



IMU al servicio del tenis:
Un estudio sobre la clasificación
golpes

Resumen

I. Introducción	
segundo. Contexto	4
III. Metodología	5
IV. Resultados	9
V. Discusión	10
VI. Conclusión	11
VII. Recursos	12
VIII.Bibliografía	13
IX. Anexidades	14

I. Introducción

A. Presentación del proyecto

Según Budi et al. (2020), el entrenamiento basado en bloques donde cada golpe se trabaja individualmente sería menos efectivo que practicar diferentes golpes al azar. Sin embargo, esto requiere una atención especial por parte del entrenador para que el jugador no Inconscientemente una mayoría de un tipo de disparo en detrimento de otros.

Para ello, teníamos el proyecto de utilizar una IMU (Unidad de Medición Inercial) para clasificar Diferentes tipos de tiros en el tenis. La IMU mide la aceleración lineal y la velocidad angular de los movimientos del jugador, lo que nos permite identificar las características únicas de cada tipo de movimiento. cortar. Hemos desarrollado una interfaz informática que permite recopilar y analizar datos en tiempo real y ofrece funcionalidades para visualizar los resultados y brindar asesoramiento a los jugadores.

Nuestro sistema utiliza datos IMU para clasificar diferentes tipos de tiros en tenis, incluidos Número de disparos realizados y conocer su distribución según tipo. Esto puede ayudar a los jugadores a comprender su juego e identificar áreas en las que pueden mejorar. En resumen, no El sistema proporciona a los jugadores información precisa sobre sus movimientos y rendimiento.

[Enlace de GitHub al final del informe](#)

B. Revisión de la literatura

Clasificar los tiros de tenis utilizando una IMU (unidad de medición inercial) es un tema interesante. En desarrollo.

Un artículo publicado en IEEE Xplore (Kos et al., 2016) presenta un trabajo relacionado con la detección y Clasificación de golpes de tenis utilizando un dispositivo IMU miniatura portátil colocado en el antebrazo del jugador. El sistema propuesto es capaz de detectar y clasificar los tres golpes de tenis común: derecha, revés y servicio. Información obtenida de los datos El acelerómetro se utiliza para la detección fiable y precisa de tiros de tenis, mientras que Se extrae y procesa información de los datos del giroscopio para la clasificación del tenis. A pesar de la memoria limitada y la falta de potencia de procesamiento, los algoritmos propuestos para la detección y clasificación de tiros pueden lograr una tasa de clasificación y una precisión de clasificación de tiros de tenis del 98%.

Según Wu et al. (2022), publicado en la revista Internet of Things, presenta un sistema basado en Internet de las cosas (IoT) para la evaluación del nivel de tenis en tiempo real y la clasificación de tiros.

El sistema utiliza una unidad de medición inercial (IMU) portátil y un dispositivo en miniatura para adquirir los movimientos del brazo del jugador. Los datos del acelerómetro y del giroscopio se utilizan para Detectar y clasificar los cinco golpes de tenis más comunes: derecha, revés y volea de derecha. La volea de revés y el saque. Algoritmos de aprendizaje automático como las máquinas vectoriales Se utilizan sistemas de visión artificial de apoyo (SVM) para identificar etiquetas de modelos para cada categoría y ca los nuevos conjuntos de datos. El sistema ha conseguido una precisión de clasificación del 100%. 99% de tenis usando SVM con Min-normalización

Se han realizado otros estudios, como el de Zhao et al. (2019), que propone un enfoque para estimar la Velocidad de una pelota de tenis mediante un sensor de movimiento montado en una raqueta. Los golpes El tenis se divide en tres categorías: saque, golpe de derecha y volea. Para un servicio, se propone un modelo de regresión para estimar la velocidad de la pelota. Para un golpe de derecha o una volea, dos modelos Se proponen: un modelo de regresión y un modelo físico. El modelo físico se utiliza para Estimación de la velocidad de la pelota para jugadores avanzados y modelo de regresión para jugadores avanzados principiantes.

segundo. Contexto

A. Presentación de la UMI

Una IMU (unidad de medición inercial) es un dispositivo electrónico que mide e informa datos de movimiento de un objeto utilizando una combinación de sensores como acelerómetros, giroscopios y magnetómetros. Estos sensores permiten a la IMU medir la aceleración lineal, la velocidad angular y la orientación de un objeto en el espacio. Las IMU de la marca Xsens están diseñadas para proporcionar mediciones de alta precisión y alta confiabilidad.

