



ÉCOLE CENTRALE NANTES

Option DATASIM

---

Projet : **Détection automatique de figures de skateboard**

**Rendu final du projet.**

---



*Elèves :*

*Julien Douxami ;*

*Pierre Libault*

# Sommaire

1.	Description du projet.....	3
1.1.	Contexte général du projet .....	3
1.2.	Objectifs du projet.....	4
1.3.	Description de la démarche suivie .....	4
2.	Etat de l'art.....	5
3.	Travaux réalisés .....	7
3.1.	Architecture générale de l'algorithme .....	7
3.2.	Détection d'événements. ....	8
3.2.1.	Pré-traitement des données. ....	8
3.2.2.	Détection des figures.....	10
3.3.	Algorithme de classification. ....	12
3.3.1.	Les données à disposition.....	12
3.3.2.	Méthode du filtre adapté.....	12
3.3.3.	Mesure de la distance à une moyenne. ....	13
3.4.	Evaluation de la performance de l'algorithme.....	14
3.4.1.	Evaluation de la performance de la détection de figures .....	14
3.4.2.	Evaluation de la performance de la classification. ....	15
3.4.3.	Exemple de résultat fourni par notre algorithme. ....	18
3.5.	Limites de notre approche .....	19
4.	Conclusion et ouverture .....	20
5.	Bibliographie.....	21
6.	Annexe.....	22
6.1.	Figures de références exploitées pour la distance euclidienne .....	22

# 1. Description du projet

## 1.1. Contexte général du projet

Le Skateboard est une discipline sportive très en vogue ces dernières années, avec près de 20 millions de pratiquants dans le monde<sup>1</sup>. Cette pratique recouvre différentes disciplines sportives qui sont apparues aux JO de Tokyo en 2021, qui ont toutes en commun la réalisation de figures acrobatiques, notées par un jury. L'officialisation de ce sport dans des compétitions de renom comme les Jeux Olympiques entraîne une recherche de performance, qui peut être assistée par l'exploitation de données sportives spécifiques au skateboard.

C'est dans ce contexte général que nous souhaitons **développer un ensemble d'algorithmes de traitement de signal permettant de détecter et de caractériser les différentes figures réalisées par des pratiquants**, afin qu'ils puissent s'améliorer en visualisant leurs entraînements. Ces algorithmes se basent sur l'exploitation de données issues d'une centrale inertielle (Capteur Movuino) présenté ci-dessous :



*A gauche le Movuino placé dans le boîtier et à droite le boîtier fixé sur le skate*

Ce capteur récolte les données d'accélération et de gyroscope, ce qui peut être suffisant pour détecter de multiples figures. Cependant, il est important de noter que certaines figures ne peuvent être identifiées uniquement à partir de données issues du skateboard puisque que les mouvements du pratiquant (un tour sur soi-même par exemple) peuvent modifier la figure. Dans le cadre de ce projet, nous nous intéressons principalement à certaines figures identifiables avec les données issues du skate uniquement.

Ce capteur a déjà été utilisé par Pierre lors de sa césure, ce qui facilite son utilisation et l'extraction des données. Cependant, certaines problématiques associées à l'extraction des données vont faire leur apparition au fur et à mesure des travaux réalisés.

---

<sup>1</sup> <https://www.etudier.com/dissertations/March%C3%A9-Du-Skate/279975.html#:~:text=Les%20ventes%20fran%C3%A7aises%20sont%20environ,%C3%A0%20leur%20client%C3%A8le%20des%20skateboards.>

## **1.2. Objectifs du projet**

Comme mentionné précédemment, l'objectif général est de mettre en place un pipeline d'algorithmes qui permettent de détecter et de classer les différentes figures réalisées par un pratiquant de Skateboard.

Cet objectif général est découpé en plusieurs sous-objectifs intermédiaires :

- Détecter le nombre de figures réalisées sur une plage d'enregistrement ;
- Collecter suffisamment de données labellisées ;
- Mettre en place un protocole d'évaluation de la qualité des algorithmes ;
- Classer les différentes figures dans certaines classes ;
- Détecter si une figure est réussie ou non ;
- Déterminer des données pertinentes d'évaluation de la qualité d'une figure :
  - Hauteur ;
  - Vitesse initiale vs vitesse finale ;
  - Etc...

Les objectifs mentionnés ci-dessus sont les objectifs initiaux de ce projet, qui sont naturellement amenés à évoluer en fonction de la difficulté de certaines tâches et de l'apparition de nouvelles difficultés techniques à résoudre.

## **1.3. Description de la démarche suivie**

Afin de mener à bien ce projet, nous avons dans un premier temps réalisé un état de l'art, sur les différentes approches qui permettent de détecter et de classer des mouvements. De plus en plus, les données deviennent un réel enjeu qui permettent d'améliorer des performances sportives, ce qui explique la quantité d'articles scientifiques traitant de ce sujet. Ainsi, nous n'avons pas eu de difficultés à répertorier une dizaine d'articles traitant de la reconnaissance de mouvement, et plus particulièrement de classification de figures (soit dans le Skateboard directement ou dans d'autres activités sportives similaires comme le Snow, surf...).

L'analyse de cet état de l'art et les discussions avec M. Le Carpentier (encadrant de ce projet) nous a poussé à adopter une démarche segmentée, avec deux tâches principales à réaliser. Dans un premier temps, nous nous sommes concentrés sur la détection et le pré-traitement de données d'évènements dans un signal d'acquisition. Cela nous permet ainsi d'isoler des évènements, ce qui facilite ainsi la labellisation des données et la mise en place d'un algorithme de classification. Une fois cette première étape passée, nous souhaitons mettre en place un algorithme de classification puis de déterminer une méthode permettant d'évaluer la performance de cet algorithme.

## 2. Etat de l'art

Comme mentionné précédemment, la data devient un réel enjeu de performance dans le sport. En particulier, de plus en plus de scientifiques s'intéressent aux sports impliquant des figures comme le skateboard ou encore le snowboard, le ski... L'objectif est d'accompagner le sportif dans son entraînement en fournissant des métriques qualitatives sur les figures réalisées. Ainsi, il existe de multiples articles récents traitant du sujet de reconnaissance de figures. L'état de l'art réalisé regroupe certains articles jugés pertinents et qui nous ont fourni des pistes de réflexion pour mener à bien notre projet.

Dans l'article [1], les auteurs segmentent leur algorithme en deux parties :

- Une partie sur la détection d'événements. Cela permet d'identifier et d'isoler les figures dans un signal d'une grande durée, ce qui assure une réduction de la quantité de la donnée que l'on souhaite classer ;
- Un algorithme de classification qui détecte le type de figure réalisée.

La classification est ensuite assurée en étudiant 6 types de figures différentes identifiées par de multiples features statistiques sur le signal (moyenne, variance, kurtosis, skewness...). Au total, 343 signaux sont à disposition pour l'entraînement et l'évaluation de leurs algorithmes. La matrice de confusion obtenue est la suivante :

predicted \ true						
	O	N	K	H	P	360
O	33	1	0	0	0	0
N	2	31	0	0	0	0
K	0	0	36	0	0	1
H	0	0	0	37	0	0
P	0	0	0	0	35	1
360	0	0	0	0	0	49

Figure 1 : Matrice de confusion obtenue dans [1].

La méthode exploitée dans cet article assure ainsi une bonne performance de la classification mais nous ne possédons pas suffisamment de données pour envisager une classification exploitant autant de features statistiques. Cependant, segmenter l'algorithme en deux parties distinctes semble être une bonne idée pour évaluer la performance de détection et de classification de manière indépendante.

