Rapport de projet

Static and Dynamic Hand Gestures

Étudiants :
Idles MAMOU
Amine DJEGHRI

Numéros Étudiants :
3803676
3801757

Novembre 2020

Table des matières

1 Introduction

La reconnaissance des gestes de la main est un domaine de recherche en vogue depuis plusieurs années. C'est une partie importante dans l'interaction homme-machine et plusieurs applications de reconnaissance de gestes ont été développées et sont utilisées dans la réalité virtuelle, le contrôle de robot, les opérations cliniques et le langage des signes.

Les gestes de la main sont des mouvements de la main et du bras qui varient d'une posture statique à des gestes dynamiques. La reconnaissance de ces gestes nécessite leur modélisation dans les domaines spatial et temporel. Elle peut être catégorisée en deux catégories : la reconnaissance des gestes statiques (une posture stable de la main) et la reconnaissance des gestes dynamiques (une série de mouvements de la main) Principalement, les méthodes de reconnaissance de gestes se divisent en deux catégories : les méthodes basées vision et les méthodes basées sur des dispositifs attachés aux corps tels que les gants. La plupart des chercheurs utilisent des systèmes de reconnaissance de gestes basées sur la vision comme par exemple Microsoft Kinect, ils ne requirent que d'une caméra, cependant les variations d'illumination et les problèmes liés à l'arrière-plan qui perturbent les traitements d'images ainsi que les capteurs qui ne sont pas trop précis constituent des limites qui ont poussé les chercheurs à utiliser d'autres systèmes tels que les gants de données (cyberglove ou dataglove en anglais) et qui permettent de capturer des gestes plus complexes et d'obtenir des résultats plus précis.

Les gants de données sont des gants comportant des capteurs et permettent à un utilisateur de saisir et de manipuler un objet virtuel, en numérisant en temps réel les mouvements de sa main. Peu de datasets qui utilisent ce genre d'équipements existent dû à leur cout élevé.

Dans ce projet, nous utiliserons le jeu de données de reconnaissances de gestes à partir de mesures prises par un gant de données UC2017 Static and Dynamic Hand Gestures .

2 Analyse Préliminaire des Données

2.1 Exploration des Données

2.1.1 Présentation du DataSet :

Les données sur lesquelles nous travaillons proviennent du *UC2017 Static Dynamic Hand Gestures DataSet*.L'ensemble de données est divisé en deux sous ensembles correspondants à deux types de gestes : les gestes statiques et les gestes dynamiques. Les gestes statiques **SG** sont décrits par un seul pas de temps de données, représentant donc une pose et une orientation d'une seule main. Les gestes dynamiques **DG** sont des séries temporelles de poses et d'orientations de longueur variable avec des significations particulières. Certains gestes de l'ensemble de données sont corrélés à une certaine signification dans le contexte de l'IRH (Interaction Humain Robot), tandis que d'autres sont arbitraires, pour enrichir l'ensemble de données et ajouter de la complexité au problème de classification.

Pour capturer la forme, la position et l'orientation de la main au fil du temps, un gant de données (CyberGlove II) et un tracker magnétique (Polhemus Liberty) ont été utilisé. Le gant fournit des signaux numériques qui sont proportionnels à l'angle de flexion de chacun des 22 capteurs qui sont attachés élastiquement à un sous-ensemble des articulations de la main. De cette façon, nous avons une approximation de la forme de la main. Le capteur du tracker est fixé rigidement au gant sur le poignet et mesure sa position et son orientation par rapport à un cadre fixé au sol. L'orientation est la rotation entre le cadre fixe et le cadre du capteur, donnée sous forme des angles intrinsèques d'Euler de lacet, tangage et roulis (ZYX).

Dans le cadre de ce projet, nous nous intéressons qu'au sous-ensemble de gestes statiques SG.

Le DataSet contient 2400 exemples (gestes) obtenus auprès de 8 utilisateurs différents.Les variables sont toutes quantitatives. Ainsi, les 22 premières colonnes représentent l'angle de flexion de chacun des capteurs attachés à la main. Les 6 colonnes suivantes représentent l'orientation du cadre du capteur par rapport au cadre fixé au sol. Les 2 dernières colonnes représentent la classe de l'exemple (le geste) et l'utilisateur qui a généré l'exemple.

