset (%)
For a part of the dataset in optimization stage by using proposed method with optimal reference feature set	97.3%	80.7%	63.7
For validation set by using proposed method with optimal reference feature set	97.3	80.0	71.1

Ahsaee, Mostafa & sadoghi yazdi, Hadi & Naghibzadeh, M.. (2011). Curve fitting space for classification. Neural Computing and Applications. 20. 273-285. 10.1007/s00521-010-0383-7.

Kalman Filter Algorithm

Improving Joint Position Estimation Of Kinect Using Anthropometric Constraint Based Adaptive Kalman Filter For Rehabilitation

- They proposed a novel algorithm to improve accuracy of Kinect skeletal joint.
- Using a second order Kalman filter with adaptive measurement noise to accurately track dynamic trajectory joint center location over time.
- Results: The STD of the bone length computed improves by at least 40%.

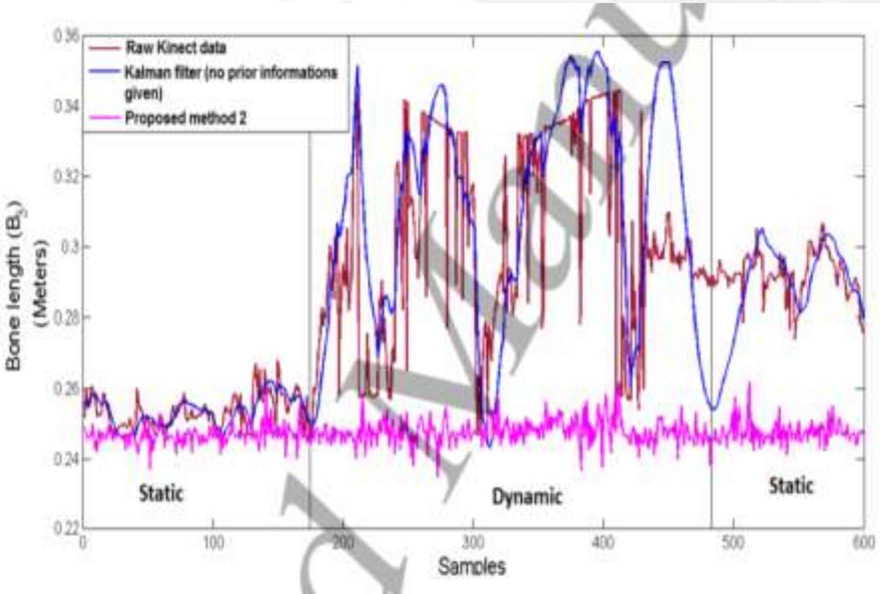
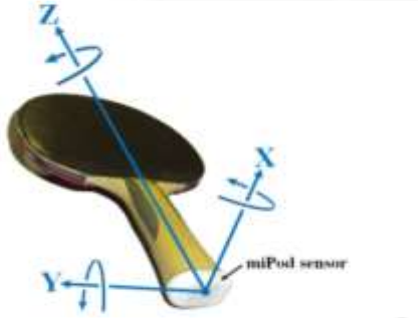


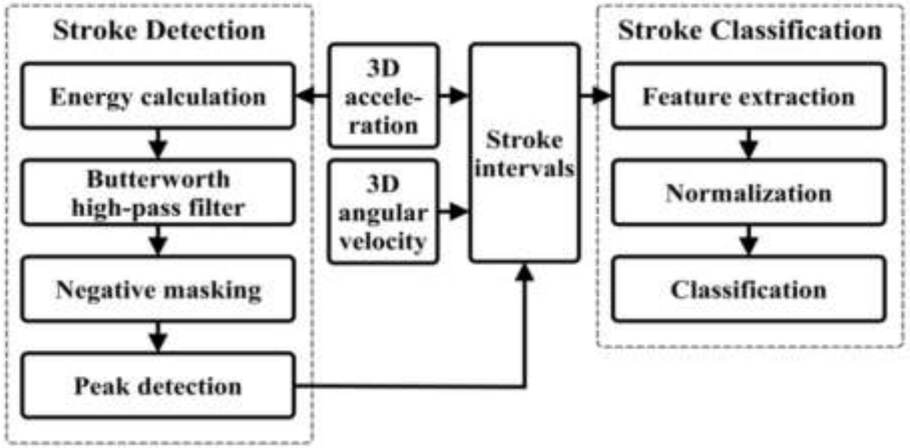
Figure 12. Performance of Kalman filter and our proposed approach. The left arm length is shown for the raw data, Kalman and Proposed method2.