Los sensores están integrados en carcasas robustas, lo que proporciona protección contra golpes y vibraciones. Las IMU Xsens también están equipadas con filtros para minimizar los errores de medición y mejorar la estabilidad del sistema. También vienen con funciones avanzadas como compensación de deriva y calibración automática, lo que los hace ideales para ciertas aplicaciones.

Las IMU Xsens también son configurables: las frecuencias de muestreo, los rangos de medición y los filtros se pueden personalizar para satisfacer diferentes necesidades. Por último, Xsens también dispone de una aplicación (Xsens DOT) que permite una conexión rápida a la IMU y la visualización en tiempo real del acelerómetro y el giroscopio.

Para este proyecto simplemente mantendremos los datos de aceleración en m/s^2 así como los datos del giroscopio en grados/s. En modo Bluetooth, la frecuencia de adquisición del sensor es de 60 Hz. Una SD proporcionada por Xsens permite el desarrollo de herramientas que se comunican directamente con el sensor. En nuestra aplicación, los datos se recopilan a través de Bluetooth. Para ello, hemos adaptado los ejemplos proporcionados en el SDK de Xsens. En general, el sistema Bluetooth funciona con una función de devolución de llamada que recupera paquetes de información con una frecuencia de 60 Hz. La documentación completa está disponible en el sitio web de Xsens DOT.

B. Uso de IMU en la clasificación de tiros de tenis

Según el artículo de Ebner y Findling (2019), el posicionamiento de la IMU es importante porque puede afectar la precisión de los datos recopilados. Los autores encontraron que a pesar de las diferencias fisiológicas entre la muñeca y la raqueta como posición del sensor, los resultados indican rendimientos similares en la detección de tiros (98,5% - 99,5%) y en la clasificación dependiente e independiente del usuario (89% - 99%). En otras palabras, este estudio muestra que posicionar la IMU en la muñeca o en la raqueta puede ser efectivo para detectar y clasificar diferentes tipos de tiros en el tenis. Sin embargo, es importante señalar que este estudio se realizó en un entorno controlado y que otros factores pueden influir en los resultados en condiciones del mundo real.

C. Población interesada

Tuvimos la suerte de recuperar un conjunto de datos que ya estaba etiquetado en Internet, lo que tiene ventajas y desventajas. Una de las ventajas importantes es el tiempo que se ahorra en la creación del conjunto de datos, así como la precisión con la que se obtiene el mismo, que nunca habríamos podido conseguir con nuestros propios equipos. Sin embargo, el mayor inconveniente es que no conocemos los sujetos que crearon el conjunto de datos ni su nivel de competencia. La forma en que juega un jugador principiante o experto puede afectar en gran medida al conjunto de datos, que puede no ser adecuado para una población más grande. Según nuestras pruebas, parece que los datos son relativamente consistentes con los jugadores de tenis clasificados en el puesto 30. Además, observamos que el conjunto de datos fue creado por jugadores diestros, lo que puede tener un impacto, ya que un golpe de derecha realizado por un jugador diestro no coincidirá exactamente con un golpe de revés realizado por un jugador zurdo.

III. Metodología

A. Recopilación de datos de la UMI

Primero, recuperamos 2000 tiros de tenis de los datos del artículo de Z. y otros. Estas tomas se realizaron en condiciones controladas. En segundo lugar, por falta de disponibilidad del equipo adecuado para realizar la adquisición de datos en condiciones de partido, recibimos Datos sobre nosotros mismos y sobre los jugadores en entrenamiento. La primera persona era una mujer de nivel principiante avanzado y el segundo era un hombre de 30 años. Usamos una Aplicación para grabar tomas, creando así varias grabaciones de 7 a 8 tomas para un total de aproximadamente 200 disparos, cada uno de ellos etiquetado. Luego en nuestro programa tenemos Datos aislados de aceleración y giroscopio

