

IMT Atlantique

Technopôle de Brest-Iroise - CS 83818

29238 Brest Cedex 3

Téléphone : +33 (0)2 29 00 13 04

Télécopie : +33 (0)2 29 00 10 12

URL : www.imt-atlantique.fr



Rapport de projet de 3ème année d'IMT Atlantique

Evaluation automatique 3D de l'épaisseur de cartilage fémoro-tibial

Emmanuel LEGUET

emmanuel.leguet@imt-atlantique.net

Thomas RODER

thomas.roder@imt-atlantique.net

Vanessa VANDENBERGHE

vanessa.vandenberghe@imt-atlantique.net

Dylan WECKELS

dylan.weckels@imt-atlantique.net

Tuteurs de projet :

Valérie BURDIN

Guillaume DARDENNE

Mai Quyen PHAM

Date d'édition : 11 octobre 2024

Version : 1.0



IMT Atlantique

Bretagne-Pays de la Loire

École Mines-Télécom

Sommaire

Introduction	4
1. Anatomie du genou et pathologies	5
1.1. Anatomie du genoux	5
1.2. Pathologie	6
1.3. Prothèse	6
2. Méthodes de réduction de dimension	7
2.1. ACP	7
2.2. NMF	8
2.3. TNF	9
3. Méthodes de classification	11
3.1. CNN	11
3.2. MeshCNN	12
3.3. SVM	12
4. Méthodologie	14
4.1. Données	14
4.2. Préparation des données	15
4.2.1. MeshCNN	15
4.2.2. Cartes d'épaisseur	16
4.2.3. Recalage 3D	18
4.3. Métriques d'évaluation	19
5. Résultats	20
5.1. Réseaux de neurones	20
5.1.1. MeshCNN	20
5.1.2. CNN	21
5.2. Machine learning "classique"	22
5.2.1. ACP	24
5.2.2. NMF	24
5.2.3. TNF	25
5.2.4. Sans méthode de réduction	26
5.3. Récapitulatif des résultats	26
6. Conclusions	27
7. Management de projet	28
7.1. Organisation, planification et répartition des tâches	28
7.2. Identification des risques	29
7.3. Communication interne et externe	29
Références	30

Liste des figures

1.	Os et cartilages du genou [5]	5
2.	Soutien ligamentaire [5]	5
3.	Vascularisation du genou [5]	5
4.	Innervation du genou [5]	5
5.	Exemple gonarthrose [7]	6
6.	Genou implanté d'une prothèse complète [8]	6
7.	Visualisation de l'analyse en composante principale[9]	7
8.	Visualisation des 4 premières composantes obtenus grâce à l'ACP sur une carte d'épaisseur du fémur	8
9.	Factorisation de matrices NMF[10]	8
10.	Visualisation des 4 premières composantes obtenus grâce à la NMF sur une carte d'épaisseur du fémur	9
11.	Visualisation des 4 premières composantes obtenus grâce à la TNF sur une carte d'épaisseur du fémur	10
12.	Architecture pytorch du CNN utilisée lors de notre projet	11
13.	Visualisation de différentes tailles de maillage pour une même image [16]	12
14.	Illustration du résultat de SVM[17]	13
15.	Histogramme du nombre de faces des maillages de cartilages de fémur	15
16.	Visualisation de la première composante sur les cartes d'épaisseur du cartilage du tibia et du fémur (rouge : valeur positive, bleu : valeur négative)	16
17.	Visualisation des cartes moyennes pour les tibias	16
18.	Visualisation des cartes moyennes pour les femurs	17
19.	Cartes d'épaisseur du cartilage du tibia et du fémur chez un patient sain	17
20.	Cartes d'épaisseur du cartilage du tibia et du fémur pour un genou pathologique	17
21.	Nuages de points des cartilages de 2 fémurs avant et après recalage (15 itérations)	18
22.	Paramètres d'entraînement MeshCNN	20
23.	Résultats moyens des entraînements des différents classifieurs	23
24.	Résultats des entraînements labellisés par méthode de réduction utilisée	23
25.	Exemple du trello utilisé par l'équipe	28
26.	Diagramme de Gantt	29

Liste des tableaux

1.	Données fournies par le CHU de Brest	14
2.	Données trouvées sur le site de Standford	14
3.	Données totales pour la réalisation du projet	14
4.	Matrice de confusion pour la classe <i>Pathologique</i>	19
5.	Matrice de confusion pour la classe <i>Sain</i>	19
6.	Matrice de confusion pour la classe <i>Sain</i> , Mesh CNN sur Fémur	20
7.	Matrice de confusion pour la classe <i>Sain</i> , CNN sur Fémur	21
8.	Matrice de confusion pour la classe <i>Sain</i> , CNN sur Tibia	21
9.	Matrice de confusion pour la classe <i>Sain</i> , ACP sur Fémur	24
10.	Matrice de confusion pour la classe <i>Sain</i> , ACP sur Tibia	24
11.	Matrice de confusion pour la classe <i>Sain</i> , NMF sur Fémur	24
12.	Matrice de confusion pour la classe <i>Sain</i> , NMF sur Tibia	25
13.	Matrice de confusion pour la classe <i>Sain</i> , TNF sur Fémur	25
14.	Matrice de confusion pour la classe <i>Sain</i> , TNF sur Tibia	25
15.	Matrice de confusion pour la classe <i>Sain</i> , Sans réduction sur Fémur	26

16.	Matrice de confusion pour la classe <i>Sain</i> , Sans réduction sur Tibia	26
-----	--	----

Introduction

L'arthrose est une maladie qui altère le cartilage au niveau des articulations du corps humain. L'une des arthroses considérée comme les plus douloureuses et handicapantes est la gonarthrose. La gonarthrose concerne le genou et résulte d'une réduction du cartilage fémoro-tibial ou fémoro-patellaire au cours de la vie du patient. Dans le cas d'une atteinte au niveau de l'articulation fémoro-tibiale, la réduction du cartilage entraîne le frottement des os l'un contre l'autre, ce qui est très douloureux pour le patient[1]. Actuellement, les manières d'augmenter significativement la qualité de vie du patient sont soit des traitements antalgiques, soit la pose d'une prothèse de genou afin d'empêcher les deux os de l'articulation de se toucher[2]. Les prothèses actuelles ne sont pas très diversifiées et ne permettent pas de prendre en compte les spécificités du genou du patient, ce qui rend souvent les patients peu satisfaits du résultat. D'après une étude de J. Noble[3] de 2006, 25% des patients sont insatisfaits ou neutres par rapport à leur opération, dont 11% très insatisfaits.

Dans le cadre du projet "FollowKnee"[4], des prothèses spécifiques patient sont développées afin que la satisfaction des patients à la suite de la pose de telles prothèses soit plus importante. Une automatisation d'un maximum de parties de la conception des prothèses est donc recherchée afin de rendre la création des prothèses et l'opération de mise en place les moins coûteuses, les plus rapides et les plus simples possibles. Pour cela, des données nous ont été fournies en début de projet ainsi que des codes pour le traitement de ces données.