Dans l'article [8], les auteurs exploitent cette fois-ci un réseau de neurones artificiel qui est entraîné avec 459 échantillons de données (80% sont utilisées pour l'entraînement et 20% pour la validation). L'architecture du réseau est schématisée ci-dessous :

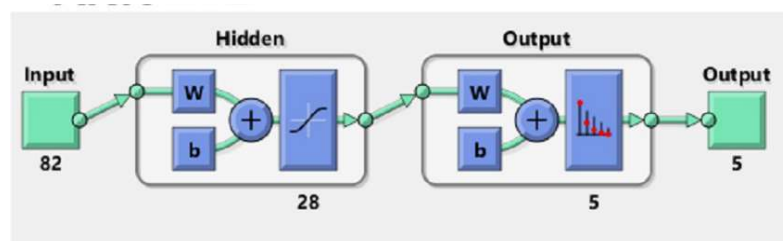


Figure 2 : Schéma du réseau de neurones exploité dans [8].

Pour assurer la performance d'un réseau de neurones artificiel, chaque paramètre du réseau doit être optimisé. Cette optimisation est assurée dans la phase d'entraînement, qui est d'autant plus performante qu'il y a de données disponibles. Notre quantité de données est bien trop faible pour espérer exploiter un tel algorithme. Cependant, le capteur utilisé par les auteurs de l'article fournit les mêmes types de donnée que le capteur que nous utilisons. L'article étudie en parallèle la performance de leur algorithme en n'exploitant qu'un seul axe sur les trois disponibles et l'axe de rotation Z semble être suffisant pour classer les différents types de figure. Cela peut s'expliquer par l'extraction de multiples features et leur interprétation par le réseau de neurones. Bien que nous ne pouvons pas exploiter un tel algorithme, cette méthode nous permet également d'envisager l'étude uniquement de certains axes qui dépendent de la figure étudiée lors de la classification nous donne une piste de réflexion pour étudier seulement certains axes en fonction de la figure étudiée.

D'autres articles comme [4] exploitent des données GPS pour améliorer la précision et le suivi du sportif. Il est intéressant de savoir qu'il est possible de prendre en compte ces données mais le capteur utilisé actuellement pour le projet ne permet pas de récupérer de telles données.

Enfin, l'article [3] recense de multiples études sur la reconnaissance de mouvements mais la plupart de ces méthodes exploitent des algorithmes qui nécessitent une phase d'entraînement ainsi qu'une phase de validation, qui ne sont possible que lorsque les données sont suffisamment abondantes. Ainsi, certains scientifiques comme dans l'article [7] exploitent des CWT ou des SVM qui semblent très performants mais restent inenvisageables dans le cadre de ce projet.

### **Conclusion sur l'état de l'art**

Dans cet état de l'art, nous nous sommes concentrés sur les études qui se basent sur l'exploitation du même type de donnée :

- Accélération selon les trois axes de rotation ;
- Données sur le gyroscope.

Nous avons remarqué qu'il est fréquent d'exploiter des algorithmes de Machine Learning ou d'intelligence artificielle et que ces algorithmes fournissent de très bons résultats, que ce soit pour la détection d'événements ou bien pour la classification des figures. Cependant, ces algorithmes nécessitent une phase d'entraînement importante qui exploite de multiples données labellisées.

Or, ces données ne sont pas en libre-accès et nous souhaitons exploiter les données fournies par notre capteur. Ainsi, nous devons labelliser nous-même ces données, ce qui est une tâche chronophage et nous ne pouvons pas regrouper suffisamment de données pour mettre en place ces algorithmes d'apprentissage. Cependant, ces différents articles nous ont fourni de bonnes pistes de réflexion sur l'architecture général de notre algorithme ainsi que sur certaines bonnes pratiques à adopter.

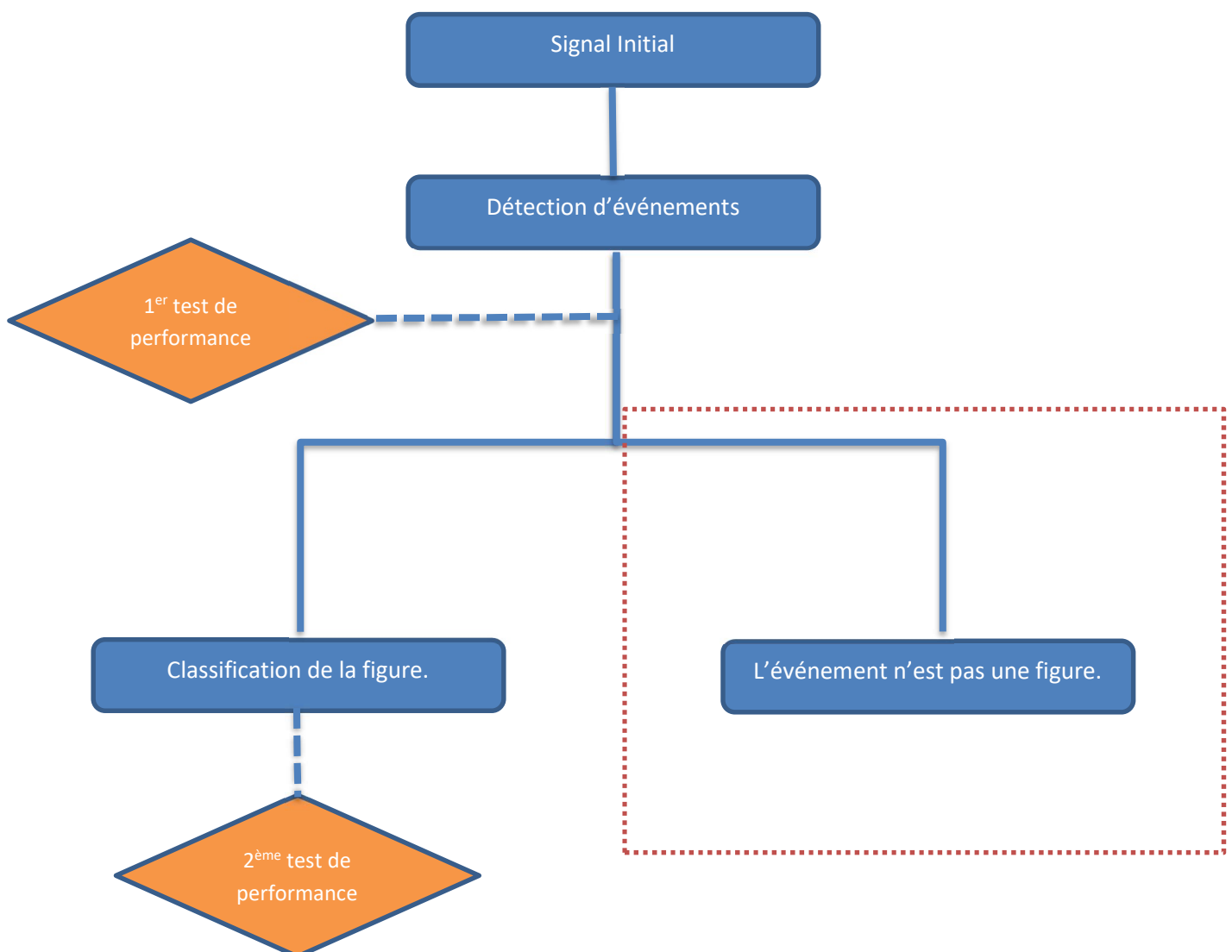
Ces articles seront d'autant plus pertinents dans le futur lorsque nous aurons été en mesure de collecter et de labelliser suffisamment de données.

### 3. Travaux réalisés

L'ensemble des développements est réalisé sur Python. Nous appelons événement une particularité dans le signal, qui peut être soit une figure (ce que l'on cherche à identifier), ou un mouvement parasite (virage, récupération de planche...).

