



L'IMU au service du tennis : Une étude sur la classification des coups

Sommaire

I. Introduction	3
II. Contexte	4
III. Méthodologie	5
IV. Résultats	9
V. Discussion	10
VI. Conclusion	11
VII. Ressources	12
VIII. Bibliographie	13
IX. Annexes	14

I. Introduction

A. Présentation du projet

D'après Budi et al. (2020), un entraînement basé sur des blocs où chaque coup est travaillé individuellement serait moins efficace que de pratiquer différents coups de manière aléatoire. Toutefois, cette méthode nécessite une attention particulière de la part de l'entraîneur afin que le joueur n'effectue pas inconsciemment une majorité d'un type de coup au détriment des autres.

Pour ce faire, nous avons eu comme projet d'utiliser un IMU (Inertial Measurement Unit) pour classifier les différents types de coups au tennis. L'IMU mesure l'accélération linéaire et la vitesse angulaire des mouvements du joueur, ce qui nous permet d'identifier les caractéristiques uniques de chaque type de coup. Nous avons développé une interface sur ordinateur qui permet de collecter et d'analyser les données en temps réel et offre des fonctionnalités pour visualiser les résultats et fournir des conseils aux joueurs.

Notre système utilise les données de l'IMU pour classifier les différents types de coups au tennis, compter le nombre de coups effectués et connaître leur répartition selon le type. Cela peut aider les joueurs à mieux comprendre leur jeu et à identifier les domaines dans lesquels ils peuvent s'améliorer. En somme, notre système fournit aux joueurs des informations précises sur leurs mouvements et leur performance.

Lien GitHub en fin de rapport

B. Revue de littérature

La classification des coups de tennis à l'aide d'un IMU (unité de mesure inertielle) est un sujet intéressant et en développement.

Un article publié dans IEEE Xplore (Kos et al., 2016) présente des travaux liés à la détection et à la classification des coups de tennis à l'aide d'un dispositif IMU miniature et portable positionné sur l'avant-bras du joueur. Le système proposé est capable de détecter et de classer les trois coups de tennis les plus courants : le coup droit, le revers et le service. Les informations obtenues à partir des données de l'accéléromètre sont utilisées pour une détection fiable et précise des coups de tennis, tandis que les informations provenant des données du gyroscope sont extraites et traitées pour la classification des coups de tennis. Malgré une mémoire limitée et un manque de puissance de traitement, les algorithmes proposés pour la détection et la classification des coups sont capables d'atteindre un taux de classification élevé, avec une précision de classification des coups de tennis de 98,1 %.

D'après Wu et al. (2022), publié dans la revue Internet of Things, est présenté un système basé sur l'Internet des Objets (IoT) pour l'évaluation en temps réel du niveau de tennis et la classification des coups. Le système utilise une unité de mesure inertielle (IMU) portable et un appareil miniature pour acquérir les mouvements du bras du joueur. Les données de l'accéléromètre et du gyroscope sont utilisées pour détecter et classer les cinq coups de tennis les plus courants : le coup droit, le revers, le coup droit volée, le revers volée et le service. Les algorithmes d'apprentissage automatique tels que les machines à vecteurs de support (SVM) sont utilisés pour identifier les étiquettes de modèle pour chaque catégorie et catégoriser les nouveaux ensembles de données. Le système a atteint une précision de classification des coups de tennis de 99 % en utilisant SVM avec normalisation Min-Max.

D'autres études ont été menées, telles que Zhao et al. (2019), qui propose une approche pour estimer la vitesse de la balle de tennis en utilisant un capteur de mouvement monté sur une raquette. Les coups de tennis sont divisés en trois catégories : service, coup droit et volée. Pour un service, un modèle de régression est proposé pour estimer la vitesse de la balle. Pour un coup droit ou une volée, deux modèles sont proposés : un modèle de régression et un modèle physique. Le modèle physique est utilisé pour estimer la vitesse de la balle pour les joueurs avancés et le modèle de régression pour les joueurs débutants.