La mission à travers notre projet est donc de proposer et de comparer différentes méthodes pour la classification des images médicales en deux catégories : sain et pathologique. Nous avons donc implémenté des algorithmes de machine learning ainsi que des architectures de deep learning et comparé leurs performances à l'aide de métriques d'évaluation adaptées. Nous avons également implémenté des algorithmes de réduction de dimension afin d'optimiser le temps de calcul et les performances de nos algorithmes. Ces méthodes ont également été évaluées en même temps que les algorithmes de classification. Une partie du projet a aussi été consacrée aux traitements de nos données initiales ainsi que la recherche et l'adaptation de nouvelles données.

1. Anatomie du genou et pathologies

1.1. Anatomie du genoux

Le genou est une articulation complexe du membre inférieur reliant la cuisse et la jambe. Elle doit supporter le poids du corps tout en restant flexible et mobile.

Cette articulation fait intervenir trois os : le fémur, le tibia et le patella (ou communément appelé rotule). On y retrouve donc 2 articulations : l'articulation fémoro-patellaire et l'articulation fémoro-tibiale (articulation d'intérêt dans le cadre de notre projet). Ces dernières sont appelées des articulations synoviales puisqu'elles permettent un mouvement libre des os entre eux. Chaque os est composé d'un cartilage qui permet aux articulations de fonctionner sans frottement entre les os, ceux-ci sont visibles dans la figure 1. Cela est notamment aussi possible grâce au liquide synovial que l'on retrouve dans les cavités articulaires.

Toute l'articulation est soutenue par un système ligamentaire comprenant les 2 ligaments croisés (antérieur et postérieur) et les 2 ligaments latéraux (interne et externe), ceux-ci sont visibles dans la figure 2. Les ménisques internes et externes sont aussi responsables du soutien de l'articulation lors de la marche par exemple.

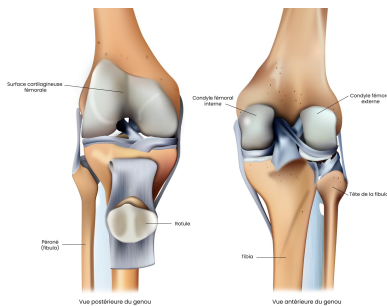


FIGURE 1 – Os et cartilages du genou [5]

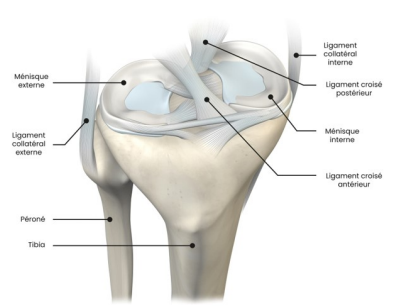


FIGURE 2 – Soutien ligamentaire [5]

La vascularisation du genou est assurée par l'artère poplitée et la veine poplitée, toutes deux situées dans la partie postérieure du genou et visibles dans la figure 3. Ces deux vaisseaux assurent aussi la transition entre la cuisse et la jambe et permettent une vascularisation de la partie distale du membre inférieur.

Dans la figure 4, on peut voir que le genou est innervé par des branches des deux principaux nerfs de la cuisse :

- **Le nerf sciatique** qui se divise pour donner le nerf tibial et les nerfs fibulaires commun et superficiel qui innervent la partie antérieure du genou.
- **Le nerf scaphène**, qui accompagné du nerf obturateur, innerve la partie postérieure du genou.

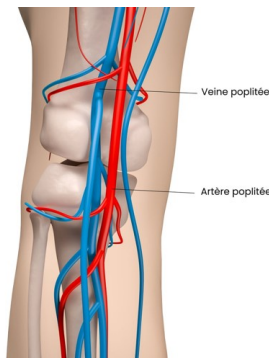


FIGURE 3 – Vascularisation du genou [5]

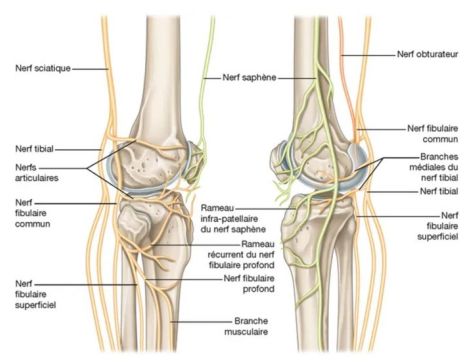


FIGURE 4 – Innervation du genou [5]

1.2. Pathologie

L'arthrose, ou plus précisément la gonarthrose, est due à une réduction du cartilage fémoro-tibial ou fémoro-patellaire et ne se guérit pas. Une illustration de gonarthrose est visible dans la figure 5. L'arthrose est causée par des facteurs comme l'âge, le sexe, les antécédents familiaux, le surpoids ou une mobilisation trop importante de l'articulation. Elle peut aussi être la conséquence d'autres pathologies comme des lésions osseuses (fractures par exemple), un traumatisme (rupture de ligaments ou lésion du ménisque) ou une déviation de l'axe de la jambe en varus ou en valgus [6].



FIGURE 5 – Exemple gonarthrose [7]

La douleur arthrosique peut se traiter par des médicaments antalgiques, la prise d'anti-inflammatoires non stéroïdiens (AINS) par voie orale ou locale, ou par des injections intra-articulaires de corticoïdes ou d'acide hyaluronique[2].

La prévention de l'apparition et de l'aggravation de l'arthrose peut cependant aussi se faire et repose sur une kinésithérapie et une activité physique régulière, adaptée pour maintenir la force des muscles.

Si ces traitements ne sont pas efficaces ou que l'arthrose est trop grave, alors une opération est nécessaire pour la pose d'une prothèse. Celle-ci ne prend actuellement pas suffisamment en compte la morphologie du patient et l'insatisfaction des patients résulte souvent d'une discordance entre la géométrie de la prothèse et la morphologie du genou.

1.3. Prothèse

Il existe deux types de prothèses du genou :

- La prothèse dite **totale** qui remplace les 3 parties de l'articulation du genou : le fémur, le tibia et le patella.
- La prothèse dite **partielle** qui ne remplace qu'une partie du genou.

La prothèse consiste donc à remplacer toute ou partie de l'articulation par des alliages métalliques composés de cobalt, de chrome et de molybdène. La partie mobile est composée quant à elle de polyéthylène, un plastique très utilisé dans le domaine médical pour son faible coût et sa résistance. Un exemple de prothèse est visible dans la figure 6.



FIGURE 6 – Genou implanté d'une prothèse complète [8]

Afin de poser une prothèse à un patient, une planification pré-opératoire et per-opératoire est nécessaire. Celle-ci se fait grâce à un arthroscanner qui a pour principe la réalisation d'un scanner à la suite d'une injection intra-articulaire de produit de contraste iodé.