2.1.2 Étude de la répartition des exemples par classe

Le DataSet répertorie 24 classes au total, et nous pouvons observer sur la figure 2.1.1 que les classes sont parfaitement équilibrées. En effet, nous avons 100 exemples par classe (par geste).

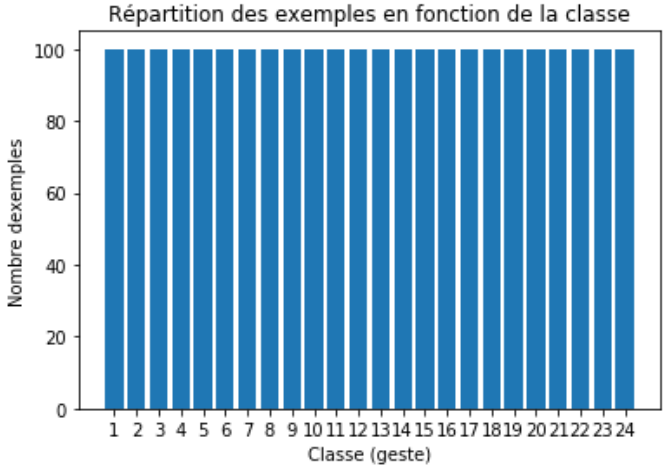

FIGURE 2.1.1 – Répartition des exemples par classes

2.1.3 Étude de la répartition des exemples par utilisateur

Nous observons sur la figure 2.1.2 que les exemples ne sont pas générés équitablement par les 8 utilisateurs. En effet, les utilisateurs 1 et 2 génèrent à eux seuls la moitié des exemples du DataSet, tandis que l'utilisateur 7 n'en génère que 120.

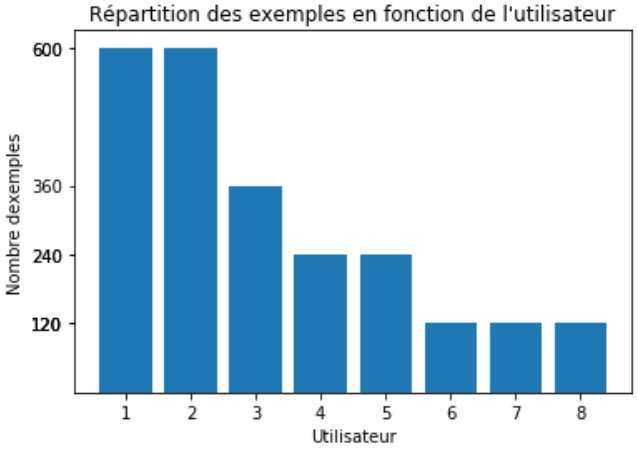

FIGURE 2.1.2 – Répartition des exemples par utilisateur

2.1.4 Étude de la répartition des exemples par classe et par utilisateur

Nous observons sur la figure 2.1.3 que les classes sont équilibrées pour les exemples provenant du même utilisateur. Ainsi, chaque utilisateur génère le même nombre d'exemples pour une classe (un geste) donnée.