II. Contexte

A. Présentation de l'IMU

Un IMU (Inertial Measurement Unit) est un dispositif électronique qui mesure et rapporte les données de mouvement d'un objet en utilisant une combinaison de capteurs tels que des accéléromètres, des gyroscopes et des magnétomètres. Ces capteurs permettent à l'IMU de mesurer l'accélération linéaire, la vitesse angulaire et l'orientation d'un objet dans l'espace.

Les IMU de la marque Xsens sont conçus pour offrir des mesures de haute précision et une grande fiabilité. Les capteurs sont intégrés dans des boîtiers robustes, offrant une protection contre les chocs et les vibrations. Les IMU Xsens sont également équipés de filtres pour minimiser les erreurs de mesure et améliorer la stabilité du système. Ils sont également dotés de fonctionnalités avancées telles que la compensation de la dérive et le calibrage automatique, ce qui les rend idéaux pour certaines applications.

Les IMU de Xsens sont également configurables : les fréquences d'échantillonnage, les gammes de mesure et les filtres peuvent être personnalisés pour répondre aux différents besoins. Enfin, Xsens a également mis en place une application (Xsens DOT) permettant une connexion rapide à l'IMU et une visualisation en temps réel de l'accéléromètre et du gyroscope.

Pour ce projet, nous garderons simplement les données d'accélération en m/s^2 ainsi que les données du gyroscope en degrés/s. En mode Bluetooth, la fréquence d'acquisition du capteur est de 60 Hz. Un SDK fourni par Xsens permet de développer des outils qui communiquent directement avec le capteur.

Dans notre application, les données sont récoltées via Bluetooth. Pour cela, nous avons adapté les exemples fournis dans le SDK Xsens. Globalement, le système Bluetooth fonctionne avec une fonction Callback qui récupère les paquets d'informations (packets) avec une fréquence de 60 Hz. La documentation complète est disponible sur le site Xsens DOT.

B. Utilisation de l'IMU dans la classification des coups au tennis

Selon l'article d'Ebner et Findling (2019), le positionnement de l'IMU est important car il peut affecter la précision des données collectées. Les auteurs ont constaté que malgré les différences physiologiques entre le poignet et la raquette en tant que position du capteur, les résultats indiquent des performances similaires dans la détection des coups (98,5 % - 99,5 %) et dans la classification dépendante et indépendante de l'utilisateur (89 % - 99 %).

En d'autres termes, cette étude montre que le positionnement de l'IMU au poignet ou à la raquette peut être efficace pour détecter et classer les différents types de coups au tennis. Cependant, il est important de noter que cette étude a été menée dans un environnement contrôlé et que d'autres facteurs peuvent influencer les résultats dans des conditions réelles.

C. Population concerné

Nous avons eu la chance de récupérer un jeu de données déjà labellisé sur internet, ce qui présente à la fois des avantages et des inconvénients. L'un des avantages importants est le temps économisé pour la création du dataset, ainsi que la précision avec laquelle il a été obtenu, que nous n'aurions jamais pu atteindre avec notre propre équipement. Toutefois, l'inconvénient majeur est que nous ne connaissons pas les sujets qui ont permis de réaliser le dataset, ni leur niveau de compétence. La façon dont un joueur débutant ou expert joue peut grandement affecter le dataset, qui peut ne pas convenir à une population plus large.

À partir des essais que nous avons effectués, il semble que les données correspondent relativement bien aux joueurs de tennis classés 30. De plus, nous notons que le dataset a été créé par des joueurs droitiers, ce qui peut avoir une incidence car un coup droit réalisé par un droitier ne correspondra pas exactement à un revers réalisé par un gaucher.