2. Méthodes de réduction de dimension

Les méthodes de réduction de dimension permettent de réduire les dimensions de l'espace des données, en retirant une partie de l'information. On peut ainsi facilement passer d'un dataset avec des images de tailles 300×300 , soit 90000 valeurs par images, à un dataset réduit avec des images représentées seulement par quelques centaines de valeurs, tout en conservant une grande majorité de l'information (cela dépend du dataset).

Ces méthodes permettent donc de réduire la taille des données, ce qui est utile d'un point de vue de coût computationnel, mais aussi de ne garder que l'information pertinente et minimiser le bruit des données, ce qui peut aider pour l'étape de classification. Enfin, en général, ces méthodes permettent d'obtenir des dimensions que l'on peut interpréter, et elles facilitent donc l'analyse des données et potentiellement l'analyse des résultats des algorithmes de classification. Nous avons expérimenté 3 approches de réduction, décrites ci-dessous.

2.1. ACP

L'algorithme ACP (Analyse en composantes principales) est un algorithme de réduction de dimensionnalité dont le but est de trouver une nouvelle base orthonormée dans laquelle représenter nos données, telle que la variance des données selon ces nouveaux axes soit maximisée itérativement pour chaque nouvel axe.

La première étape est de construire un vecteur w_1 , qui maximise la variance de $w_1^T X$, avec X la variable aléatoire considérée ou dans notre cas notre donnée. Celle-ci correspond au vecteur propre de la valeur propre la plus prépondérante dans la matrice de covariance de σ , définie comme XX^T .

On peut ensuite essayer d'en définir un second w_2 orthogonal à w_1 , qui correspond de la même manière au vecteur propre de la seconde valeur propre la plus prépondérante.

Le nombre de composantes à choisir afin de créer notre nouvelle base descriptive de nos données correspond à la dimensionnalité que nous recherchons. Ce nombre doit être plus petit que la dimensionnalité originale afin de pouvoir limiter la quantité de données et doit normalement être étudié afin de déterminer le minimum nécessaire à la bonne utilisation des méthodes de classification. Cela se fait habituellement grâce à un calcul de variance, qui, lorsqu'elle dépasse un seuil ou ne varie plus assez suite à l'ajout d'une composante, indique que le nombre de composantes est suffisant.

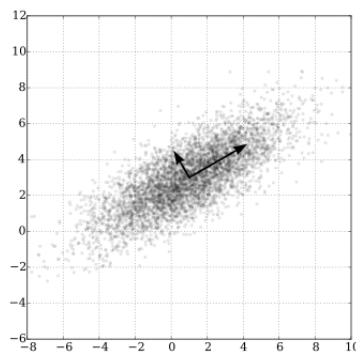


FIGURE 7 – Visualisation de l'analyse en composante principale[9]

La figure 7 présente un exemple de décomposition par ACP sur des données fictives. On peut y voir la nouvelle base qui sert à décrire les données. La projection des données sur les axes peut être utilisée par la suite pour de nouvelles méthodes.

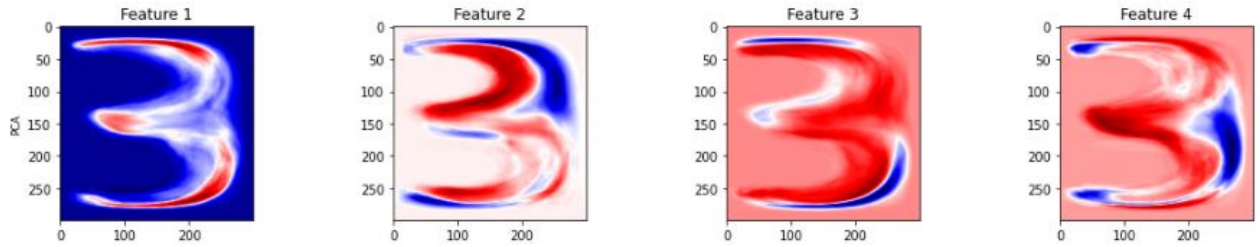


FIGURE 8 – Visualisation des 4 premières composantes obtenus grâce à l'ACP sur une carte d'épaisseur du fémur

2.2. NMF

L'approche NMF (Non-Negative Matrix Factorisation) est une opération de réduction de dimension de données. Elle sert à factoriser une matrice de données V en deux matrices de plus faibles dimensions, dont l'une correspond aux caractéristiques extraites W , et l'autre aux données réduites H , où chaque donnée est réduite en vecteur de poids (qui pondèrent les caractéristiques). La méthode profite du fait que les matrices traitées ne présentent que des valeurs positives ou nulles, et comme le problème est très rarement soluble, il s'agit d'une approximation.

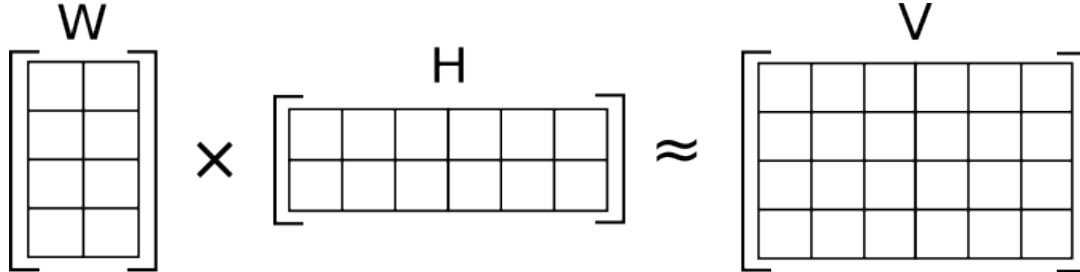


FIGURE 9 – Factorisation de matrices NMF[10]

Dans la figure 9, la matrice V est factorisée en deux matrices W et H dont la première ligne de H peut par exemple être utilisée en entrée des méthodes de classification.

Il s'agit d'une opération itérative qui converge vers deux matrices dont les valeurs sont non négatives et dont les valeurs peuvent être itérativement définies de cette manière[11] :

$$H_{[i,j]}^{n+1} \leftarrow H_{[i,j]}^n \frac{((W^n)^T V)_{[i,j]}}{((W^n)^T W^n H^n)_{[i,j]}}$$

et :

$$W_{[i,j]}^{n+1} \leftarrow W_{[i,j]}^n \frac{(V(H^{n+1})^T)_{[i,j]}}{(W^n H^{n+1} (H^{n+1})^T)_{[i,j]}}$$

Cet algorithme est déjà utilisé dans le domaine médical, par exemple avec les électromyogrammes (EMG)[12]. Il est aussi bien adapté à l'utilisation de machine learning et permet de réduire considérablement le temps de calcul et est notamment déjà implémenté avec SVM dans la classification d'autres pathologies telles que la détection de la maladie d'Alzheimer avec 90% d'efficacité dans une étude[13] de Padilla P. de 2012.