P. Das, K. Chakravarty, A. Chowdhury, D. Chatterjee, A. Sinha, and A. Pal, “Improving joint position estimation of kinect using anthro-pometric constraint based adaptive kalman filter for rehabilitation,”Biomedical Physics and Engineering Express, vol. 4, 12 2017.

SVM Algorithm

- Used *miPod* sensor attached to the racket handle.
- Detected and classified 8 types of strokes with overall Precision of 95.7%
- Best accuracy was SVM algorithm.
- Classification based on the player movement of the racket.
- Detection the wrist movement.
- Offline Feedback.





Blank, P., Hoßbach, J., Schuldhaus, D., & Eskofier, B. M. (**2015**). Sensor-based stroke detection and stroke type classification in table tennis. Proceedings of the 2015 **ACM** International Symposium on Wearable Computers - ISWC '15.

k-NN Algorithm

Light Sport Exercise Detection Based on Smartwatch and Smartphone using k-Nearest Neighbor and Dynamic Time Warping Algorithm

- They proposes a light sport exercise activity detection system.
- They used **k-Nearest Neighbor algorithm.**
- Result: On the value of k =3, the accuracy of push up motion is 76.67%, then 80% for sit up, and 96.67% for squat jump activity.

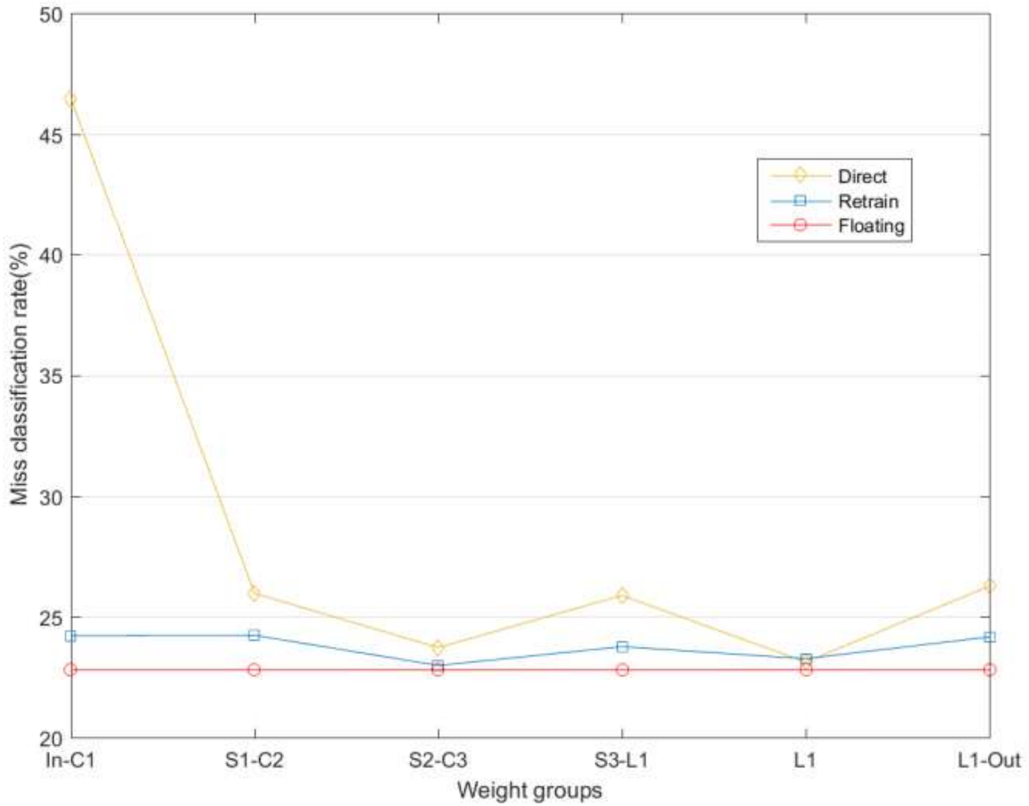
Table 2. Result of Iteration 3 Data Training Process

Motion	Parameter	k=1 (%)	k=3 (%)	k=5 (%)	k=7 (%)
Push Up	Sensitivity	100	100	100	100
	Specificity	66.67	83.33	83.33	66.67
	Accuracy	77.78	88.89	88.89	77.78
Sit Up	Sensitivity	66.67	100	33.33	33.33
	Specificity	100	100	100	100
	Accuracy	88.89	100	77.78	77.78
Squat Jump	Sensitivity	66.67	66.67	100	100
	Specificity	100	100	83.33	100
	Accuracy	88.89	88.89	88.89	100

Nurwanto, F., Ardiyanto, I., Wibirama, S., 2016. Light sport exercise detection based on smartwatch and smartphone using k-nearest neighborand dynamic time warping algorithm, pp.