III. Méthodologie

A. Collecte des données de l'IMU

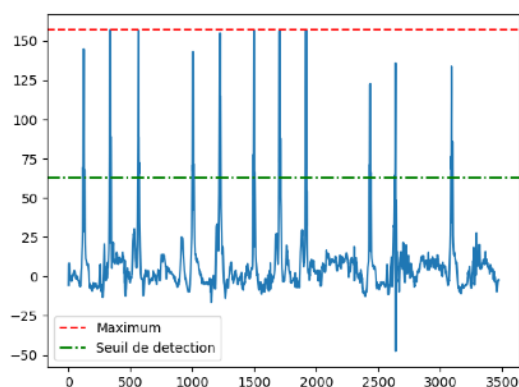
Dans un premier temps, nous avons récupéré 2000 coups de tennis à partir des données de l'article de Zhao et al. Ces coups ont été effectués dans des conditions contrôlées. Dans un second temps, faute de disposer du matériel adéquat pour effectuer l'acquisition de données en conditions de match, nous avons recueilli des données sur nous-mêmes ainsi que sur des joueurs en entraînement. La première personne était une femme de niveau débutant avancé, et la deuxième était un homme classé 30. Nous avons utilisé une application pour enregistrer les coups, créant ainsi plusieurs enregistrements de 7 à 8 coups pour un total d'environ 200 coups, avec une labellisation de chaque coup. Ensuite, dans notre programme, nous avons isolé les données d'accélération et de gyroscope.

B. Analyse des données de l'IMU

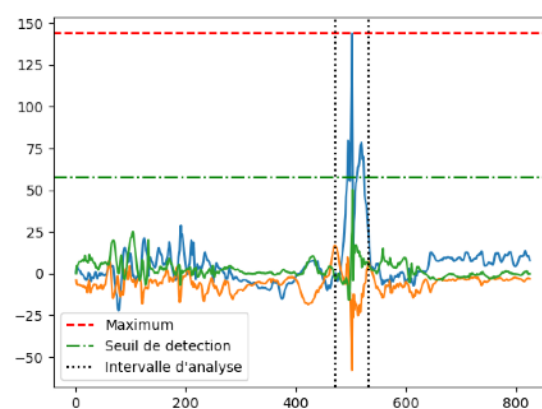
Une fois les données collectées, il a été nécessaire de trouver une méthode permettant de les séparer en coups afin de les classifier. Pour cela, nous sommes allés sur le terrain et avons observé différentes variables qui nous semblaient les plus pertinentes pour chaque coup. À l'issue de ces tests, nous avons pu en déduire que l'accélération sur l'axe X (dans notre positionnement d'IMU) serait un choix judicieux pour isoler chaque coup. Nous utiliserons donc une fonction permettant de trouver les pics sur cet axe avec une hauteur minimale adaptée au joueur pratiquant, afin de ne pas fixer un seuil trop important pour un joueur novice.

De plus, nous avons suivi les créateurs du dataset dans l'intervalle permettant d'isoler un coup, à savoir 0,5 seconde avant et 0,5 seconde après. Cet intervalle nous a causé une erreur lors de l'enregistrement des fichiers. Si le pic se trouve dans les limites de l'enregistrement à une distance inférieure à l'intervalle que nous utilisons, cela provoque un dysfonctionnement. Afin de résoudre ce problème, il a été nécessaire d'ajouter une condition préalable vérifiant que la distance entre le début et le premier pic ainsi que le dernier pic et la fin soit supérieure à l'intervalle d'analyse. Si cette condition n'est pas remplie, alors l'intervalle sera réduit en conséquence afin de poursuivre la classification.

Graphique 1 : Seuil de détection



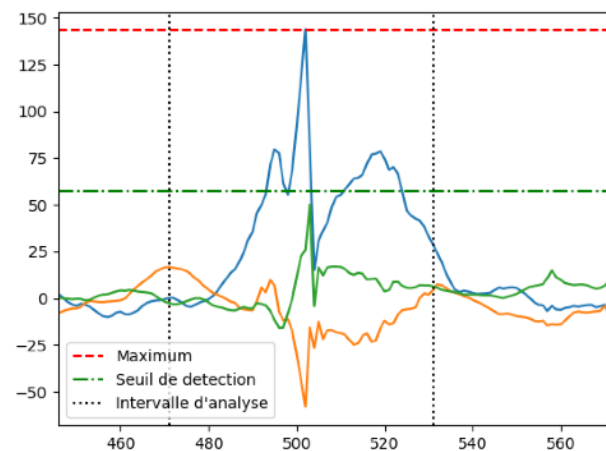
Graphique 2 : Intervalle d'isolation du coup permettant le calcul des différents paramètres



Comme nous pouvons le voir sur le graphique, seuls les pics au-dessus de la ligne verte seront considérés comme valides.