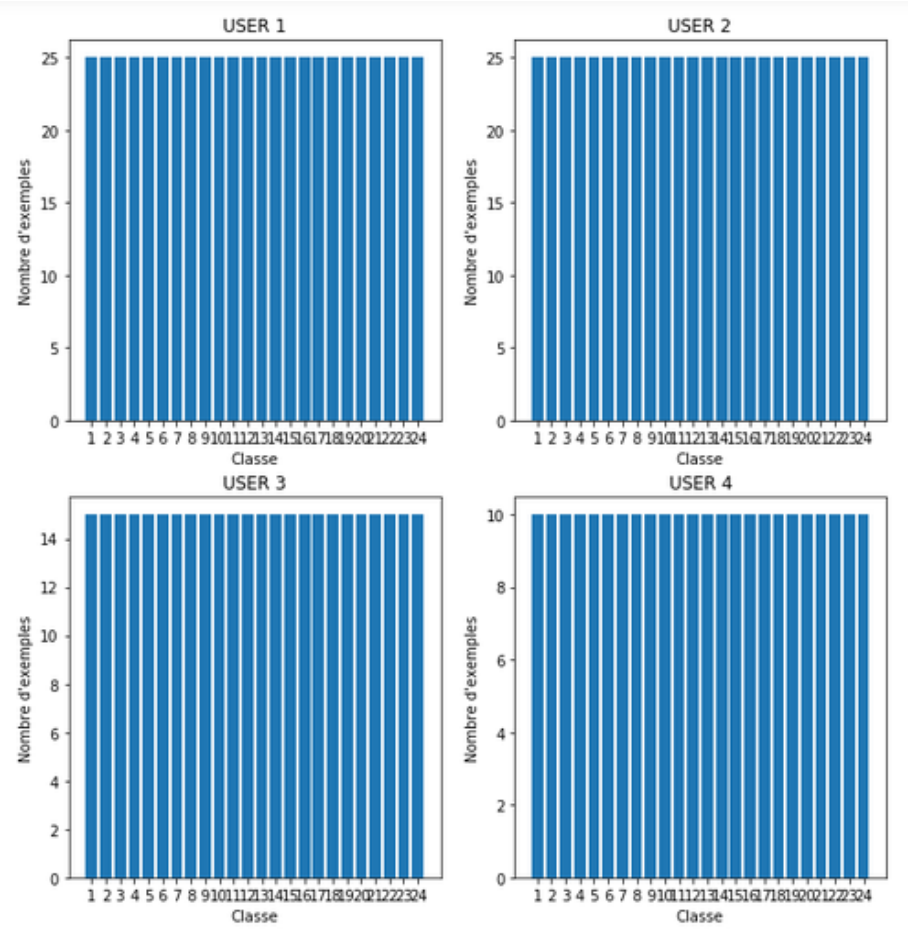

FIGURE 2.1.3 – Répartition des exemples par classe pour les 4 premiers utilisateurs

2.1.5 Étude des corrélations entre variables

- Corrélations entre features :
  Le niveau de corrélation entre les différentes caractéristiques (features) du DataSet est décrit par la matrice de corrélation 2.1.4 ci-dessous. Afin de pouvoir faire ressortir les variables les plus corrélées, nous avons tronqué la matrice au seuil de corrélation 0.7. Ainsi nous pouvons observer une très forte corrélation entre les caractéristiques [0 et 3], entre les attributs [11 et 12], [11 et 14] etc...

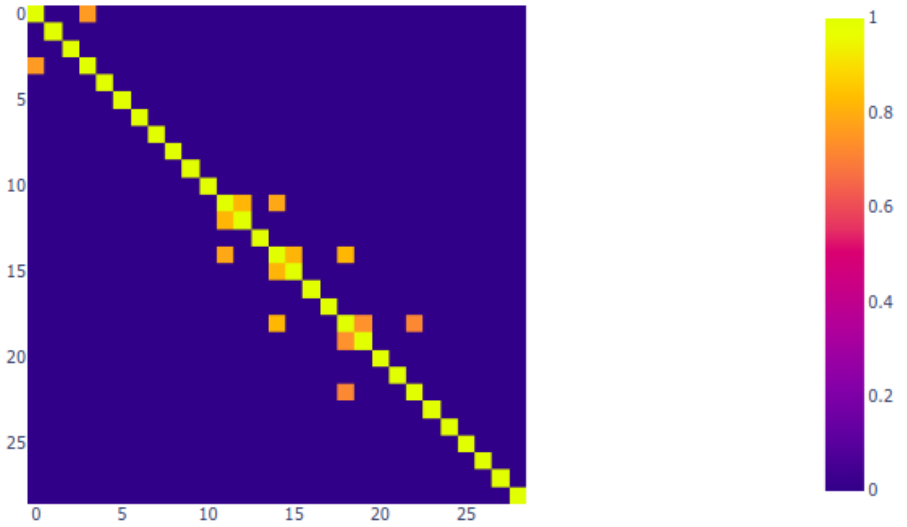

FIGURE 2.1.4 – Matrice des corrélations entre features au seuil 0.7

- Corrélations entre les features et la Target :
  La figure 2.1.5 ci dessous présente le niveau de corrélation linéaire entre les caractéristiques et la classe des exemples. On peut voir que les attributs 3,4,15 et 19 sont légèrement plus corrélés avec la target que les autres.