B. Análisis de datos de la UMI

Una vez recopilados los datos, fue necesario encontrar un método para separarlos y disparos con el fin de clasificarlos. Para ello, salimos al campo y observamos diferentes variedades. que nos pareció más relevante para cada toma. Tras estas pruebas, pudimos deducir que la aceleración en el eje X (en nuestro posicionamiento IMU) sería una elección inteligente para iso Cada disparo. Entonces usaremos una función para encontrar los picos en este eje con Altura mínima adaptada al jugador practicante, con el fin de no establecer un umbral demasiado alto para un juego. principiante.

Además, seguimos a los creadores del conjunto de datos en el intervalo que permite aislar un resultado, es decir, 0, segundo antes y 0,5 segundos después. Este intervalo nos provocó un error al guardar los archivos. Si el pico está dentro de los límites de la grabación a una distancia menor que el intervalo que lo usamos, provoca un mal funcionamiento. Para solucionar este problema fue necesario para agregar una condición previa que verifique que la distancia entre el inicio y el primer pico, así como El último pico y final es mayor que el intervalo de análisis. Si no se cumple esta condición, el intervalo se reducirá proporcionalmente para continuar con la clasificación.

Figura 1: Umbral de detección

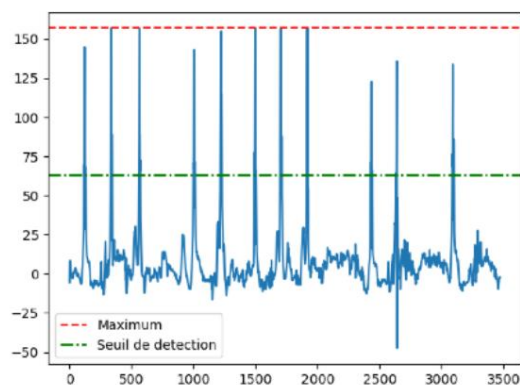
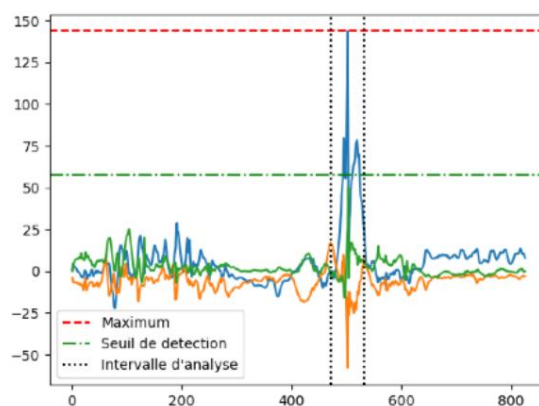


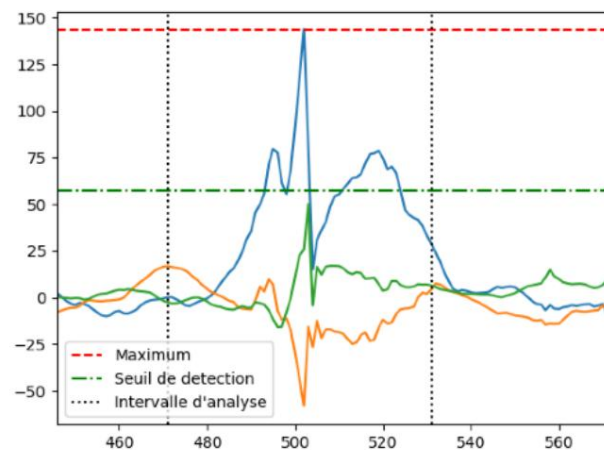
Figura 2: Intervalo de aislamiento del c permitiendo el cálculo de los diferentes parámetros



Como podemos ver en el gráfico, solo los picos por encima de la línea verde se considerarán válidos.