Graphique 3 : Zoom sur l'intervalle utilisé dans le calcul des paramètres

Une fois chaque coup isolé, il nous a fallu trouver différents paramètres à attribuer à chaque courbe d'accélération et de rotation. Nous avons repris un exemple de la littérature pour obtenir 36 variables. Pour chaque courbe, nous utiliserons la moyenne, l'écart type, le minimum, le maximum, l'asymétrie (qui mesure le degré de dissymétrie de la distribution par rapport à sa moyenne) et le kurtosis (qui mesure la "pointe" ou l'aplatissement de la distribution par rapport à une distribution normale).



Nous avons été confrontés à différentes méthodes de Machine Learning permettant la classification. Rapidement, trois solutions se sont offertes à nous : le réseau de neurones, le random forest et les SVM. Les réseaux de neurones peuvent être utilisés pour classer les coups de tennis à partir de données d'IMU en apprenant des motifs complexes dans les données. Les avantages des réseaux de neurones comprennent leur capacité à apprendre des motifs non linéaires et à produire des résultats précis. Cependant, la formation d'un réseau de neurones peut être longue et exigeante en termes de ressources, et l'interprétation des résultats peut être difficile.

Random Forest est une méthode efficace pour la classification de coups au tennis à partir de données d'IMU en traitant plusieurs variables simultanément. Les avantages de Random Forest incluent une précision élevée, une rapidité de formation et une interprétation relativement facile des résultats. Toutefois, le modèle peut ne pas bien fonctionner avec des données hautement corrélées, et il peut être difficile de comprendre comment les variables ont contribué à la classification.

Les SVM peuvent également être utilisés pour classer les coups au tennis à partir de données d'IMU en traitant des données non linéaires. Les avantages de SVM incluent une grande précision, une efficacité élevée avec des ensembles de données de taille moyenne, et une interprétation facile des résultats. Néanmoins, les SVM peuvent être sensibles aux paramètres, et l'entraînement d'un modèle peut être lent pour les ensembles de données volumineux.

Notre choix s'est porté sur le random forest qui nous paraissait la méthode la plus adaptée et la plus simple à mettre en place.

Le fonctionnement du Random Forest est assez simple. Tout d'abord, l'algorithme construit un grand nombre d'arbres de décision (ou des forêts) en utilisant des échantillons aléatoires de données d'entrée et des échantillons aléatoires de variables d'entrée pour chaque arbre.

Ensuite, lorsque l'algorithme reçoit une nouvelle donnée en entrée, il passe cette donnée dans chaque arbre de décision de la forêt. Chaque arbre de décision renvoie une prédiction (par exemple, une classe pour un problème de classification ou une valeur numérique pour un problème de régression) pour la nouvelle donnée.

Enfin, la prédiction finale du Random Forest est calculée en combinant les prédictions de tous les arbres de décision. Pour un problème de classification, la prédiction finale est la classe la plus fréquente prédite par les arbres de décision, tandis que pour un problème de régression, la prédiction finale est la moyenne des valeurs prédites par les arbres de décision.

L'une des caractéristiques importantes du Random Forest est qu'il utilise un mécanisme de vote pour combiner les prédictions des différents arbres de décision. Cela signifie que même si certains arbres de

décision sont moins précis que d'autres, leur contribution sera tout de même prise en compte dans la prédiction finale.