Nous avons utilisé cet algorithme afin de tenter d'améliorer les résultats des méthodes de classification. Ainsi, pour obtenir la matrice de poids des données de test, on peut simplement remplacer la matrice V des données d'entraînement par celle des données de test tout en gardant la matrice W obtenue par l'entraînement. On peut donc obtenir la matrice des poids des données de test de cette manière :

$$H_{test} = W_{train}^{-1} V_{test}$$

La matrice W n'est cependant pas inversible, car elle n'est pas carrée, mais nous pouvons réécrire cette formule de cette manière afin d'obtenir une bonne approximation :

$$H_{test} = (W_{train}^T W_{train})^{-1} W_{train}^T V_{test}$$

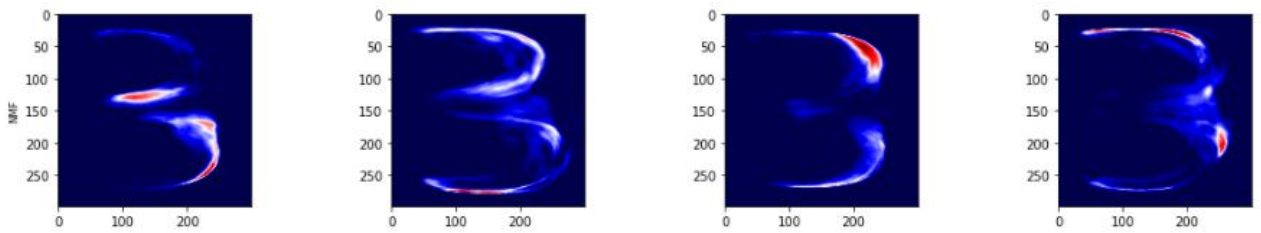


FIGURE 10 – Visualisation des 4 premières composantes obtenus grâce à la NMF sur une carte d'épaisseur du fémur

2.3. TNF

Nous avons également implémenté une version tensorielle de l'algorithme NMF : le TNF (Tensor Non-Negative Factorisation). Contrairement aux deux méthodes précédentes, cette méthode ne nécessite pas de vectoriser nos données et prend donc en compte la structure matricielle de nos données qui seront des images : il s'agit de son principal intérêt. Celle-ci fonctionne de la même manière et permet d'obtenir un tenseur réduit du tenseur original. Cette fois-ci, le tenseur d'entraînement est obtenu par concaténation des images sur la 3^e dimension afin d'obtenir un tenseur de dimension 60x60x13 par exemple, pour 13 images de taille 60x60. La décomposition que nous avons choisie est PARAFAC2, celle-ci étant disponible sur tensorly[14], une bibliothèque python spécialisée dans les tenseurs et leurs traitements. Elle décompose notre tenseur $X \in \mathbb{R}^{I \times J \times K}$ en 3 matrices A, B et C de cette manière :

$$X_{ijk} = \sum_{r=1}^R A_{ir} B_{\sigma_i(j)r} C_{kr}$$

La matrice que nous cherchons à obtenir est C , car elle caractérise nos images. Le paramètre R correspond à la dimension à laquelle nous souhaitons réduire le tenseur original. La fonction PARAFAC2 de tensorly prend en entrée ce paramètre et renvoie les valeurs des matrices A, B et C . Les valeurs de C des images d'entraînement ne nous intéressent pas, mais grâce aux matrices A et B , nous pouvons calculer les valeurs de C pour nos données de test cette fois-ci.

Dans l'équation ci-dessus, la matrice B présente une permutation sur sa première dimension, mais celle-ci est déjà prise en compte dans le calcul de B par la méthode et celle-ci renvoie donc plutôt B_i tel que :

$$B_i = P_i B$$

avec P_i une projection de B . Nous pouvons donc ignorer cette permutation dans les calculs suivants.

Afin de retrouver la matrice C de nos données de test, nous pouvons utiliser, comme dans la méthode NMF, les matrices A et B_i de la décomposition pour recomposer notre nouvelle matrice C . On peut donc réécrire la première formule de cette manière, avec $\alpha_{ijr} = A_{ir} B_{\sigma_i(j)r}$:

$$X_{test,ijk} = \sum_{r=1}^R \alpha_{ijr} C_{test,kr}$$

Pour une même image k , on peut obtenir un système de $n * m$ équations (pour des images de tailles $n \times m$) avec R inconnues : $C_{test,kr}$, avec $0 < r \leq R$ et $X_{test,ijk}$ les valeurs du tenseur formé avec les images des données de test. En résolvant le système, on obtient les données réduites test $C_{test,kr}$.

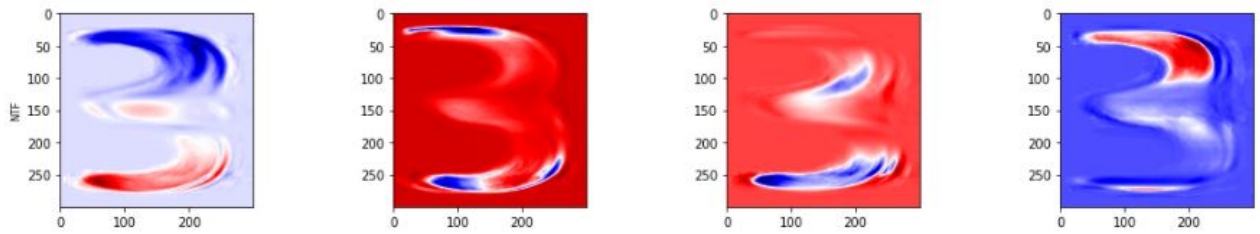


FIGURE 11 – Visualisation des 4 premières composantes obtenus grâce à la TNF sur une carte d'épaisseur du fémur

3. Méthodes de classification

Il existe beaucoup de méthodes pour la classification d'images médicales. Dans ce chapitre, nous avons identifié celles qui semblent correspondre à nos attentes et celles que nous avons implémentées. Nous discuterons aussi de la pertinence de celles-ci.

La plupart de ces méthodes peuvent aussi être utilisées pour la segmentation d'images. Dans le cadre de ce projet, nous ne les utiliserons que pour la classification, car nos données sont déjà segmentées par un expert, même si celles-ci ont dû être adaptées pour l'application de nos méthodes.

3.1. CNN

La plupart des architectures de deep learning ayant pour but de traiter des images utilisent des CNN (Convolutional Neural Network).

Les CNN[15] sont des types de réseaux de neurones basés sur des couches convolutives. Ces couches permettent de traiter l'image par région et non dans sa globalité, ce qui réduit le nombre de paramètres. La matrice de convolution, aussi appelée filtre, est alors appliquée sur une région de taille définie par le paramètre "kernel-size" et dont le pas de déplacement sur l'image équivaut au paramètre "stride". Ces différents filtres permettent d'extraire les caractéristiques de l'image utiles pour la tâche à réaliser.