#### 3.1. Architecture générale de l'algorithme

Comme mentionné dans l'état de l'art, nous nous sommes inspirés de l'article [1] pour construire notre algorithme de détection et de classification. Cela permet de dissocier la performance de détection et de classification, ce qui nous permet d'identifier quelle méthode fonctionne mal et quels sont les leviers techniques à lever pour améliorer la performance globale. Cette dissociation permet également de réduire la quantité de donnée à traiter par notre algorithme de classification. Ainsi, nous pouvons schématiser le fonctionnement de notre algorithme comme ci-dessous :



L'encadré en rouge correspond aux événements qui ne sont finalement pas une figure mais un mouvement jugé parasite : un virage, une récupération de la planche... Nous pouvons noter que chaque test de performance ne dépend que de l'algorithme associé (soit uniquement la prédiction soit uniquement la classification) mais qu'il est également possible de réaliser un test global de la performance en considérant un signal d'entrée et si nous nous intéressons uniquement à la

classification des figures dans ce signal. Dans la suite de ce document, nous allons nous intéresser à chaque étape de notre algorithme et nous allons décrire les différents modèles mis en place.

### 3.2. Détection d'événements.

L'utilisation du capteur étant maitrisée en amont du démarrage de ce projet, nous avons pu directement nous concentrer sur l'exploitation des données. Pour être en mesure d'analyser correctement les signaux, nous devons prétraiter les données en amont de la détection de figures.

#### 3.2.1. Pré-traitement des données.

Le pré traitement des données est une étape indispensable en traitement du signal car cela permet d'uniformiser et de mettre les données à la même « échelle », améliorant ainsi l'analyse de celles-ci. Pour les figures de skate, nous utilisons principalement 2 méthodes :

- Centrage des figures avec le temps moyen de la norme du gyroscope ;
- Normalisation des données d'accélération et gyroscopiques par l'énergie de leur norme respective ;

##### Centrage des figures en exploitant le temps -moyen :

Nous centrons les figures détectées de la même manière en calculant le temps moyen d'une figure par rapport à son énergie gyroscopique :

$$tempsMoyen = \frac{\sum_{k=1}^n t_i \cdot y_i^2}{\sum_{k=1}^n y_i^2}$$

Après avoir calculé le temps moyen de la figure, nous considérons que la durée de la figure est de 1.2 secondes, et que le temps moyen est au centre de cette figure. Cette durée est définie à l'expérience en considérant que les figures étudiées ne dépassent jamais cette durée. Ci-dessous un exemple de figure (encadrée en rouge dans le signal) extraite et centrée avec cette méthode.

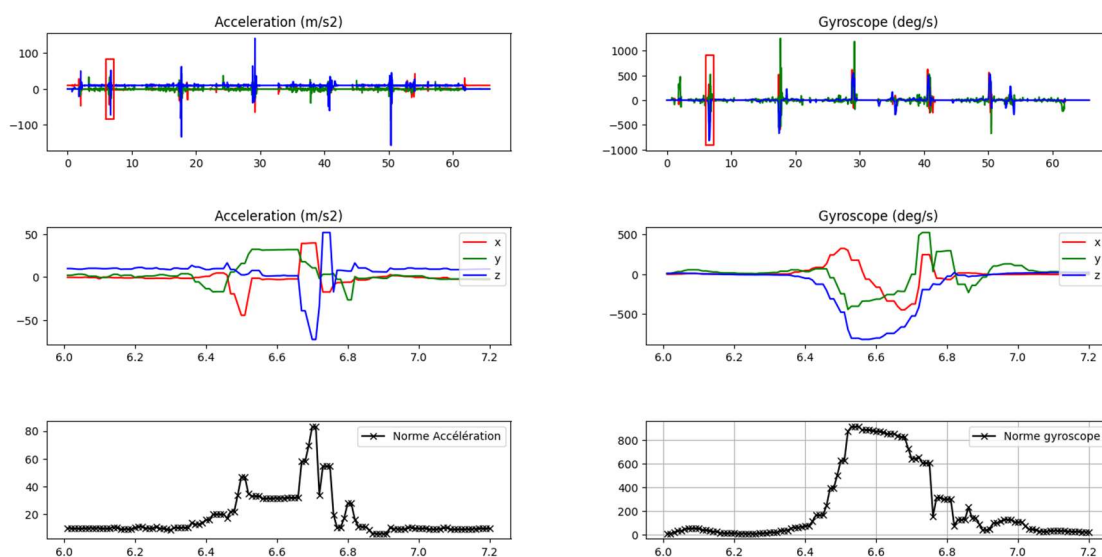


Figure 3 : Exemple de centrage d'une figure de Skate.



### Normalisation des données d'accélération et gyroscopiques :

Cette normalisation permet d'avoir des amplitudes de même ordre de grandeurs peu importe l'énergie globale de la figure. Cette étape est importante puisque l'énergie d'une même figure diffère fortement selon l'énergie mise dans la figure par le skateur.

### Problématique liée au capteur

En analysant nos résultats de temps moyen, nous nous sommes rendu compte que le centrage n'était pas parfaitement réalisé. Visuellement, il semblait que certaines données du signal étaient manquantes, ce qui faussait le calcul. Afin d'identifier la source de ce problème, nous traçons ci-dessous le temps entre chaque prise de points :

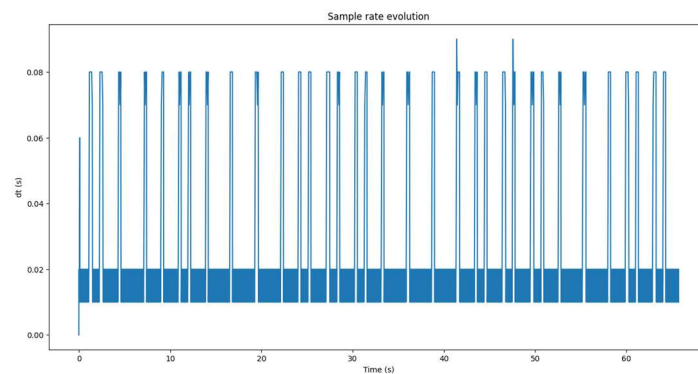


Figure 4 : Temps entre deux points

La figure 4 ci-dessus montre bien que la durée entre deux points est pratiquement constante (et alterne  $T_e=0,02s$  et  $T_e=0,01s$ ), sauf à quelques instants, où la durée devient un multiple de  $T_e$ . Cela indique que le problème provient simplement d'un défaut du capteur, et non d'une fluctuation de la période d'échantillonnage, ce qui facilite ainsi la reconstruction du signal. Les graphes ci-dessous correspondent à trois mêmes figures de skateboard et nous observons bien les conséquences de ce défaut du capteur :

- Elles n'ont pas toutes le même nombre de points ;
- Les espaces entre les points ne sont pas contrôlés ;
- Les temps moyens ne centrent pas de la même manière toutes les figures ;

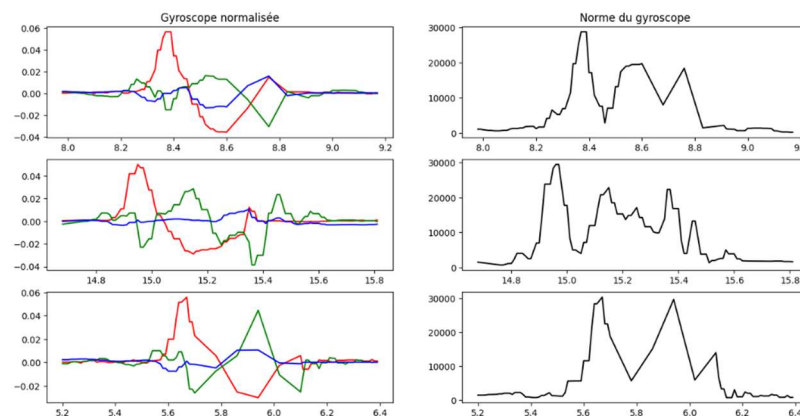


Figure 5 : Superposition de trois figures de Skate, avec mauvais nombre de points.