La mise en place du random forest se fera via la bibliothèque scikit-learn qui permet une grande polyvalence en Python. Nous avons alors pu explorer les différents paramètres offerts par la fonction afin d'optimiser l'algorithme pour notre jeu de données. Nous avons ensuite utilisé une autre fonction offerte par scikit-learn : GridSearchCV qui permet de tester les combinaisons que nous lui proposons pour retenir la meilleure. Nous avons donc cherché quels paramètres pourraient être modifiés, et nous nous sommes fixé 7 paramètres :

- **n_estimators** : c'est le nombre d'arbres de décision à construire dans la forêt. Une valeur élevée peut améliorer la précision du modèle, mais peut également ralentir le temps de calcul ou provoquer un surapprentissage.
- **criterion** : il s'agit de la fonction de mesure de la qualité de la division des nœuds de l'arbre de décision. Le paramètre "gini" utilise l'indice de Gini pour mesurer la pureté des nœuds et "entropy" utilise l'entropie.
- **max_depth** : c'est la profondeur maximale de chaque arbre de décision dans la forêt. Une valeur plus élevée peut augmenter la précision du modèle, mais peut également le rendre plus sujet au surapprentissage.
- **min_samples_split** : c'est le nombre minimum d'échantillons requis pour diviser un nœud interne de l'arbre de décision. Une valeur plus élevée peut rendre la forêt plus robuste aux données bruyantes, mais peut également réduire la précision du modèle.
- **max_features** : c'est le nombre maximum de variables d'entrée à considérer lors de la recherche de la meilleure division pour chaque nœud. Une valeur plus élevée peut augmenter la précision du modèle, mais peut également augmenter la complexité de la forêt et rendre le temps de calcul plus long.
- **bootstrap** : il s'agit de l'utilisation ou non de l'échantillonnage avec remplacement lors de la construction de chaque arbre de décision. Une valeur de "True" signifie que l'échantillonnage avec remplacement est utilisé et "False" signifie que l'échantillonnage sans remplacement est utilisé.
- **min_samples_leaf** : ce paramètre détermine le nombre minimum d'échantillons qui doivent se trouver dans une feuille de l'arbre de décision pour que cette feuille soit considérée comme une feuille finale.

A l'issue de l'optimisation des paramètres, nous avons obtenu les paramètres suivants :

- | | |
|-------------------------|------------------------|
| - n_estimators = 250 | - min_samples_leaf = 1 |
| - criterion = entropy | - max_features = 8 |
| - max_depth = 12 | - bootstrap = False |
| - min_samples_split = 4 | |

Cependant, suite à différents tests personnels, nous avons préféré revenir sur le criterion par défaut, à savoir « gini ».

Enfin, pour voir si le fait d'intégrer des valeurs de terrain à notre modèle d'entraînement permettrait d'améliorer notre précision, nous avons intégré 68 coups au dataset de référence que nous avons dû étiqueter.

C. Visualisation des données

L'objectif de cette petite application est simple : pouvoir enregistrer des séquences de jeu facilement et permettre la restitution visuelle des résultats de la prédiction de coups. Les besoins se traduisent par une interface claire et simple, une connexion rapide aux capteurs et une présentation visuellement rapide et efficace des résultats.

Cette application possède quatre principales fonctionnalités :

- Une fonctionnalité de connexion / déconnexion au XsensDot souhaité
- Un chronomètre qui permet d'avoir une idée du temps d'enregistrement.
- Une fonctionnalité d'enregistrement en direct des données collectées par le capteur
- Une possibilité de visualiser les résultats de la prédiction sous forme de graphiques

Concernant la conceptions de l'interface utilisateur, cette dernière a été réalisée à l'aide de la version prise en charge en python de la librairie Qt (pyQt). Le choix d'une interface simple et efficace nous a semblé évident.



L'ensemble de l'application a été développé avec le langage Python, maîtrisé par tous les membres du groupe, c'est pourquoi nous l'avons choisi pour ce projet. Elle respecte également le paradigme de la programmation orientée objet, qui est indispensable pour utiliser la bibliothèque Qt.