Après la couche convolutive, on applique une fonction non-linéaire comme la sigmoïde, la tangente hyperbolique ou une fonction ReLU (Rectified Linear Unit). Enfin, le réseau comprend des couches de Maxpooling et linéaires. L'architecture pytorch que nous avons utilisée est visible dans la figure 12.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 16, 240, 313]	160
ReLU-2	[-1, 16, 240, 313]	0
Conv2d-3	[-1, 32, 238, 311]	4,640
ReLU-4	[-1, 32, 238, 311]	0
MaxPool2d-5	[-1, 32, 119, 155]	0
Conv2d-6	[-1, 64, 117, 153]	18,496
ReLU-7	[-1, 64, 117, 153]	0
Conv2d-8	[-1, 96, 115, 151]	55,392
ReLU-9	[-1, 96, 115, 151]	0
MaxPool2d-10	[-1, 96, 57, 75]	0
Flatten-11	[-1, 410400]	0
Linear-12	[-1, 500]	205,200,500
ReLU-13	[-1, 500]	0
Dropout-14	[-1, 500]	0
Linear-15	[-1, 2]	1,002
Total params: 205,280,190		
Trainable params: 205,280,190		
Non-trainable params: 0		
Input size (MB): 0.29		
Forward/backward pass size (MB): 108.18		
Params size (MB): 783.08		
Estimated Total Size (MB): 891.55		

FIGURE 12 – Architecture pytorch du CNN utilisée lors de notre projet

Pour entraîner notre réseau, nous avons utilisé l'optimizer SGD (Stochastic gradient descent) et comme loss function la cross entropy.

3.2. MeshCNN

MeshCNN est une forme de CNN qui manipule des maillages, c'est-à-dire sur des formes polygonales. Ce réseau de neurones est donc très utile pour toutes les applications de deep learning qui utilisent les formes géométriques en sous forme 3D. Son architecture est très proche des CNN classiques en utilisant les mêmes fonctions de convolution et de réduction de taille (pooling) sur les points saillants des formes. Un point fort de cette méthode est de faire en sorte de maintenir au mieux la forme globale des maillages lors des opérations de pooling. La méthode est disponible sur internet et a été publiée en 2019 par Rana Hanocka[16].

Le réseau de neurones prend donc en entrée un maillage pouvant appartenir à différentes classes définies lors de l'entraînement et donne en sortie une prédiction avec une valeur entre -1 et 1 pour chaque classe. On sélectionne alors la classe avec la plus haute valeur comme étant la classe prédite par le réseau. En fonction de la taille des maillages et de la taille du dataset, l'étape d'entraînement peut devenir relativement longue.

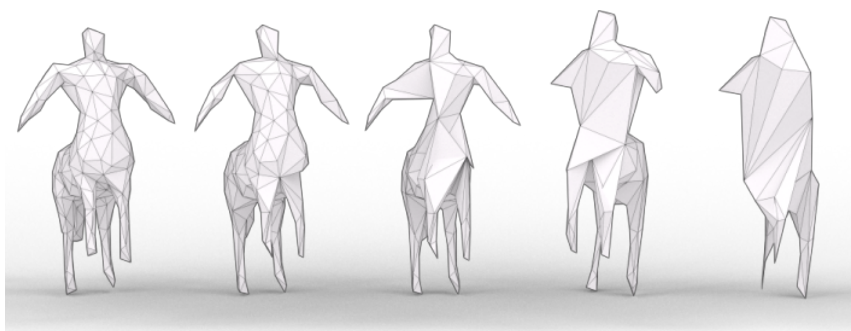


FIGURE 13 – Visualisation de différentes tailles de maillage pour une même image [16]

3.3. SVM

Les SVM (Support Vector Machine) sont un groupe de méthodes de machine learning supervisé très utilisées pour la classification non-probabiliste binaire qui se basent sur des filtres prédéfinis, contrairement au deep learning où les features sont "appries". Ces méthodes sont donc un peu plus adaptées que les réseaux de neurones pour une plus faible quantité de données, même si les résultats en dépendent également. Nous avons utilisé les méthodes de scikit-learn pour l'implémentation de SVM [17].

Parmi les avantages des SVM se trouvent la simplicité d'utilisation et la modularité. Elles sont aussi très adaptées à une grande dimensionnalité de données, comme c'est notre cas avec nos données. Une modification itérative des hyper-paramètres est mise en œuvre durant la phase d'apprentissage. Les filtres utilisés dans ces méthodes doivent être adaptés au contexte des données que l'on traite. Parmi ces défauts, on peut citer le temps de calcul qui peut être très long en fonction de la dimensionnalité des données, même si des optimisations de cet algorithme existent, notamment sa version avec des moindres carrés (LS-SVM pour Least Squared SVM) en utilisant une fonction de coût SSE (Sum Squared Error) proche de celle utilisée par des réseaux de neurones[18].

Le but est de délimiter les données grâce à un hyperplan séparateur de classes. Pour chaque "ligne" de délimitation, dans le cas 2D, on peut calculer la marge de ce délimiteur. La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Ces derniers sont appelés vecteurs supports. Dans les SVM, la frontière de séparation est choisie comme celle qui maximise la marge. Le problème est de trouver cette frontière séparatrice optimale, à partir d'un ensemble d'apprentissage, ce qui correspond à un problème d'optimisation quadratique. Un exemple de résultat fourni par SVM est présenté dans la figure 14.

Ceci est simple dans le cas où les données sont linéairement séparables, mais cela est plutôt rare. Une option est donc de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension (possiblement infinie), dans lequel il est plus probable qu'il existe une séparation linéaire. Enfin, une astuce calculatoire permet de garder un temps de calcul tout à fait raisonnable malgré le passage en plus grandes dimensions.

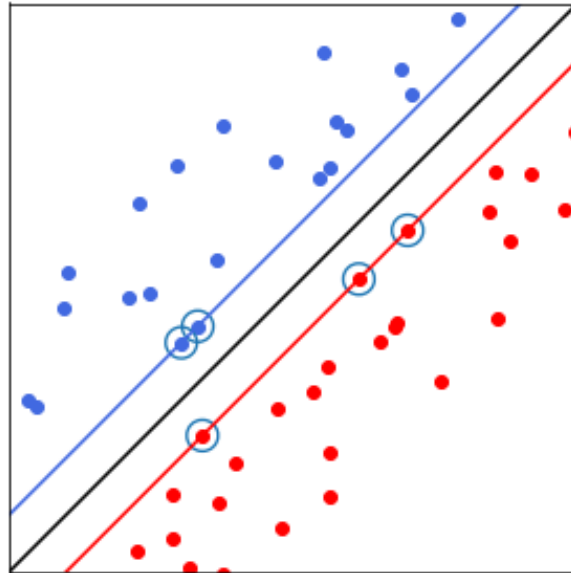


FIGURE 14 – Illustration du résultat de SVM[17]

4. Méthodologie

4.1. Données

Au début du projet, nous avons reçu des images d'arthroscanner de cartilages de fémurs et de tibias du CHU de Brest comprenant des données saines et pathologiques. Celles-ci nous ont été transmises par Agathe Yvinou, interne en chirurgie orthopédique au CHU de Brest. Dans le tableau 1 se trouve la répartition de ces données. Pour réaliser manuellement la labellisation sain/pathologique des données, nous avons obtenu les comptes-rendus des diagnostics réalisés par un professionnel du CHU.