Afin de palier à ce problème, nous avons réalisé une **interpolation linéaire**, ce qui assure que chaque figure possède bien le même nombre de points. Cela est primordial pour les analyses futures, et notamment la classification des figures puisque chaque point devient une caractéristique importante permettant de classer la figure.

Nous traçons ci-dessous les figures correctement prétraitées :

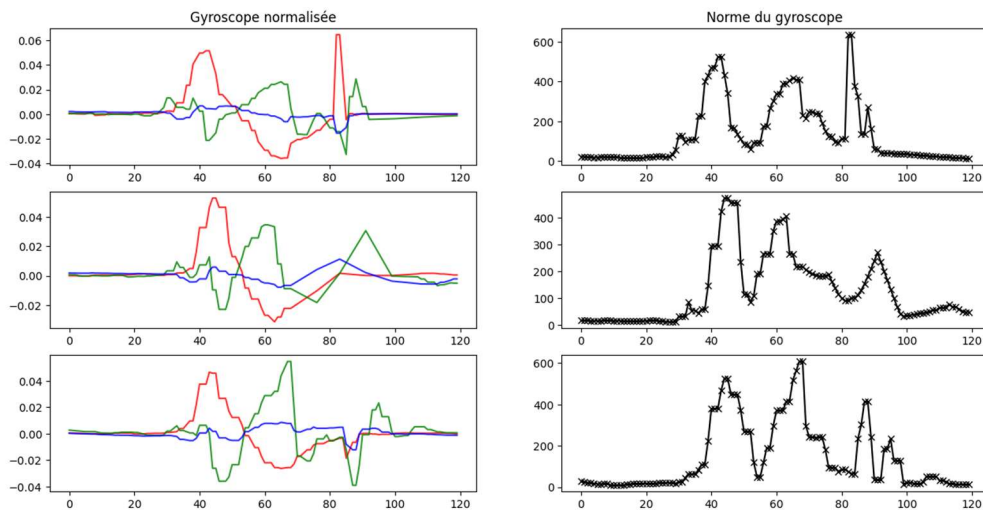


Figure 6: Superposition de trois figures de Skate, avec le bon nombre de points.

A présent, nous pouvons nous concentrer sur l'algorithme de détection d'événements :

### 3.2.2. Détection des figures.

Avant de pouvoir classer les différentes figures, la première étape est de pouvoir détecter les événements (qui correspondent aux figures) dans un signal d'entrée. Le graphe ci-dessous présente un signal de 65 secondes, et qui comporte 5 figures (dont 2 ratées).

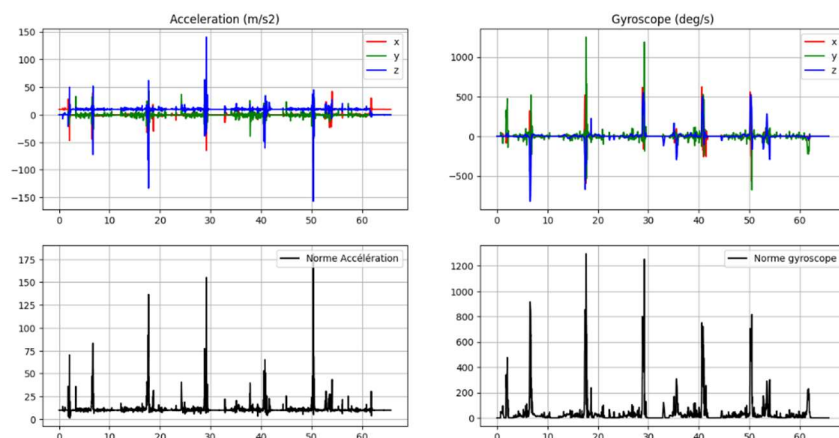


Figure 7 : Accélération et gyroscope d'un signal.

Comme nous pouvons le voir, des événements sont détectables à l'œil nu en analysant la présence de pics en norme d'accélération et de gyroscope. En effet, le skateur apporte de l'énergie au skateboard

pour effectuer ses mouvements ce qui se traduit par la présence de pics en accélération et en gyroscope. Cependant, certains mouvements, jugés parasites, comme des virages peuvent avoir un fort impact sur ces données de gyroscope. Etant donné que nous ne souhaitons récupérer que les figures, notre approche algorithmique exploite en simultanée les données d'accélération et de gyroscope afin de supprimer ces données parasites.

Pour détecter et isoler une figure, nous réalisons donc un fenêtrage des signaux de norme afin de calculer la moyenne fenêtrée de chaque norme. Pour s'assurer de ne louper aucun événement, nous avons choisi un over-lap de la taille d'une demi-fenêtre. Sur ces deux signaux fenêtrés, nous appliquons la fonction *find\_peaks* du module Scipy de python. Nous ajustons sur les paramètres *prominence* et *distance* afin d'optimiser les performances :

- Prominence : évaluation de l'importance du pic dans le signal ;
- Distance : Distance horizontal minimum entre les pics.

Nous parcourons ainsi l'ensemble du signal et lorsqu'il y a un pic à la fois en accélération et en angle sur la même fenêtre, nous définissons cet événement comme étant une figure à isoler. Une fois cette figure isolée, un pré-traitement est nécessaire pour uniformiser l'analyse de la figure et ainsi améliorer la performance de l'algorithme de classification de figures.

### Résultats de la détection d'événements :

Sur la figure 2 ci-dessous, nous représentons la séquence présentée dans la figure 1 avec les différents événements détectés par la méthode de fenêtrage. Nous remarquons que nous avons détecté 6 figures au lieu de 5, en effet le dernier événement détecté correspond à un virage. Nous le savons puisque nous avons enregistré le signal et nous connaissons ainsi le nombre de figures.

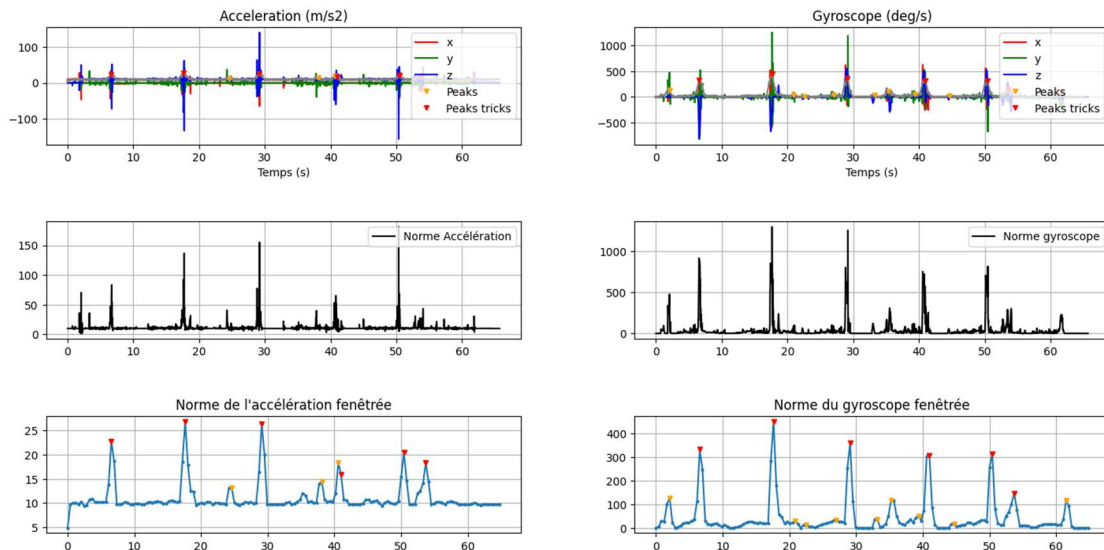


Figure 8 : Signal enregistré et détection de figures (en rouge sont les événements répertoriés comme des figures et en jaune des pics qui ne correspondent à un pic parasite).

Cet exemple montre bien que notre algorithme est pertinent puisque nous sommes en mesure de détecter les pics correspondant à des figures.