Gráfico 3: Zoom sobre el intervalo utilizado en el cálculo de parámetros

Una vez aislada cada toma, tuvimos que encontrar diferentes parámetros para atribuirle. Cada curva de aceleración y rotación. Tomamos un ejemplo de la literatura para obtener 36 variables. Para cada curva, utilizaremos la media, desviación estándar, mínimo, máximo, asimetría (que mide el grado de asimetría de la distribución en relación con su media) y curtosis (que mide la "agudeza" o planitud de la distribución en comparación con una distribución normal).



Nos enfrentamos a diferentes métodos de aprendizaje automático que permiten la clasificación. Rápidamente se nos presentaron tres soluciones: la red neuronal, el bosque aleatorio y las redes neuronales se pueden utilizar para clasificar los golpes de tenis a partir de datos IMU. Aprendiendo patrones complejos en datos. Las ventajas de las redes neuronales incluyen: su capacidad para aprender patrones no lineales y producir resultados precisos. Sin embargo, la Entrenar una red neuronal puede consumir mucho tiempo y recursos, y La interpretación de los resultados puede ser difícil.

Random Forest es un método eficaz para clasificar los tiros de tenis a partir de datos procesando múltiples variables simultáneamente. Las ventajas de Random Forest incluyen alta precisión, velocidad de entrenamiento y una interpretación relativamente fácil de los resultados. Sin embargo, Es posible que el modelo no funcione bien con datos altamente correlacionados y puede resultar difícil. Comprender cómo las variables contribuyeron a la clasificación. Los SVM también se pueden utilizar para clasificar golpes de tenis a partir de datos IMU procesamiento de datos no lineales. Las ventajas de SVM incluyen alta precisión y eficiencia. Alta con conjuntos de datos de tamaño mediano y fácil interpretación de los resultados. Sin embargo, las SVM pueden ser sensibles a los parámetros y el entrenamiento de un modelo puede ser lento para conjuntos de datos grandes. Elegimos el bosque aleatorio porque nos pareció el método más adecuado y fácil de implementar.

El funcionamiento de Random Forest es bastante sencillo. En primer lugar, el algoritmo construye una gran Número de árboles de decisión (o bosques) que utilizan muestras aleatorias de datos de entrada. Luego, cuando el algoritmo recibe una nueva entrada, pasa estos datos a cada uno árbol de decisión forestal. Cada árbol de decisión devuelve una predicción (por ejemplo, una clase) para un problema de clasificación o un valor numérico para un problema de regresión) para nuevos datos.

Finalmente, la predicción final del Bosque Aleatorio se calcula combinando las predicciones de todos los árboles de decisión. Para un problema de clasificación, la predicción final es la predicción de clase más frecuente. árboles de decisión, mientras que para un problema de regresión la predicción final es el promedio de los valores predichos por los árboles de decisión. Una de las características importantes de Random Forest es que utiliza un mecanismo de votación combinar las predicciones de diferentes árboles de decisión. Esto significa que incluso si algunos árboles

Las decisiones son menos precisas que otras, pero su contribución se tendrá en cuenta de todos modos.
predicción final

La implementación del bosque aleatorio se realizará a través de la biblioteca scikit-learn que permite una gran versatilidad en Python. Luego pudimos explorar los diferentes parámetros que ofrece la función.

para optimizar el algoritmo para nuestro conjunto de datos. Luego utilizamos otra función de scikit-learn:

GridSearchCV que nos permite probar las combinaciones que proponemos para conservar la mejor. Así que analizamos qué parámetros se podrían cambiar y descubrimos que...

7 parámetros:

- n_estimators : este es el número de árboles de decisión a construir en el bosque. Un valor alto puede mejorar la precisión del modelo, pero también puede ralentizar el tiempo de cálculo o provocar un sobreaprendizaje
- criterio : es la función para medir la calidad de la división de los nodos del árbol de decisión
El parámetro "gini" utiliza el índice de Gini para medir la pureza de los nodos y la "entropía" utiliza entropía.
- max_depth : esta es la profundidad máxima de cada árbol de decisión en el bosque. Un valor más alto puede aumentar la precisión del modelo, pero también puede hacerlo más propenso al sobreajuste.
- min_samples_split : este es el número mínimo de muestras necesarias para dividir un nodo interno
El árbol de decisiones. Un valor más alto puede hacer que el bosque sea más resistente a los datos ruidosos, pero también puede reducir la precisión del modelo.
- max_features : este es el número máximo de variables de entrada a considerar al encontrar la mejor división para cada nodo. Un valor más alto puede aumentar la precisión del modelo, pero también puede aumentar la complejidad del bosque y hacer que el tiempo de cálculo sea más largo.
- bootstrap : esto es si se debe o no utilizar muestreo con reemplazo al compilar de cada árbol de decisión. Un valor de "Verdadero" significa que se utiliza el muestreo con reemplazo y "Falso" significa que se utiliza el muestreo sin reemplazo - min_samples_leaf : Este parámetro determina el número mínimo de muestras que debe haber en la hoja.
en una hoja del árbol de decisión para que esta hoja se considere como una hoja final

Después de la optimización de parámetros, obtuvimos los siguientes parámetros:

- | | |
|-----------------------------------|---------------------------|
| - n_estimadores = 2 | - min_muestras_hoja = 1 |
| - criterio = entropía | - max_caracteristicas = 8 |
| - profundidad máxima = 12 | - bootstrap = Falso |
| - división mínima de muestras = 4 | |

Sin embargo, tras varias pruebas personales, preferimos volver al criterio por defecto, es decir "gini".

Por último, ver si la incorporación de valores del terreno a nuestro modelo de entrenamiento permite

Para mejorar nuestra precisión, incorporamos 68 resultados al conjunto de datos de referencia que tuvimos que etiquetar.

C. Visualización de datos

El objetivo de esta pequeña aplicación es simple: poder grabar secuencias de juego sencillas. permitir la restitución visual de los resultados de la predicción del movimiento. Las necesidades se traducen en una interfaz clara y sencilla, una conexión rápida a los sensores y una presentación visualmente rápida. resultados efectivos

Esta aplicación tiene cuatro características principales:

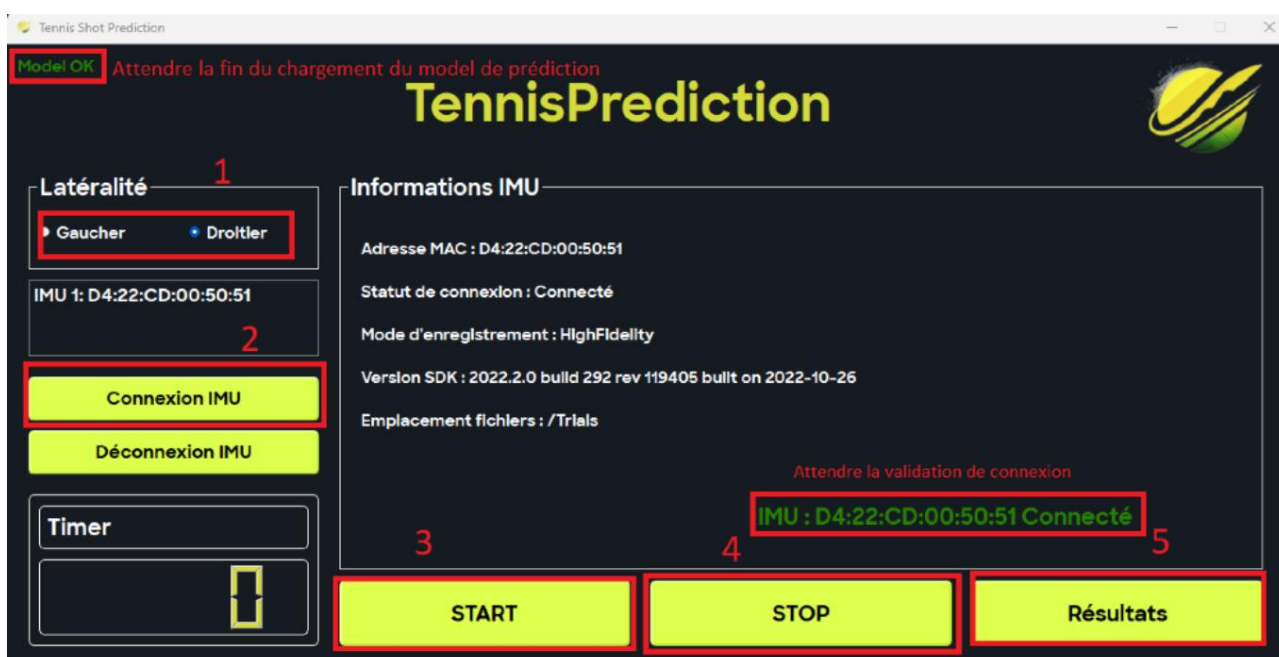
- Una funcionalidad de inicio y cierre de sesión en el XsensDot deseado
- Un cronómetro que te da una idea del tiempo de grabación.
- Una función de grabación en vivo de los datos recopilados por el sensor.
- Posibilidad de visualizar los resultados de la predicción en forma de gráfico.