Le code source de l'application est disponible sur les pages GitHub respective des membres du projet. Voici un lien vers le repository GitHub : <https://github.com/VictorVatt/ShotPredictionApp>



Enfin le fonctionnement de l'application qui se décline en plusieurs étapes :

1. La sélection de la latéralité du joueur qui est utile pour la prédiction
2. La connexion de l'IMU à l'application. Attention à bien attendre la mise à jour du statut de connexion à « Connecté »
3. Le lancement de l'enregistrement
4. L'arrêt de l'enregistrement
5. La visualisation des résultats de la prédiction sur les données enregistrées précédemment.
(bien attendre la fin du chargement du model de prédiction, en haut à gauche)

IV. Résultats

Tableau 1 : Résultats classification

	Coup droit		Revers		Service		Valide	Total	Coups non détecté
	V	T	V	T	V	T			
Total	90	104	36	41	28	36	154	181	8
% réussite	92,60 %		87,80 %		77,78 %		85,08 %		95,58 %

V= Valide, T= Total

Afin d'évaluer notre algorithme de prédiction, nous avons effectué un ensemble de 181 coups, comprenant 104 coups droits, 41 revers et 36 services. L'ensemble des coups a été collecté sur 2 sujets, un homme et une femme. Nous avons classé le nombre de coups valides par rapport au nombre de coups total afin d'effectuer l'analyse statistique et y avons ajouté les coups non détectés (voir Annexe 1).

Tableau 2 : Correction des erreurs avec un autre modèle

	Coup droit	Revers	Service	Total
Erreur	14	5	8	27
Résolue	2	-1	3	4
% Résolution	14,3 %	-20,0 %	37,5 %	14,8 %

Nous avons cherché à comparer les erreurs de détection en utilisant un modèle regroupant les données de Zhao et al avec nos données de terrain.

V. Discussion

A. Analyse des résultats

Les résultats obtenus sont satisfaisants, avec une précision de 92,6 % pour les coups droits, de 87,8 % pour les revers et de 77,78 % pour les services. Ces résultats montrent l'efficacité de l'algorithme de classification que nous avons développé pour détecter les différents types de coups.

Cependant, la précision de 77,78 % pour les services suggère que des améliorations doivent encore être apportées pour augmenter la précision globale. L'analyse des résultats a montré que la classification des services était plus difficile que celle des coups droits et des revers, en raison de la complexité de la technique de service. Cela suggère que l'algorithme doit être optimisé pour mieux identifier les différentes variations de la technique de service, afin d'améliorer sa précision.

En outre, il est important de noter que 4,5 % des coups n'ont pas pu être détectés par l'algorithme. Bien que ce soit un taux relativement faible, cela signifie que l'algorithme peut ne pas être en mesure de détecter tous les coups, ce qui affecte la précision globale de la classification.

Enfin, l'ajout de nos données de terrain aux données de Zhao et al permet de réduire le nombre d'erreurs, même si nous intégrons un minimum de coups.

B. Pertinence et fiabilité des résultats

Le taux de réussite élevé pour les coups droits, avec 92,6 % de coups détectés, rend leur classification plus fiable. Les revers, bien qu'ils soient moins nombreux, obtiennent également un taux de réussite satisfaisant. En revanche, notre algorithme présente des faiblesses pour la classification des services, avec seulement 25 % des services classifiés. Ce faible taux peut s'expliquer par les différences notables observées entre les différents types de services.

Le nombre total de coups estimés se rapproche de la réalité, avec seulement 4,5 % d'erreurs. Ces erreurs peuvent être expliquées par une moindre force appliquée à ces coups, qui les ferait passer en dessous du seuil de détection. Un seuil de détection plus faible produirait des faux coups à d'autres endroits, ce qui ne serait pas acceptable si l'on souhaite améliorer notre système.

Enfin, nos essais ont été menés sur deux joueurs, pour un total de moins de 200 coups, ce qui peut s'avérer insuffisant pour obtenir des résultats fiables.

VI. Conclusion

A. Résumé des principales conclusions du projet

En conclusion, la classification des coups est satisfaisante, mais les résultats obtenus peuvent être améliorés en augmentant la taille du dataset. Ce dernier pourrait être rendu plus général afin de pouvoir classer des coups pour des joueurs présentant des niveaux différents. Outre la classification, la détection pourrait également être affinée afin d'évaluer pleinement l'algorithme de classification. L'utilisation d'un autre facteur que l'accélération en X, comme la norme, pourrait rendre la détection plus efficace.