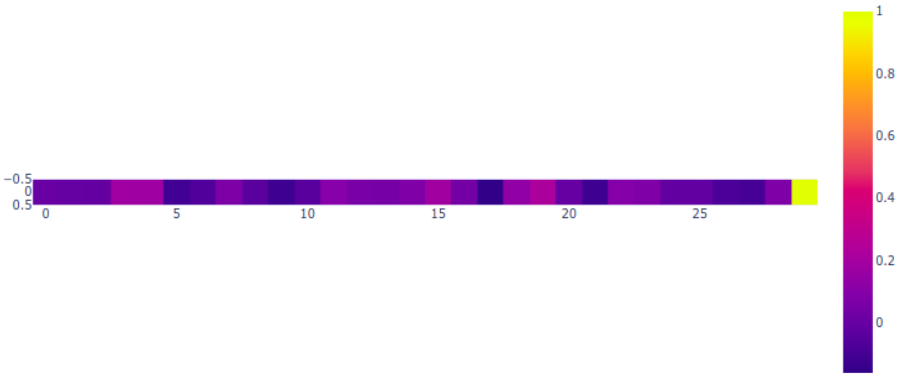

FIGURE 2.1.5 – Corrélations entre feature et target

2.2 Protocole expérimental

2.2.1 Description de la tache

Comme nous l'avons vu lors de l'exploration des données, les exemples du Dataset définissent chacun un geste de la main parmi 24 gestes au total. Nous proposons donc de nous confronter à un problème de classification supervisée multi-classes.

2.2.2 Partitionnement des données

Afin de partitionner de manière équitable notre ensemble de données et permettre à nos modèles de généraliser au mieux nous avons choisi de diviser notre DataSet en ensembles d'apprentissage et de test

suivant deux critères :

1. Faire en sorte que notre modèle ne soit pas collé aux utilisateurs et qu'il puisse bien généraliser sur de nouveaux exemples générés par d'autres utilisateurs (autres que les 8). Pour ce faire, nous optons pour un **apprentissage en validation croisée à 7 folds** tel qu'à chaque itération les exemples générés par un utilisateur sont mis de coté pour le test. Comme nous avons pu voir sur la figure 2.1.3, les classes sont équitablement réparties selon les utilisateurs. Nous avons donc la garantie que notre partitionnement ne changera pas l'équilibre global des classes 2.1.1. Les exemples générés par un huitième utilisateur seront ensuite utilisés pour le test final.

2. Faire en sorte que chaque utilisateur contribue de la même manière (avec le même nombre d'exemples) à l'apprentissage. En effet, comme nous avons pu l'observer sur la figure 2.1.2, les utilisateurs 1 et 2 génèrent beaucoup plus d'exemples que les autres et sont donc sur-représentés dans l'ensemble d'apprentissage, ce qui pourrait créer un biais. Pour palier à ce problème, nous proposons de construire nos 7 folds de sorte à ce qu'il y ai le même nombre d'exemples dans chaque fold. Notre ensemble d'apprentissage se retrouve, cependant, fortement diminué.

2.2.3 Métriques d'évaluation

Pour répondre à notre tache de classification multi classes, nous évaluerons nos modèles à l'aide des mesures suivantes :

— **Accuracy :** Comme les classes de notre Dataset sont équilibrées, le taux de bonne classification est un bon estimateur de la performance de nos modèles.

— **F1 Score Macro :** qui représente la moyenne harmonique entre de la précision et le rappel. Dans le cas multi classes un score F1 est calculé de manière indépendante pour chaque classe, puis la moyenne est calculée. Toutes les classes sont traitées de façon égale ce qui est adapté à notre tache de classification équilibré.

3 Analyse Expérimentale

3.1 Baselines

Nous allons d'abord utiliser des modèles de bases afin qu'ils servent de comparatif aux modèles que nous allons définir par la suite. Nous utiliserons pour cela un classifieur dummy selon différentes stratégies :

— **Stratified :** génère des prédictions aléatoires en respectant la distribution des classes de l'ensemble d'apprentissage.

— **Most frequent :** : prédit toujours l'étiquette la plus fréquente dans l'ensemble d'apprentissage.

— **Uniform :** : génère des prédictions uniformes au hasard.