Données CHU de Brest		
	Tibia	Fémur
Pathologique	26	21
Sain	11	13

TABLE 1 – Données fournies par le CHU de Brest

Afin d'améliorer l'efficacité de nos méthodes et pour enrichir notre entraînement, nous avons cherché d'autres données. Nous avons trouvé, avec l'aide de Pierre-Henri Conze, une banque de données issue du site de Stanford AIMI[19], nommée SKM-TEA. Cette banque fournit 155 images et segmentations d'arthroscanner du genou (gauche ou droit) qui sont annotées pour 16 différentes pathologies. Elles ont été acquises directement à l'université de Stanford par échantillonnage IRM elliptique avec 8 ou 15 couronnes. Leur répartition se trouve dans le tableau 2.

Pour labelliser ces données, nous avons utilisé les métadonnées présentes sur le site et avons croisé 2 types d'information :

- le type de pathologies : nous avons sélectionné les pathologies correspondant à la détérioration du cartilage (n°12,13,14 et 15 dans les pathologies des métadonnées)
- les tissus touchés par la pathologie : nous avons sélectionné uniquement les cartilages fémoraux et tibiaux (n°4 et 6 dans les tissus des métadonnées).

Données Stanford		
	Tibia	Fémur
Pathologique	118	84
Sain	37	71

TABLE 2 – Données trouvées sur le site de Stanford

Données totales

Données totales		
	Tibia	Fémur
Pathologique	144	105
Sain	48	84

TABLE 3 – Données totales pour la réalisation du projet

Ainsi, pour la partie suivante "Préparation des données", on part des maillages de cartilages de fémurs et de tibias obtenus après segmentation des données 3D.

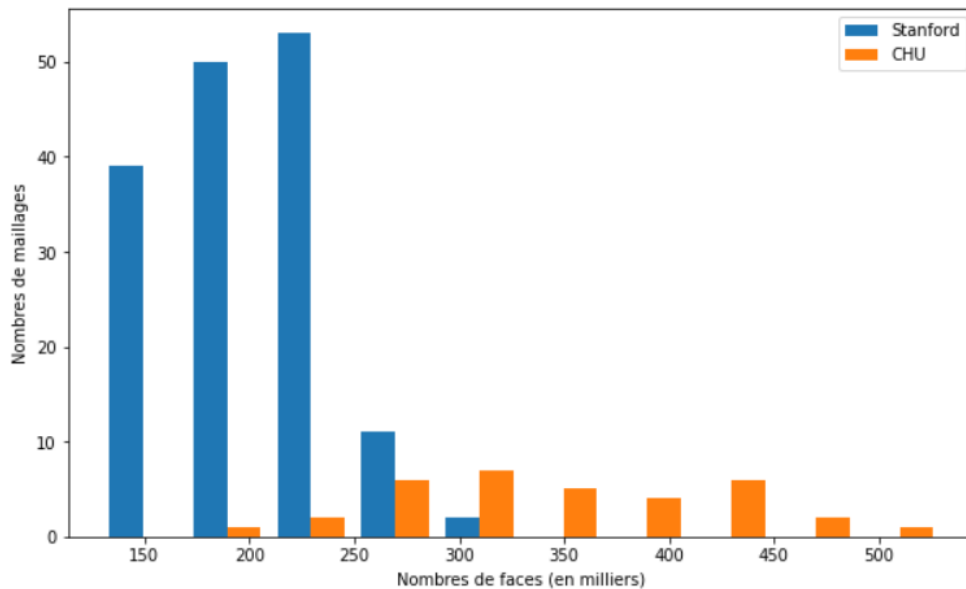


FIGURE 15 – Histogramme du nombre de faces des maillages de cartilages de fémur

4.2. Préparation des données

À partir des maillages, nous avons réalisé une première étape de "nettoyage", pour retirer des impuretés liées à la segmentation et afin d'obtenir un maillage avec une seule composante connectée (ou deux dans le cas des cartilages de tibia). Ces impuretés sont invisibles à l'œil et peuvent sembler négligeables par rapport à la taille des maillages, mais peuvent être sources d'erreurs dans la suite des traitements.

La préparation des données vise à créer deux types de format final pour les données : des données sous forme de maillages 3D et des données 2D sous formes de cartes d'épaisseur. La première est une forme plus lourde et complexe à manipuler, mais celle qui contient le plus d'information. La seconde, quant à elle, est plus légère et classique à traiter, mais une partie de l'information est perdue. En fonction du format, nous avons utilisé différents outils de classification.

4.2.1. MeshCNN

La préparation des données pour utiliser MeshCNN est décrite sur le github associé [20]. Pour l'instant, le réseau MeshCNN ne supporte que des maillages avec une seule composante connectée, ainsi l'étape ci-dessus de "nettoyage" est particulièrement importante. Sinon, comme décrit sur la page, il faut convertir nos maillages (initialement sous forme STL) en format OBJ.

Il faut aussi égaliser la taille des maillages. De toute manière, les maillages originaux sont trop massifs pour être traités tels quels. Après plusieurs tentatives, nous avons trouvé que réduire les maillages à 5000 faces était un compromis honnête pour limiter la perte d'information et ne pas saturer la mémoire lors de l'entraînement.

Enfin, MeshCNN supporte les maillages non-manifold et nous n'avons pas réussi à corriger les "mauvais" triangles de nos maillages. Cependant, cela est en fait fortement conseillé. Il s'agit donc d'une piste d'amélioration potentiellement importante.

Pour réaliser l'entraînement, il faut créer manuellement un dossier pour chaque classe, ainsi que des sous-dossiers "test" et "train" dans lesquels on vient placer les fichiers sélectionnés.

4.2.2. Cartes d'épaisseur

Pour certaines méthodes, nous avons transformé les segmentations en cartes d'épaisseur dont on peut voir des exemples dans les figures 19 et 20 grâce à un travail réalisé par une stagiaire au Latim. Il s'agit de projections en 2D de l'épaisseur des cartilages.

Nous avons ensuite effectué un effet miroir sur les cartes ainsi obtenues pour régulariser genoux gauches et genoux droits. Cependant, pour certaines données, notamment la banque de données venant de Stanford, nous n'avons pas la distinction genoux droits ou genoux gauches déjà effectuée. Nous avons dans un premier temps cherché à identifier à l'œil le côté analysé, mais cette tâche s'est avérée fastidieuse. En effet, la différence est parfois peu discernable, surtout pour des élèves non experts en anatomie, et la banque de données de Stanford comportant 155 patients différents, la reconnaissance droite/gauche est très longue. Nous avons donc décidé de faire une PCA sur les cartes d'épaisseurs et d'afficher la première composante pour différencier les deux côtés.