### 3.3. Algorithme de classification.

Maintenant que nous avons bien identifié les différents événements, nous avons déterminé deux méthodes qui nous permettent à priori de réaliser une classification efficace au vu du peu de données que nous avons.

#### 3.3.1. Les données à disposition.

Nous disposons de 50 signaux d'environ une minute qui comportent chacun entre une et quatre figures à identifier. Ces figures sont regroupées en 6 classes différentes, qui sont décrites ci-dessous :

Figures	Caractéristiques du signal
360 Flip	Rotation principale selon Z et Y ; Changement de signe et ondulation de l'angle
Front Side Shovit	Rotation principale dans le sens des Z positifs
Pop Shovit	Rotation principale dans le sens des Z négatifs
Heelflip	Rotation principale dans le sens des Y positifs
Kickflip	Rotation principale dans le sens des Y négatifs
Ollie	Saut avec rotation selon X

Pour un adepte du skateboard, la classification peut se faire visuellement sur le signal puisque chacune de ces figures possède un pattern bien précis que l'on peut identifier sur les données d'accélération.

#### 3.3.2. Méthode du filtre adapté.

La première méthode que nous avons décidé d'implémenter correspond à la méthode du filtre adapté. Cela consiste à considérer un « *signal de référence* » pour chacune des 6 classes puis de calculer la corrélation entre ce signal de référence et la figure que l'on souhaite classer. La plus grande corrélation correspond ainsi à la classe que l'on a souhaité identifier.

Bien qu'intuitive, cette méthode reste lacunaire puisque la corrélation est mesurée sur chaque axe de rotation, et certains axes sont parfois bruités et possèdent des variations importantes mais qui ne sont pas caractéristiques de la figure réalisée. Par exemple, le signal ci-dessous correspond à un Ollie, qui est principalement repéré par une rotation selon l'axe x. Cependant, nous pouvons tout de même remarquer une forte fluctuation de l'accélération selon l'axe y, qui risque de modifier la corrélation par rapport à la figure de référence.

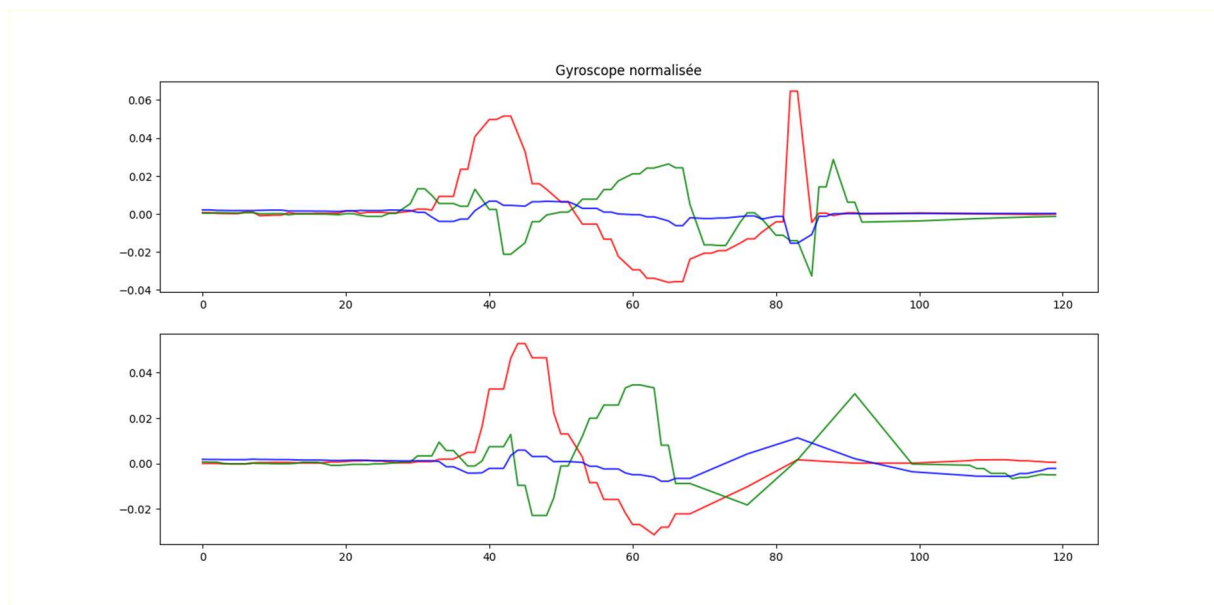


Figure 9 : Exemple de deux ollies, mettant en avant les différences entre signaux.

Ainsi, les corrélations mesurées ne possèdent pas le même ordre de grandeur, ce qui limite la comparaison des résultats ainsi que l'extraction d'informations supplémentaires comme le succès ou l'échec d'une figure. De plus, cette méthode semble trop dépendante du signal de référence que l'on choisit, ce qui limite la robustesse de la méthode. Puisque nous maîtrisons mal les résultats du filtre adapté et que nous avons du mal à interpréter les résultats à cause des différents ordres de grandeurs, nous avons développé une seconde méthode plus intuitive qui correspond à la mesure d'une distance euclidienne entre les signaux.

### 3.3.3. Mesure de la distance à une moyenne.

Pour pallier les problèmes précédents, nous avons choisi de mettre en place une méthode de mesure de la distance euclidienne entre la figure que l'on souhaite classer et une figure de référence pour chaque classe. Cette fois-ci, cette figure de référence correspond à la moyenne de 50% des figures labellisées. Ces moyennes sont calculées sur l'accélération et sur le gyroscope du capteur. Les figures moyennes obtenues pour chaque classe sont fournies en annexe du document. Pour exemple, la figure de référence du ollie est fournie ci-dessous :

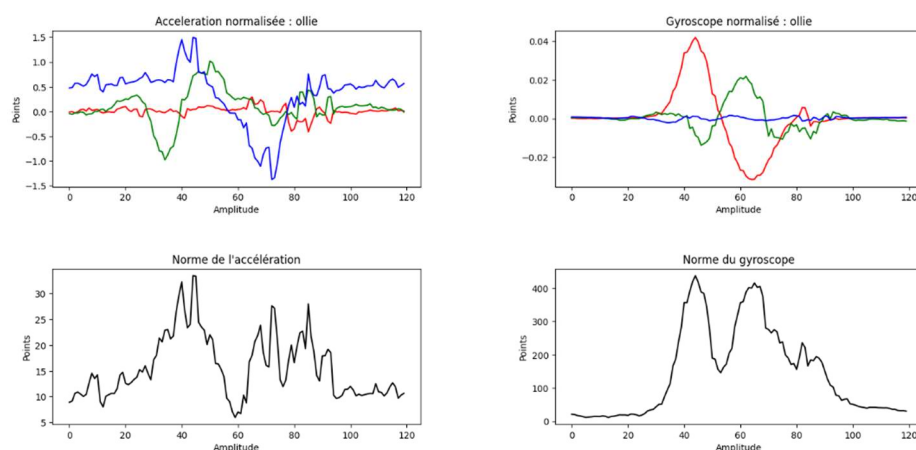


Figure 10 : Figure de référence pour le Ollie.

Comme nous pouvons le remarquer, les figures de références permettent bien de supprimer les variations sur les axes qui ne sont pas d'intérêt. Ainsi, pour chaque figure que l'on souhaite classer, nous mesurons la distance Euclidienne entre cette figure et la figure de référence pour chaque classe. La plus petite distance correspond alors à la classe d'intérêt.

Cette approche nous semble plus pertinente que celle du filtre adapté puisque nous sommes à priori en mesure de comparer des distances comparables et cette méthode permet de fournir d'avantage d'informations utiles. En effet, nous pouvons imaginer seuiller sur les distances afin de déterminer si une figure est réussie ou non. Par exemple, si la distance typique d'appartenance à une classe est de 0.1 et que pour une figure nous obtenons une distance minimale à chaque classe de 0.5, nous pouvons supposer que cette figure est soit une figure qui n'appartient à aucune classe (pas de signal de référence pour cette classe) ou bien que cette figure ne soit finalement qu'un mouvement parasite comme un virage ou une récupération de la planche.