Par ailleurs, nous avons montré que la précision peut être améliorée en ajoutant de nouvelles données à notre modèle d'entraînement. En intégrant plus de variabilité dans notre algorithme, nous avons pu augmenter de 10 % la précision dans la prédiction des services.

Cependant, la principale limite de notre projet réside dans le fait que nous ne connaissons pas le niveau des joueurs sélectionnés dans notre dataset de référence. Autrement dit, les patterns auxquels nous associons l'accélération et la vitesse angulaire peuvent être trop différents entre les joueurs que nous avons suivis et ceux de l'étude de Zhao et al. Pour garantir une meilleure précision, il aurait fallu accompagner plus de joueurs sur le terrain, pour limiter l'impact des données de l'étude de Zhao et introduire plus de variabilité avec les nôtres.

B. Suggestions pour des recherches futures.

En ce qui concerne les recherches futures pour améliorer la classification des coups de tennis, trois idées principales pourraient être explorées.

Tout d'abord, l'utilisation d'un autre algorithme de classification, tel qu'un réseau de neurones, pourrait permettre d'améliorer la précision de la classification des coups de tennis. En effet, les réseaux de neurones sont connus pour leur capacité à traiter des données complexes et non linéaires, ce qui pourrait être particulièrement utile pour traiter les données de mouvement des joueurs de tennis.

Ensuite, une application de visualisation en temps réel pourrait être développée pour permettre aux joueurs de tennis de suivre leurs performances lors des matchs. Cette application pourrait utiliser des capteurs portables pour collecter les données de mouvement des joueurs et afficher des informations en temps réel sur le type et le nombre de coup. Les joueurs pourraient utiliser ces informations pour ajuster leur stratégie de jeu en temps réel et améliorer leur performance.

Enfin, une autre piste de recherche intéressante serait d'adapter l'algorithme de classification à d'autres sports de raquette, tels que le badminton ou le squash. Cela permettrait de développer une méthode de classification plus générale qui pourrait être appliquée à différents sports, et d'ouvrir de nouvelles perspectives en matière d'analyse des performances pour les joueurs de ces sports. En somme, ces trois idées pourraient contribuer à améliorer la précision et l'applicabilité de la classification des coups de tennis, tout en ouvrant de nouvelles perspectives pour la recherche sur les sports de raquette en général.

VII. Ressources

L'ensemble du code utilisé pour notre projet est disponible sur GitHub via le lien suivant :

<https://github.com/VictorVatt/ShotPredictionApp>.

Vous y trouverez le code source, le dataset utilisé pour l'entraînement du modèle, ainsi que les fichiers que nous avons utilisés pour évaluer notre modèle. En outre, un guide pour utiliser l'application sera également disponible.

VIII. Bibliographie

- Budi, D. R., Syafei, M. M., Kusuma, M. N. H., Suhartoyo, T., Hidayat, R., & Listiandi, A. D. (2020). The significance of exercise method on forehand and backhand groundstroke skills improvement in tennis. *Jurnal sportif : jurnal penelitian pembelajaran*. https://doi.org/10.29407/js_unpgri.v6i1.13920
- Ebner, C. J., & Findling, R. D. (2019). Tennis Stroke Classification. *Advances in Mobile Multimedia*. <https://doi.org/10.1145/3365921.3365929>
- Fabian 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, 2011.
- Kos, M., Zenko, J., Vlaj, D., & Kramberger, I. (2016). Tennis stroke detection and classification using miniature wearable IMU device. *International Conference on Systems, Signals and Image Processing*. <https://doi.org/10.1109/iwSSIP.2016.7502764>
- Wu, M., Fan, M., Hu, Y., Wang, R., Wang, Y., Li, Y., Wu, S., & Xia, G. (2022b). A real-time tennis level evaluation and strokes classification system based on the Internet of Things. *Internet of things*, 17, 100494. <https://doi.org/10.1016/j.iot.2021.100494>
- Zhao, H., Wang, S., Zhou, G., & Jung, W. (2019). TennisEye. *Information Processing in Sensor Networks*. <https://doi.org/10.1145/3302506.3310404>

IX. Annexes

Annexe 1 : Ensemble des résultats

	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 %