RNN Algorithm

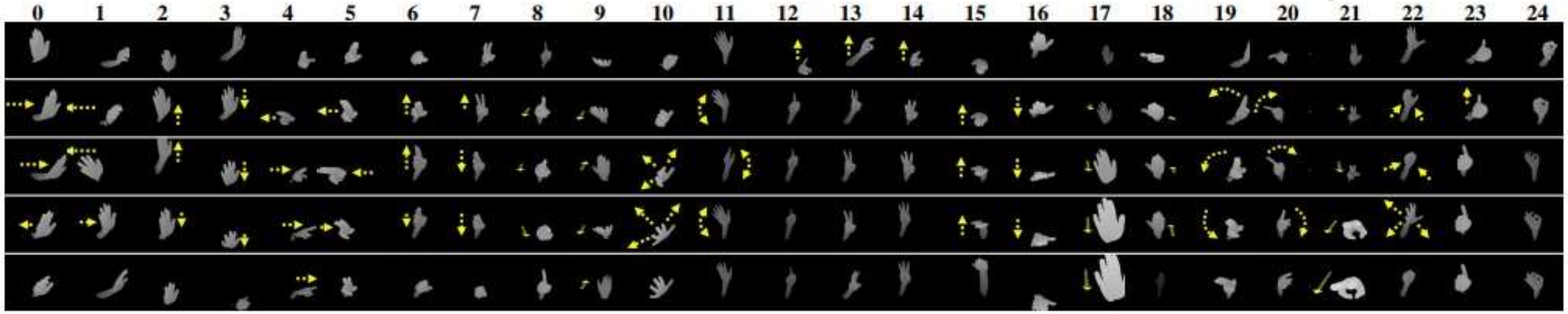
- discuss the importance to develop two dynamic hand gesture.
- The system was based on 2 methods. One is based on video signal and employs a combined structure and the other uses accelerometer data.
- By the optimization made, the required memory space for weights is reduced to 6.25% compared to floating-point implementations.



Shin, Sungho & Sung, Wonyong. (2016). Dynamic Hand Gesture Recognition for Wearable Devices with Low Complexity Recurrent Neural Networks. 10.1109/ISCAS.2016.7539037.

CNN Algorithm

- Supports online gesture classification with zero or negative lag.
- Performs simultaneous detection and classification of dynamic hand gestures from multi-modal data.
- System achieves an average accuracy 98.2%.



Twenty-five dynamic hand gesture classes extracted from either commercial systems or popular datasets.

Naïve Bayes Algorithm

- Discuss mainly the basics of hand gesture recognition as it is used daily in our lives.
- introduce a method that can distinguish various static hand movements in a complex background environment.
- Used the Naïve Bayes classifier and Gabor filter.
- overall accuracy reached was of over 90%.

TABLE I. COMPARISON OF RESULTS

Gestures	**Comparison of various techniques**		
	Nawazish et.al (2013)	***Naïve Bayes***	***Bayesian Classifier***
Stop	89.0	90.12	90.53
Pointing	85.2	87.20	88.01
Self-Pointing	87.34	89.01	89.85
Drinking	90.05	90.12	91.01
Take Care	89.66	90.00	91.00
…..	…..	…..	…..
Total	88.25	89.29	90.08

Ashfaq, Tahira & Khurshid, Khurram. (2016). Classification of Hand Gestures Using Gabor Filter with Bayesian and Naïve Bayes Classifier. International Journal of Advanced Computer Science and Applications. 7. 10.14569/IJACSA.2016.070340.

Random Forest Algorithm

- Introduce a methodology for Automatic hand motion recognition.
- They worked on ASL dataset (in-house dataset containing the 24 static letters of the alphabet).
- There experiments are encouraging with a classification rate of 98.36%.