Ainsi, cette méthode de classification permet à la fois d'ignorer les variations d'accélération sur les axes qui ne sont pas d'intérêt et permet de fournir davantage d'informations sur l'analyse du signal qui peuvent améliorer la performance globale de notre algorithme.

### 3.4. Evaluation de la performance de l'algorithme

A présent, nous souhaitons évaluer la performance de notre algorithme. Comme mentionné précédemment, l'architecture choisie nous permet d'évaluer de manière séparée la performance de la détection des figures et celle de la classification.

#### 3.4.1. Evaluation de la performance de la détection de figures

Dans un premier temps, nous souhaitons détecter l'ensemble des figures dans un signal. Cela signifie que nous souhaitons maximiser la sensibilité de l'algorithme de détection puisque nous souhaitons minimiser le nombre de faux négatifs, qui correspondraient à des figures non détectées. Ainsi, nous allons fixer les paramètres de la fonction *peak\_detection* de manière à récupérer l'ensemble des figures, quitte à sélectionner des événements qui ne sont pas des figures. Avec les paramètres fixés, nous obtenons la matrice de confusion suivante :

N = 162 (nombre de samples) True\Prediction	Positive	Negative
Positive	113	0 (FN)
Negative	49	N/A

Cette matrice de confusion démontre bien que notre choix de paramètre permet de maximiser la sensibilité (qui vaut 1) puisque nous n'avons aucun faux négatif détecté. Cependant, nous remarquons qu'il existe beaucoup de faux positifs, qui correspondent à des mouvements parasites comme des virages ou des récupérations de planche, qui sont responsables d'un pic important dans le signal.

La présence d'autant de faux positif n'est pas un problème à ce stade puisque la méthodologie choisie pour la classification permet justement d'écarter des faux positifs en seuillant sur la distance euclidienne. De plus, il est important de noter que le calcul de distance euclidienne n'est pas très

couteux ni chronophage. Ainsi, le calcul de distance sur les faux positifs n'est pas un élément trop contraignant pour notre algorithme.

### 3.4.2. Evaluation de la performance de la classification.

Pour rappel, notre algorithme de classification est basé sur la mesure de la distance euclidienne entre la figure que l'on souhaite classer et une figure de référence qui correspond à un signal moyenné sur une partie des données labellisées. Dans notre base de données, nous disposons de 85 figures labellisées réparties en 6 classes, chaque classe correspondant à une figure :

- 360 Flip : 8 figures ;
- Front Side Shovit : 16 figures ;
- Pop Shovit : 16 figures ;
- Heelflip : 7 figures ;
- Kickflip : 19 figures ;
- Ollie : 19 figures.

L'objectif de cette classification est double :

- Détecter les faux positifs de l'algorithme de détection d'événement.
- Classer les figures dans la bonne classe ;

#### Détection des faux positifs :

Dans un premier temps, nous nous sommes intéressés à la détection des faux positifs qui correspondent à des événements qui ne sont pas des figures. Pour cela, nous considérons la distance euclidienne entre chaque figure de référence et l'événement d'intérêt. Nous ne savons pas si la distance sur l'accélération ou celle sur le gyroscope permettrait de fournir le plus d'information pertinente. Ainsi, sur l'ensemble des événements détectés, nous avons mesuré la distance sur l'accélération et le gyroscope entre cet événement et les six figures de référence. Nous conservons pour chaque événement la distance minimale en accélération et gyroscope et nous traçons l'histogramme associé, en répartissant entre les événements qui sont des figures et les événements qui n'en sont pas :

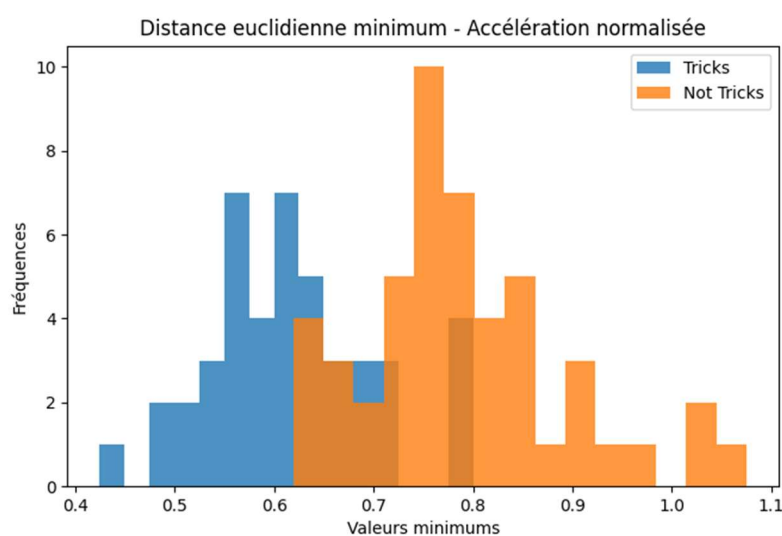


Figure 11 : Histogramme de répartition des événements en fonction de la distance euclidienne d'accélération.

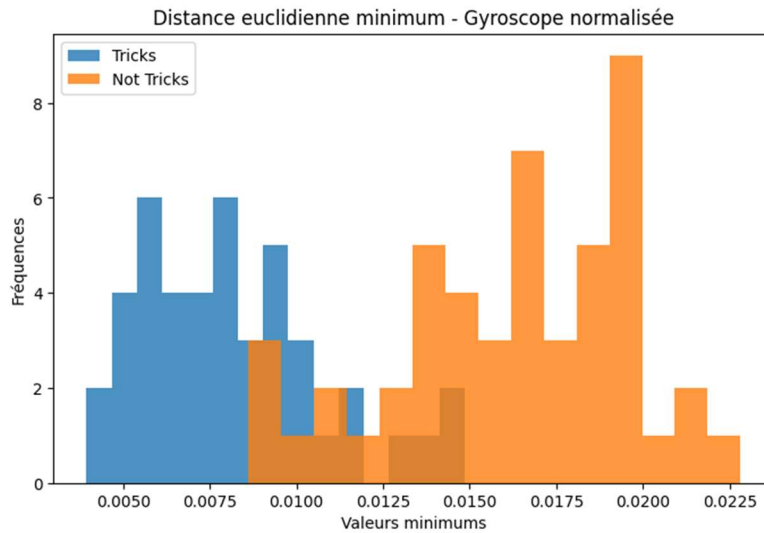


Figure 12 : Histogramme de répartition des événements en fonction de la distance euclidienne de gyroscope.

Comme prévu, les événements qui correspondent à des figures ont des distances plus faibles que les événements considérés comme des faux positifs. De plus, nous remarquons que la séparation est plus flagrante en s'intéressant uniquement aux données du gyroscope. L'idée à présent est de déterminer un seuil qui permet de supprimer suffisamment de faux positifs tout en conservant les figures d'intérêt. Pour cela, nous fixons de multiples seuils sur la distance et nous calculons pour chaque seuil le F1-Score. Nous choisissons alors le seuil qui permet de maximiser ce F1-Score ainsi que l'AUPRC :

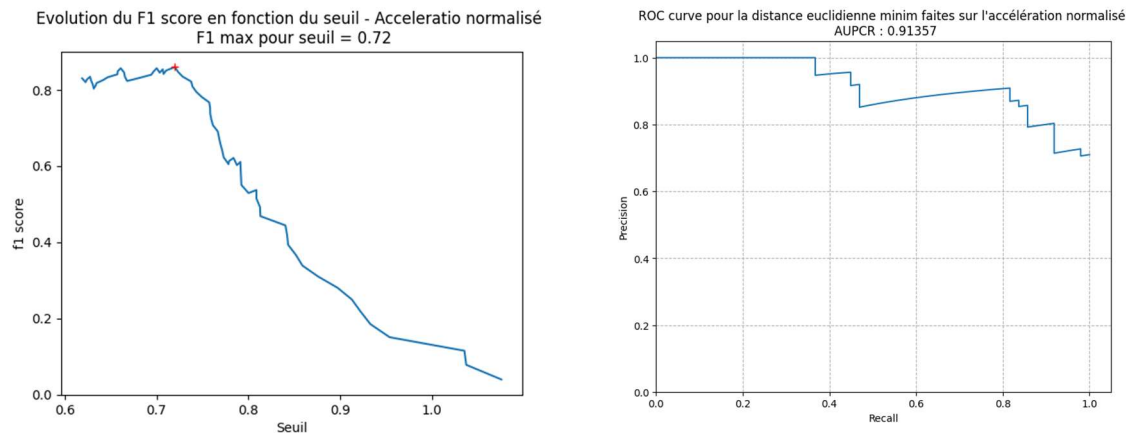


Figure 13 : Evolution du F1 score en fonction de différents seuils et courbe ROC pour la distance de l'accélération.