	Accuracy(%)	
	Train	Test
Stratified	3.8	4
Most frequent	4.1	4.1
Uniform	4.5	4.3

TABLE 1 – Performances des Baselines

3.2 Modèles de classification multi-classes

3.2.1 Sélection des modèles

Pour sélectionner le meilleur modèle (choisir la meilleure combinaison d'hyper-paramètres) nous avons opté pour une recherche en grille tel que pour chaque paramètre, on détermine une ensemble de valeurs à tester.Le Grid Search croise simplement chacune de ces hypothèses et va créer un modèle pour chaque combinaison de paramètres. Au final, le modèle ayant obtenu la meilleure moyenne (Accuracy) en validation croisée sur les k ensembles de test est retenu.

- **Application du SVM :**
  Configuration de la Grid Search :
  Pour l'obtention d'un score de classification, l'option "Un contre Tous" a été utilisé. Le noyau utilisé a été sélectionné parmi les noyaux suivants : {'Linéaire','Polynomial','RBF'} tandis que le paramètre de régularisation "C" a été sélectionné parmi {0.001, 0.01, 0.1, 1, 10, 100}.
  Résultats CV :
  Meilleure Configuration : {Noyau = Linéaire, C = 0.1 }.
  avec :
  Score Accuracy : 89 %
  Score F1 Macro : 88 %

- **Régression Logistique :**
  Configuration de la Grid Search :
  Notre modèle a été implémenté avec le classifieur "Logistic Regression" de Scikit-learn doté du solver "SAGA". La pénalité utilisée a été sélectionné parmi les suivantes {Lasso (L1), Ridge (L2)} tandis que le paramètre de régularisation "C" a été sélectionné parmi {0.001, 0.01, 0.1, 1, 10, 100}.
  Résultats CV :
  Meilleure Configuration : {Pénalité = Ridge(L2), C =1 }.
  avec :
  Score Accuracy : 86 %
  Score F1 Macro : 84 %

- **Random Forests :**
  Configuration de la Grid Search :
  Notre modèle est initialisé avec comme méthode d'ensembles le Bootstrap et comme critère de séparation l'entropie. Le nombre d'arbres de décision dans le modèle a été sélectionné parmi {10,50,100}, le nombre de features à prendre en compte ont été sélectionnées selon les stratégies {'auto','sqrt','log2' }. La profondeur des arbres a été sélectionnée parmi {10,50,100} tandis que le nombre minimum d'exemples pour un split est sélectionné parmi {2,5,10}.
  Résultats CV :
  Meilleure Configuration : {NB arbres = 50 ,Max-features = 'auto' , Profondeur = 10, min-split = 5 }.
  avec :
  Score Accuracy : 94 %
  Score F1 Macro : 93 %

- **K plus proches voisins (KNN) :**
  Configuration de la Grid Search :
  Le nombre de voisins à prendre en compte a été sélectionné parmi {5,10,20,30}, la distance utilisée parmi {'Euclidienne','Manhattan} tandis que la distribution des poids parmi {"uniforme","selon la distance"}.
  Résultats CV :
  Meilleure Configuration : {K=30, Distance = Manhattan, Poids : selon la distance }.

avec :
Score Accuracy : 87 %
Score F1 Macro : 86 %

3.2.2 Résultats

Le tableau 2 ci-dessous présente un récapitulatif des résultats obtenus pour chacune des baselines. On remarque que le KNN et les random forests présentent une forte variance (différence de 10% sur l'accuracy entre validation et test).

	Accuracy(%)		**F1-Score(%)**	
	Cross-Validation	Test	Cross-Validation	Test
KNN	87	78	86	77
Random Forests	94	84	93	83
LogRegression	86	86	84	86
SVM	89	88	88	88

TABLE 2 – Performances des modèles

4 Conclusion

Dans ce projet, nous avons étudié la classification des gestes de la main en utilisant les données de gestes statiques SG provenant du dataset UC2017 Static DynamicHand Gestures Dataset. Après avoir mené une analyse exploratoire des données et étudié la répartition des exemples ainsi que la corrélation entre les variables, nous avons suivi le protocole constitué d'abord du partitionnement des données en suivant deux critères : Faire en sorte que notre modèle ne soit pas collé aux utilisateurs (validation croisée à 7 folds), mais également veiller à ce que chaque utilisateur contribue de la même manière à l'apprentissage. Nous avons ensuite défini des métriques d'évaluation et établi des baselines avec des méthodes simples. Enfin, nous avons proposé des modèles afin d'améliorer le score et exploré les hyperparamètres.