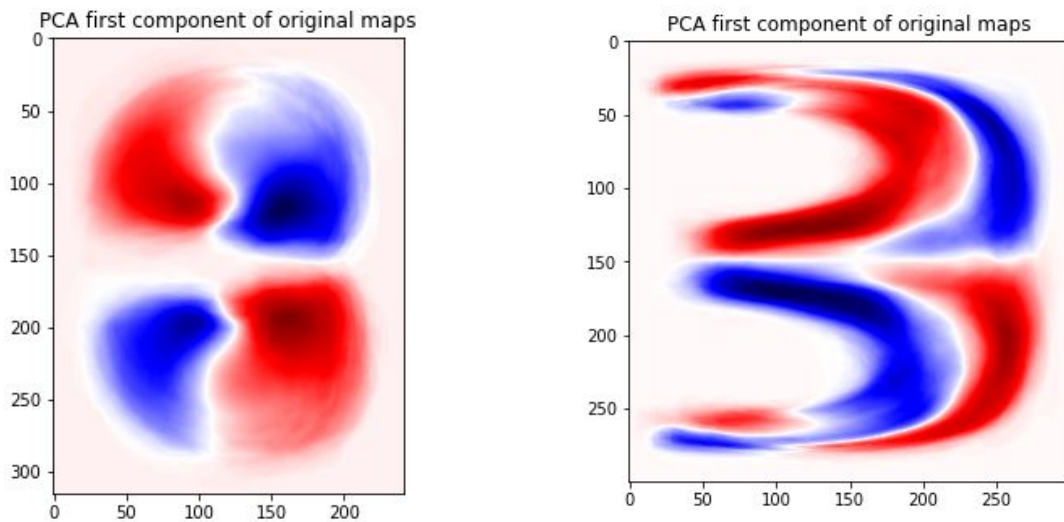


FIGURE 16 – Visualisation de la première composante sur les cartes d'épaisseur du cartilage du tibia et du fémur (rouge : valeur positive, bleu : valeur négative)

Nous avons vérifié ces résultats en affichant les cartes moyennes ayant une valeur positive pour la première composante et celles ayant une valeur négative au niveau des tibias et des fémurs. Puis nous avons effectué un effet miroir sur une des deux cartes. On remarque qu'on a alors une carte très similaire à celle qui n'a pas été retournée.

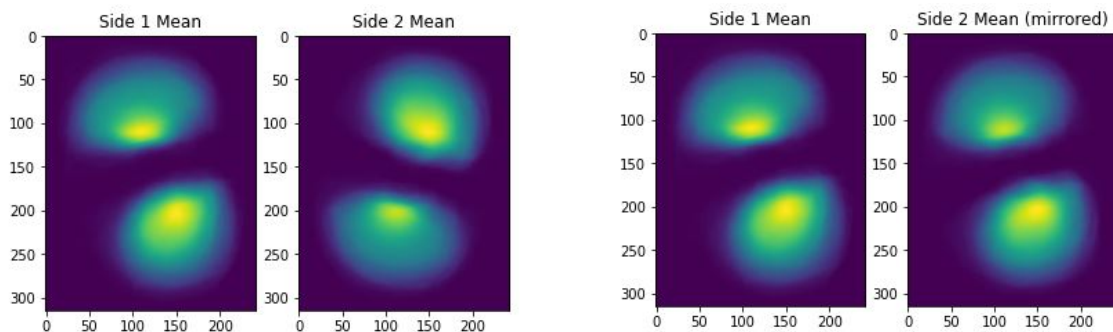


FIGURE 17 – Visualisation des cartes moyennes pour les tibias

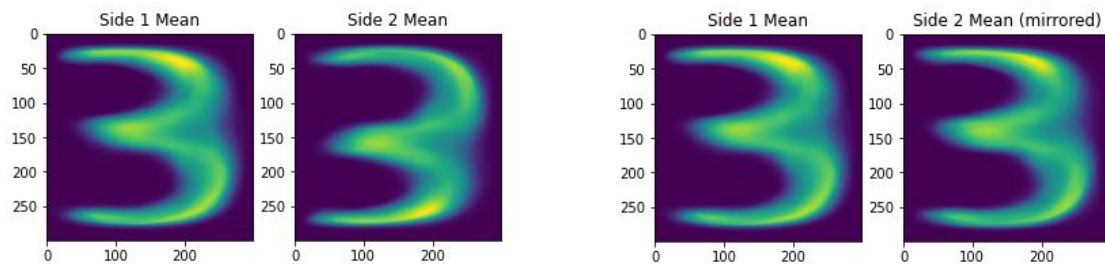


FIGURE 18 – Visualisation des cartes moyennes pour les femurs

De plus, nous avons enregistré sous 2 formats différents chaque carte d'épaisseur. En format png, qui donne à chaque épaisseur une valeur entière entre 0 et 255, et en format tiff, dans lequel les épaisseurs sont directement enregistrées. Le premier sert de format visuel, alors que le tiff permet de conserver les valeurs exactes des épaisseurs de nos cartes.

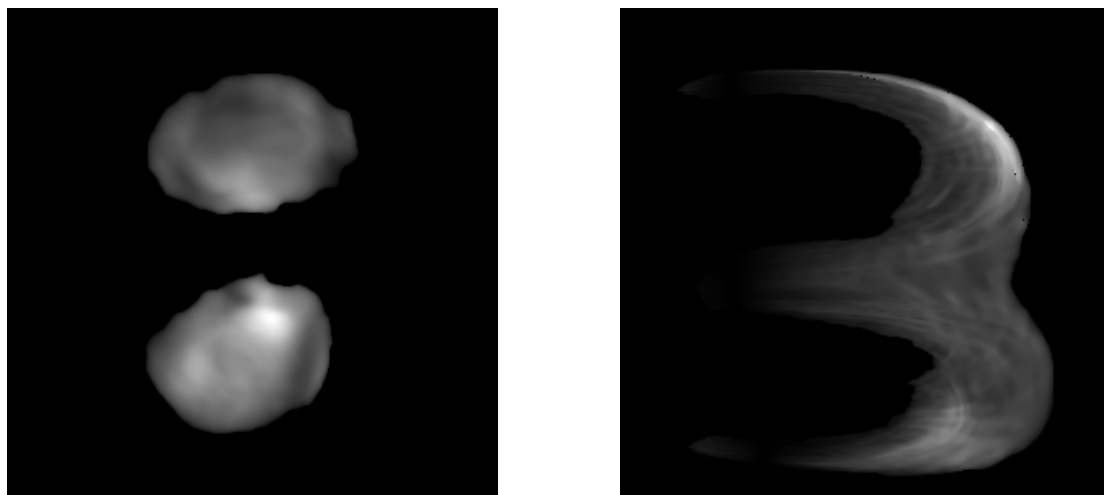


FIGURE 19 – Cartes d'épaisseur du cartilage du tibia et du fémur chez un patient sain