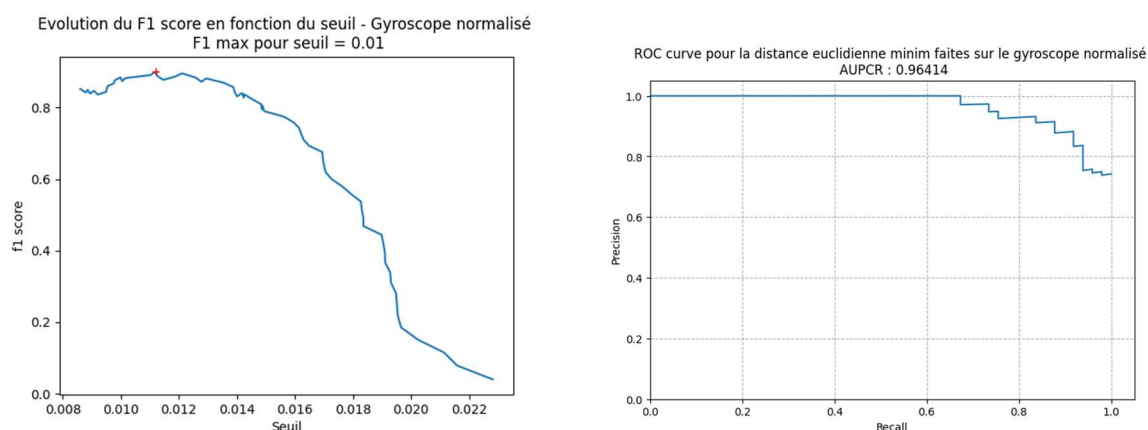


Figure 14 : Evolution du F1 score en fonction de différents seuils sur la distance du gyroscope.

Comme attendu, le meilleur F1-Score ainsi que la meilleure AUPRC est atteint pour la distance euclidienne basée sur le Gyroscope. Le seuil optimal correspond à une distance de 0.01, ce qui est également l'ordre de grandeur visible sur la figure 12. Ainsi, nous fixons ce seuil sur la distance du gyroscope pour assurer une meilleure performance de notre algorithme de détection de figures. Avec ce seuil, **nous obtenons un F1-Score de 0,9**. Avec ce seuil, la **précision est de 0.906**, et le **rappel est de 0.877**.

#### Classement des figures dans la bonne classe :

L'étape précédente nous a permis à la fois de réduire le nombre de faux positifs en éliminant les événements qui ne correspondent pas à des figures et de définir quelles données sont à exploiter pour réaliser la classification. En effet, les résultats précédents nous poussent à exploiter la distance mesurée sur le gyroscope pour réaliser la classification des figures. En ne considérant que les Tricks labellisés, nous avons un total de 41 données « d'entraînement » qui permettent de définir les figures de référence et 44 figures qu'il faut à présent classer. En considérant la distance euclidienne sur le gyroscope, nous obtenons la matrice de confusion suivante :

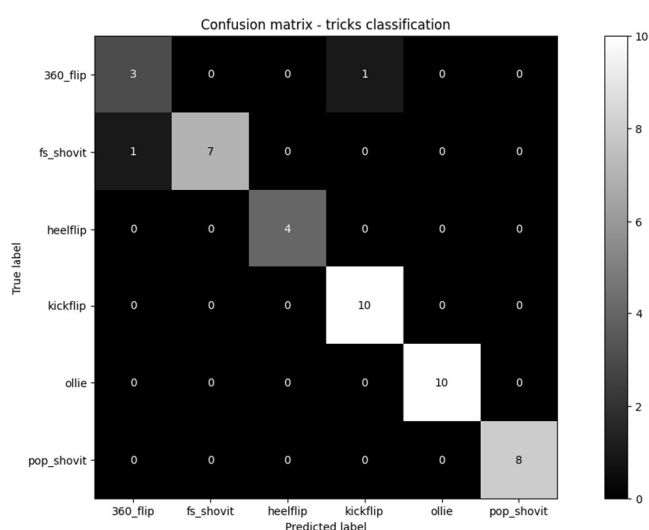


Figure 15 : Matrice de confusion pour la classification.

Comme nous pouvons le remarquer, la performance de notre algorithme de classification est très satisfaisante puisque nous obtenons **une micro average de 95.4%, une macro average de 94.3%, une weighted average de 95.6%**. Les erreurs quant à elles, se comprennent très bien. En effet, le 360 flip possède des mouvements selon l'axe y similaire au kickflip mais aussi au fs-shovit, il n'est pas surprenant que la distance entre ces deux figures puisse être proches.

### 3.4.3. Exemple de résultat fourni par notre algorithme.

Finalement, notre algorithme est en mesure de fournir des résultats assez robustes sur la détection des figures. L'image ci-dessous donne l'exemple d'un signal d'entrée de 40 secondes et l'algorithme est en mesure de détecter les différents événements puis de les classer.

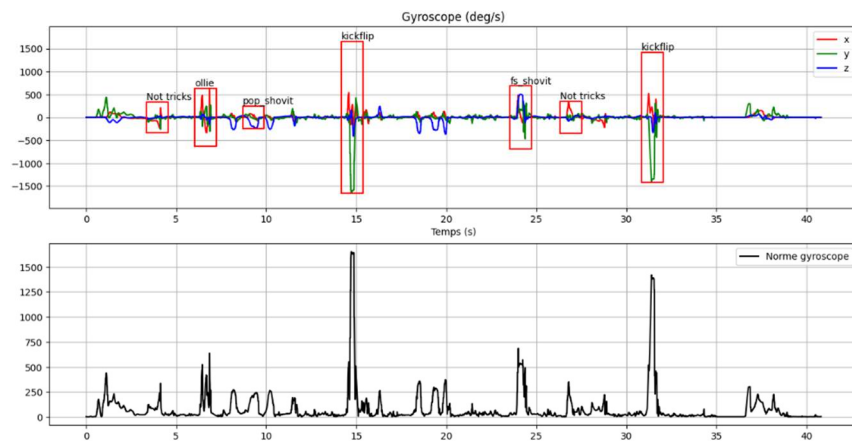


Figure 16 : Exemple de résultat de l'algorithme.

L'avantage de notre méthode est qu'elle est en mesure de détecter 6 types de figures différentes et permet également d'identifier des mouvements parasites considérés comme « Not tricks ». Pour plus de précision de l'algorithme, nous pouvons penser à augmenter le nombre de classes et de rajouter des classes typiques des mouvements parasites comme les virages ainsi que la récupération de la planche.