En cuanto al diseño de la interfaz de usuario, este último Se realizó utilizando la versión Python compatible de la biblioteca Qt (pyQt). La elección de una interfaz sencilla y eficiente nos permite Parecía obvio.



Toda la aplicación fue desarrollada utilizando el lenguaje Python, dominado por todos los miembros de grupo, por eso lo elegimos para este proyecto. También sigue el paradigma de programación orientada a objetos, que es esencial para el uso de la biblioteca.

El código fuente de la aplicación está disponible en las respectivas páginas de GitHub de los miembros del proyecto. Tú un enlace al repositorio de GitHub: <https://github.com/VictorVatt/ShotPredict>



Finalmente, el funcionamiento de la aplicación que está disponible en varias etapas

1. La selección de la lateralidad del jugador que es útil para la predicción.
2. Conexión de la IMU a la aplicación. Tenga cuidado de esperar la actualización del st Conexión a "Conectado"
3. Iniciar la grabación 4. Detener la grabación
5. Ver los resultados de la predicción sobre los datos registrados previamente

(asegúrese de esperar a que el modelo de predicción termine de cargarse, en la parte superior izquierda)

IV. Resultados

Tabla 1: Resultados de la clasificación

	Derechazo		Contrarrestar		Servicio		Válido	Total	Disparos no detectados
	V	yo	V	yo	V	yo			
Total	90	104	36	41	28	36	154	181	8
% éxito	92,60%		87,80%		77,78%		85,08%		95,58%

V= Válido, T= Total

Para evaluar nuestro algoritmo de predicción, realizamos un conjunto de 181 movimientos, incluidos 104 golpes de derecha, 41 golpes de revés y 36 servicios. Todas las imágenes fueron recogidas de dos sujetos, un hombre y una mujer. Clasificamos el número de movimientos válidos en relación con el número total de movimientos para para realizar el análisis estadístico y agregar los resultados no detectados (ver Anne

Tabla 2: Corrección de errores con otro modelo

	Derechazo	Contrarrestar	Servicio	Total
Error	14	5	8	27
Resuelto	2	-1	3	4
% Resolución	14,3%	-20,0%	37,5%	14.8

Buscamos comparar errores de detección utilizando un modelo que agrupa los datos Zhao et al con nuestros datos de campo.

V. Discusión

A. Análisis de resultados

Los resultados obtenidos son satisfactorios, con una precisión del 92,6% para los golpes de derecha y del 87,8% para los de derecha. retrocesos y 77,78% para servicios. Estos resultados muestran la eficacia del algoritmo de clasificación. que hemos desarrollado para detectar diferentes tipos de golpes

Sin embargo, la precisión del 77,78% para los servicios sugiere que aún es necesario realizar mejoras.

Diseñado para aumentar la precisión general. El análisis de los resultados mostró que la clasificación de los servicios fue más difícil que la de los golpes de derecha y de revés, debido a la complejidad de la técnica del servicio. Esto sugiere que el algoritmo necesita ser optimizado para identificar mejor las diferencias.

Variaciones de la técnica del servicio, con el fin de mejorar su precisión.