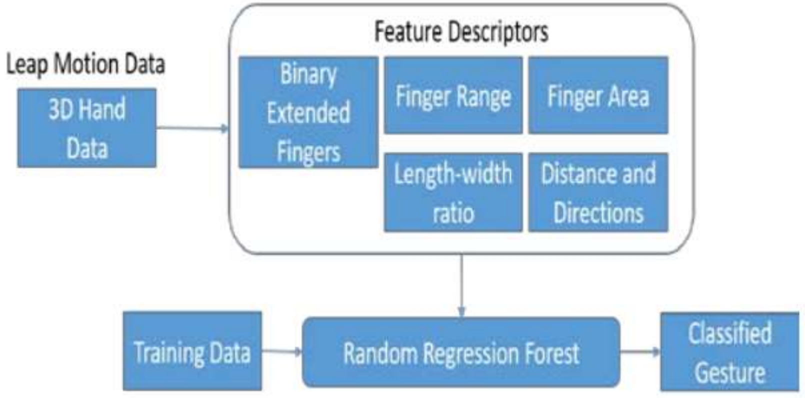


Figure 1. Proposed gesture recognition overview.

Table 2. Listing of Leap features from ASL dataset.

Feature	Data Type	Feature type
Extended fingers	Binary	Fingers (5)
Finger directions	3D vector	Fingers (5)
Fingertip positions	3D vector	Fingers (5)
Extended fingertip positions	3D vector	Fingers (5)
Hand direction	3D vector	Hand
Palm normal	3D vector	Hand
Palm Position	3D vector	Hand
Number of fingers	Unsigned	Range (1-5)

S. Canavan, W. Keyes, R. Mccormick, J. Kunnumpurath, T. Hoelzel and L. Yin, "Hand gesture recognition using a skeleton-based feature representation with a random regression forest," *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, 2017, pp. 2364-2368.

Logistic Regression Algorithm

- identify the user's hand and finger movements by the usage of smart watches.
- They use naïve bayes, logistic regression, and decision trees in their classification.
- All the classifiers show good results with 100% accuracy for detecting the arm
- Logistic regression was the best classifier with 99.20% and 97.10% for detecting finger and hand movements sequentially.

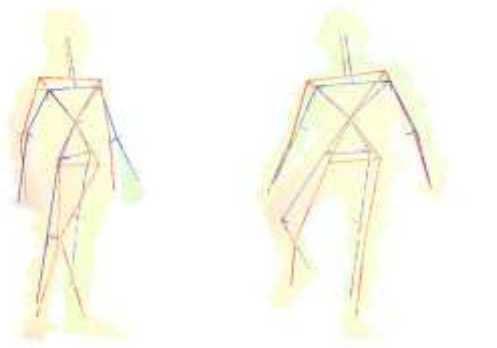
Classifier	TP Rate Max.	TP Rate Min.	TP Rate Avg.	Top 3 Misclassified
NB	100%	70.00%	90.00%	"D", "U", "W"
SL	100%	80.00%	94.62%	"D", "U", "R"
DT	100%	70.00%	88.08%	"D", "U", "A"

Table 7: Classification accuracy of recognizing finger-written alphabets and top three misclassified alphabets

Xu, Chao, Parth H. Pathak and Prasant Mohapatra. "Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition using Smartwatch." HotMobile '15 (2015).

usage of IR depth camera with motion sensor device.

- Device used: Kinect and wearable Internal sensors.
- Aim to address many of the well known limitations of the Kinect sensor.
- present a framework that allows the efficient fusion of these complementary data sources.
- Results in more accurate joint angle measurements.



Joint angle	Left knee flexion RMSE	Left knee flexion NCC	Right knee flexion RMSE	Right knee flexion NCC
Kinect L-Elbow	16.73°	0.13	9.93°	0.61
Fusion L-Elbow	**14.19°**	**0.70**	**3.81°**	**0.85**
Kinect R-Elbow	12.06°	0.41	10.34°	0.56
Fusion R-Elbow	**6.97°**	**0.89**	**5.12°**	**0.84**
Kinect L-Knee	29.51°	-0.63	26.94°	-0.02
Fusion L-Knee	**6.79°**	**0.73**	**8.98°**	**0.50**
Kinect R-Knee	9.82°	0.82	12.96°	0.80
Fusion R-Knee	**4.10°**	**0.99**	**5.86°**	**0.99**

- RMSE: root mean squared error values
- NCC: normalized cross correlation measure

Destelle, Francois & Ahmadi, Amin & O'Connor, Noel & Moran, Kieran & Chatzitofis, Anargyros & Zarpalas, Dimitrios & Daras, Petros. (2014). Low-cost accurate skeleton tracking based on fusion of kinect and wearable inertial sensors. European Signal Processing Conference.