Enfin, nous avons débuté des travaux sur la classification des figures comme étant réussies ou non. Visuellement, nous remarquons que peu importe le type de figure réalisée, nous pouvons déterminer si celle-ci était réussie ou non en s'intéressant au signal d'accélération. Ainsi, nous avons créé une nouvelle classe qui correspond à « Figure ratée » et par mesure de distance sur le signal d'accélération, nous obtenons l'histogramme ainsi que la ROC curve suivante :

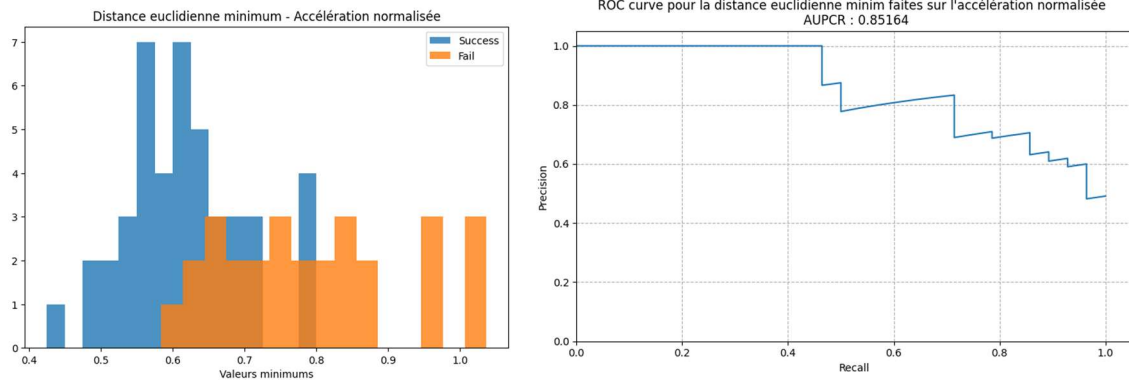


Figure 17 : Histogramme et ROC Curve pour la détection d'une figure ratée.

En réutilisant le maximum du f1 score, nous aurions un seuil permettant d'avoir une précision de \_\_\_\_ et un rappel de \_\_\_\_\_. Cependant, nous n'avons pas pu tirer davantage d'informations sur ces données par manque de temps mais les premiers résultats sont assez encourageants. Nous pensions à évaluer le succès de la figure sachant la classe prédite mais nous manquons de données.

### 3.5. Limites de notre approche

Bien que performant et précis, notre approche reste lacunaire pour différentes raisons :

- **La performance dépend des classes et le dataset n'est pas équilibré dans ces classes :** certaines classes sont mieux identifiées que d'autres, ce qui « boost » la performance. En effet, nous avons 10 ollies et 10 Kickflip correctement classés avec une précision de 100% pour chacune de ces classes. Pour le 360 Flip, nous avons 4 figures à classer mais seulement 3 sont correctement classées, ce qui donne une précision de 75%.
- **Certaines classes sont plus facilement différenciables :** Comme mentionné précédemment, les figures peuvent être identifiées à l'œil nu pour un adepte du Skateboard. D'autres figures peuvent avoir des signaux qui se ressemblent plus et la séparation uniquement par la mesure de distance risque d'être insuffisante. La seule figure où il y avait des erreurs (le 360 flip) correspond à une composition de mouvements d'autres figures (avec lesquels l'algorithme s'est trompé).
- **Manque de variabilité dans les données :** La plupart des données sont essentiellement récupérées par Pierre. Sur ces figures de « base », il possède une signature gestuelle propre à lui. Avoir des données de multiples skateurs permettrait d'avoir des figures de références plus générales mais aussi d'éviter un éventuel overfitting si l'on souhaite essayer un algorithme nécessitant un apprentissage.
- **Certains mouvement parasites peuvent être considérés comme des figures :** Nous arrivons à détecter ces mouvements parasites. Ce sont souvent des virages brusques et ils sont confondus avec des pop shovit et des fs shovit. Un travail peut être fait pour les identifier. On peut aussi penser à identifier ces mouvements parasites en créant une nouvelle classe caractéristique de ces virages.

## 4. Conclusion et ouverture

Pour conclure sur le développement du projet, nous avons trouvé intéressant de se confronter à un problème d'ingénierie classique qui est de traiter des données brutes afin d'en retirer des informations. Dans notre cas, il était particulièrement stimulant de traiter des données que nous avons-nous même générées avec un capteur disposé sur le skateboard. Le fait de générer nous-même les données nous a permis de prendre conscience de l'importance du pré-traitement de ces données et de l'aspect chronophage de leur labellisation.

De plus, les données étant propres à notre capteur, nous ne pouvions exploiter les différents articles scientifiques mentionnés dans la bibliographie. Ainsi, nous avons dû itérer sur différentes méthodes pour déterminer une méthode optimale qui permettait d'améliorer nos résultats. En ce sens, le choix de l'architecture de l'algorithme a été primordial pour tester différentes méthodes de manière indépendante. La méthode finalement choisie est performante dans notre situation avec seulement 6 types de figures à classer mais nous pouvons anticiper les lacunes de cette méthode en ajoutant de nouvelles figures.

En plus d'améliorer les performances de l'algorithme en acquérant plus de données et en testant différents modèles, il serait intéressant :

- De réaliser un travail sur l'extraction d'informations pouvant caractériser certaines figures comme des informations sur la fréquence, sur la hauteur de la figure...
- Etudier des figures mettant en jeu des rotations du corps (180, 360 etc) : prise en compte de données d'autres capteurs positionnés sur le skateur (téléphone, montre...) ;
- Utiliser le travail d'extraction d'informations afin de détecter les figures ratées voir aussi la figure tentée.
- Toutes nos figures sont captées avec le capteur situé à l'arrière du skate. Nous avons imaginé la possibilité de connaître la position du capteur sur le skate et de pouvoir retrouver une transformation permettant aux figures d'avoir la même représentation.

## 5. Bilbiographie

### Bibliographie

- [1] Benjamin & Kautz, Thomas & Schuldhuis, Dominik & Eskofier, Bjoern. (2015). IMU-based Trick Classification in Skateboarding.
- [2] Emily E Cust, Alice J Sweeting, Kevin Ball & Sam Robertson (2019) Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance, *Journal of Sports Sciences*, 37:5, 568-600, DOI: 10.1080/02640414.2018.1521769
- [3] Cust EE, Sweeting AJ, Ball K, Robertson S. Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance. *J Sports Sci*. 2019 Mar;37(5):568-600. doi: 10.1080/02640414.2018.1521769. Epub 2018 Oct 11. PMID: 30307362.
- [4] T. Hollecze, J. Schoch, B. Arnrich and G. Tröster, "Recognizing turns and other snowboarding activities with a gyroscope," *International Symposium on Wearable Computers (ISWC) 2010*, 2010, pp. 1-8, doi: 10.1109/ISWC.2010.5665871.
- [5] Anlauff, Jan & Weitnauer, Erik & Lenhardt, Alexander & Schirmer, Stefanie & Zehe, Sebastian & Tonekaboni, Keywan. (2010). A method for outdoor skateboarding video games. 40-44. 10.1145/1971630.1971642.
- [6] Harding, Jason & Mackintosh, Colin & Hahn, Allan & James, Daniel & Estivalet, Margaret. (2009). Classification of Aerial Acrobatics in Elite Half-Pipe Snowboarding Using Body Mounted Inertial Sensors (P237). 10.1007/978-2-287-09413-2\_55.
- [7] Abdullah MA, Ibrahim MAR, Shapiee MNA, et al. The classification of skateboarding tricks *via* transfer learning pipelines. *PeerJ Comput Sci*. 2021;7:e680. Published 2021 Aug 18. doi:10.7717/peerj-cs.680
- [8] Corrêa, Nicholas & Lima, Julio & Russomano, Thais & dos Santos, Marlise. (2017). Development of a skateboarding trick classifier using accelerometry and machine learning. 33. 362-369. 10.1590/2446-4740.04717.

## 6. Annexe

### 6.1. Figures de références exploitées pour la distance euclidienne