Además, es importante señalar que el 4,5% de los resultados no pudieron ser detectados por el algoritmo. Aunque se trata de una tasa relativamente baja, significa que es posible que el algoritmo no pueda

Detectar todos los resultados, lo que afecta la precisión general de la clasificación.

Finalmente, agregar nuestros datos de campo a los datos de Zhao et al. ayuda a reducir la cantidad de errores. incluso si integramos un mínimo de disparos.

B. Relevancia y fiabilidad de los resultados

La alta tasa de éxito de los golpes de derecha, con un 92,6% de golpes detectados, hace que su clasificación sea p confiable. Los contratiempos, aunque menos numerosos, también obtienen una tasa de éxito satisfactoria.

Por otro lado, nuestro algoritmo tiene debilidades para la clasificación de servicios, con sólo

% de servicios clasificados. Esta baja tasa se puede explicar por las notables diferencias observadas entre los distintos tipos de servicio.

El número total de aciertos estimado es cercano a la realidad, con sólo un 4,5% de errores. Estos errores puede explicarse por una menor fuerza aplicada a estos disparos, lo que haría que pasaran por debajo del umbral de detección. Un umbral de detección más bajo produciría resultados falsos en otras ubicaciones, lo que

No sería aceptable si queremos mejorar nuestro sistema.

Finalmente, nuestras pruebas se realizaron con dos jugadores, por un total de menos de 200 movimientos, lo que puede resultar ser insuficiente para obtener resultados de la FIA

VI. Conclusión

A. Resumen de las principales conclusiones del proyecto

En conclusión, la clasificación de las tomas es satisfactoria, pero los resultados obtenidos pueden mejorarse aumentando el tamaño del conjunto de datos. Este último podría hacerse más general para poder clasificar los tiros para jugadores con diferentes niveles. Además de la clasificación, también es necesario perfeccionar la detección p para poder evaluar completamente el algoritmo de clasificación. El uso del factor de aceleración X como estándar podría hacer que la detección sea más eficiente.

Además, demostramos que la precisión se puede mejorar añadiendo nuevos datos a nuestro modelo de entrenamiento. Al incorporar más variabilidad a nuestro algoritmo, pudimos aumentar la precisión en la predicción del servicio en un 10%.

Sin embargo, la principal limitación de nuestro proyecto es que no conocemos el nivel de los jugadores seleccionados en nuestro conjunto de datos de referencia. En otras palabras, los patrones que asociamos con la aceleración y la velocidad angular pueden ser demasiado diferentes entre los jugadores que seguimos que aquellos en el estudio de Zhao et al. Para garantizar una mayor precisión, habría sido necesario acompañar a más jugadores en el campo, para limitar el impacto de los datos del estudio de Zhao e introducir más variabilidad con los nuestros.

B. Sugerencias para futuras investigaciones.

Con respecto a futuras investigaciones para mejorar la clasificación de los tiros de tenis, se podrían explorar tres líneas principales.

En primer lugar, el uso de otro algoritmo de clasificación, como una red neuronal, puede mejorar la precisión de la clasificación de los tiros de tenis. De hecho, las redes n son conocidas por su capacidad de procesar datos complejos y no lineales, lo que podría ser particularmente útil para procesar datos de movimiento de los jugadores de tenis.

Luego, se podría desarrollar una aplicación de visualización en tiempo real para permitir a los jugadores de tenis seguir su rendimiento durante los partidos. Esta aplicación podría utilizar sensores portátiles para recopilar datos de movimiento de los jugadores y mostrar información en tiempo real sobre el tipo y número de tiros. Los jugadores podrían utilizar esta información para ajustar su estrategia de juego en tiempo real y mejorar su rendimiento.

Por último, otra vía de investigación interesante sería adaptar el algoritmo de clasificación a los deportes de raqueta, como el bádminton o el squash. Esto permitiría el desarrollo de un método de clasificación más general que podría aplicarse a diferentes deportes y abriría nuevas perspectivas en términos de análisis del rendimiento de los jugadores en estos deportes. En resumen, estas tres ideas podrían contribuir a mejorar la precisión y aplicabilidad de la clasificación de los tiros de tenis, abriendo al mismo tiempo nuevas perspectivas para la investigación de los deportes de raqueta en general.

VII. Recursos

Todo el código utilizado para nuestro proyecto está disponible en GitHub a través del siguiente enlace <https://github.com/VictorVatt/ShotPredictio>

Allí encontrará el código fuente, el conjunto de datos utilizado para entrenar el modelo, así como los archivos que usamos para evaluar nuestro modelo. Además, se proporcionará una guía de uso de la aplicación. disponible.

VIII. Bibliografía

- Budi, DR, Syafei, MM, Kusuma, MNH, Suhartoyo, T., Hidayat, R. y Listiandi, AD (2020). La importancia del método de ejercicio en la mejora de las habilidades de golpe de derecha y revés en el tenis Revista deportiva: revista científica. https://doi.org/10.29407/js_unpgri.v6i1
- Ebner, CJ, y Findling, RD (2019). Clasificación de golpes de tenis. Avances en multimedia móvil. doi.org/10.1145/3365921.3365929
- Fabián Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay; 12(85):2825–2830, 2 • Kos, M., Zenko, J., Vlaj, D. y Kramberger, I. (2016).
Detección y clasificación de golpes de tenis
Dispositivo IMU portátil en miniatura. Conferencia internacional sobre sistemas, señales y procesamiento de imágenes <https://doi.org/10.1109/iwssip.2016.75027>
- Wu, M., Fan, M., Hu, Y., Wang, R., Wang, Y., Li, Y., Wu, S. y Xia, G. (2022b). Un sistema de evaluación del nivel de tenis y clasificación de golpes en tiempo real basado en el Internet de las cosas. Internet de las cosas 100494. <https://doi.org/10.1016/j.iot.2021.1004> • Zhao, H., Wang, S., Zhou, G. y
Jung, W. (2019). Ojo de tenis. Procesamiento de información en redes Senso. <https://doi.org/10.1145/3302506.33104>

IX. Anexidades

Apéndice 1: Conjunto de resultados

	Coup droit		Revers		Service		Valide	Total	Coup non détecté
	V	T	V	T	V	T			
92212	7	7	4	4	0	0	11	11	0
92328	2	2	1	1	0	0	3	3	0
92420	2	2	1	1	0	0	3	3	0
92519	6	6	1	1	0	0	7	7	0
92640	1	2	0	0	0	0	1	2	1
92712	9	9	2	2	0	0	11	11	0
92903	5	5	1	1	0	0	6	6	0
93410	7	7	0	0	0	0	7	7	0
93512	6	6	3	3	0	0	9	9	0
93634	5	5	2	2	0	0	7	7	0
93813	3	3	0	0	0	0	3	3	0
94300	1	4	0	1	0	0	1	5	0
94659	0	0	0	0	0	2	0	2	0
94735	0	0	0	0	0	1	0	1	0
94800	0	0	0	0	3	6	3	6	0
95029	0	0	0	0	1	1	1	1	0
95137	4	5	0	0	0	0	4	5	1
95337	0	2	1	2	0	0	1	4	2
95524	4	4	1	1	1	2	6	7	0
101728	0	0	0	0	4	4	4	4	0
101827	0	0	0	0	4	4	4	4	0
101934	1	1	0	0	4	4	5	5	0
102111	1	1	0	0	0	0	1	1	0
102130	2	3	2	2	0	0	4	5	0
102234	5	5	2	2	0	0	7	7	0
102511	2	3	2	2	3	3	7	8	1
102622	0	0	2	2	3	3	5	5	0
102804	4	5	2	2	0	0	6	7	1
102858	2	3	1	1	0	0	3	4	1
102933	2	2	2	2	0	0	4	4	0
103003	2	2	1	2	0	0	3	4	0
103127	3	3	2	2	0	0	5	5	0
103205	2	2	1	1	0	0	3	3	0
103504	0	3	0	0	2	2	2	5	0
103553	0	0	0	1	2	3	2	4	0
103649	2	2	2	3	1	1	5	6	1
Total	90	104	36	41	28	36	154	181	8
% réussite	86,54 %		87,80 %		77,78 %		85,08 %		95,58 %