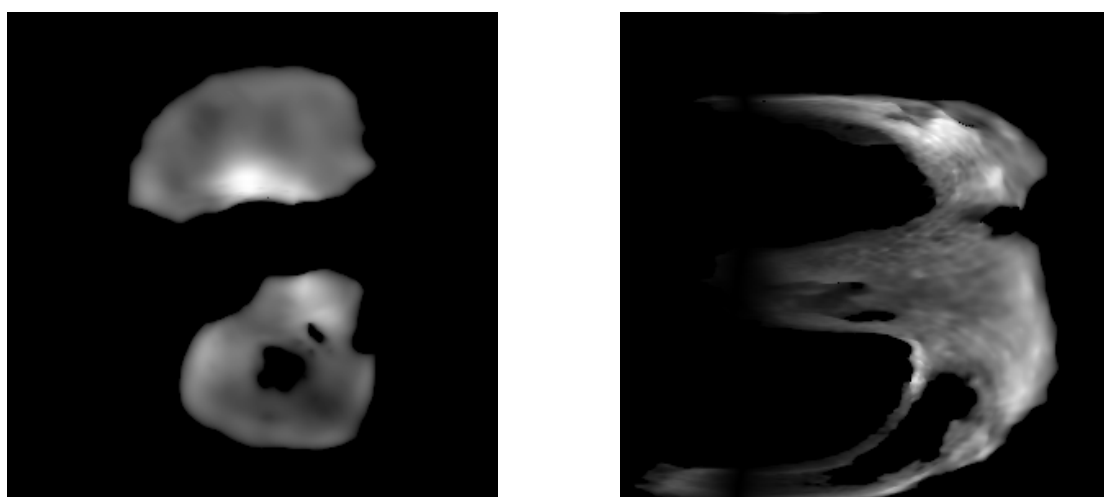


FIGURE 20 – Cartes d'épaisseur du cartilage du tibia et du fémur pour un genou pathologique

Finalement, dans l'optique d'améliorer la correspondance spatiale des différents pixels des cartes d'épaisseurs, nous avons aussi réalisé un recalage affine et rigide sur les cartes.

4.2.3. Recalage 3D

Pour effectuer une ACP sur les maillages 3D de cartilage, nous avons d'abord besoin de recaler les cartilages pour avoir la correspondance des points saillants de chaque cartilage afin de les aligner dans une matrice à partir de laquelle nous pouvons effectuer notre analyse en composantes principales.

Nous avons d'abord procédé à un recalage rigide avec la méthode ICP (Iterative Closest Point). L'ICP a pour but de minimiser itérativement la distance entre les points des maillages. On peut distinguer ainsi plusieurs étapes :

- Estimation de la correspondance entre les points
- Calcul de la transformation rigide (rotation et translation) qui permet de minimiser la distance entre les points
- Application de cette transformation sur les données.

La figure 21 montre un exemple de recalage entre deux fémurs.

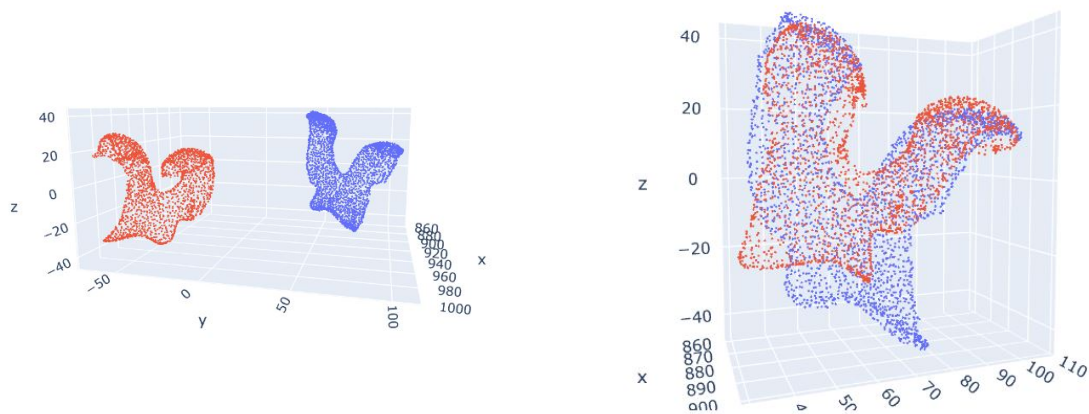


FIGURE 21 – Nuages de points des cartilages de 2 fémurs avant et après recalage (15 itérations)

Nous avons ensuite appliqué une transformation non-rigide à l'aide de la librairie probreg de python.[22]. Pour calculer cette transformation, nous utilisons la méthode CPD (Coherent Point Drift) [23].

Dans cette méthode, les 2 nuages de points sont vus de manière probabiliste et le recalage correspond à un problème d'estimation de la densité de probabilité de ces nuages. Les centroïdes GMM (Gaussian Mixture Model basé sur une distribution gaussienne des données) sont ainsi adaptés aux données en maximisant la probabilité.

Cependant, nous avons eu quelques problèmes lors de la réalisation de cette étape. En effet, après avoir essayé de modifier les paramètres de la transformation non-rigide (tels que β et lmd), nous ne sommes pas arrivés à des résultats concluants.

4.3. Métriques d'évaluation

Pour chaque métrique d'évaluation utilisée pour tester l'efficacité de nos méthodes, nous nous basons sur 2 matrices de confusion dépendantes de la classe étudiée :

		Ground truth		Total
		Pathologique	Sain	
Prediction	Pathologique	TP	FP	$TP + FP$
	Sain	FN	TN	$FN + TN$
Total		$TP + FN$	$FP + TN$	

TABLE 4 – Matrice de confusion pour la classe *Pathologique*

		Ground truth		Total
		Sain	Pathologique	
Prediction	Sain	TP	FP	$TP + FP$
	Pathologique	FN	TN	$FN + TN$
Total		$TP + FN$	$FP + TN$	

TABLE 5 – Matrice de confusion pour la classe *Sain*

Celles-ci indiquent le total d'images saines et pathologiques qui ont été prédites saines ou pathologiques en fonction de leur vérité terrain labellisée par un expert.

Accuracy

Ce score donne le pourcentage de positifs et de négatifs bien identifiés :

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN}$$

Mais ce score n'étant pas suffisant pour bien cerner la validité des résultats, nous allons utiliser d'autres métriques qui sont décrites ci-dessous.

Recall

Le score **recall** donne le nombre de positifs bien prédits sur le nombre total de positifs. En d'autres termes, il donne le pourcentage de positifs qui ont été reconnus par l'algorithme. S'il indique une bonne maîtrise de notre méthode à détecter les positifs, ce score ne témoigne aucunement de la capacité à prédire les négatifs.

$$RECALL = \frac{TP}{TP + FN}$$

Precision

Le score **precision** donne le nombre de positifs bien prédits sur le nombre total de prédictions positives. En d'autres termes, il donne le pourcentage de prédictions positives correctes.

$$PRECISION = \frac{TP}{TP + FP}$$

Score F1

[21]

Parce que les 2 métriques précédentes ne permettent pas d'évaluer entièrement nos modèles, nous utilisons aussi le **score F1**. C'est une moyenne harmonique qui donne une très bonne évaluation de l'efficacité de nos algorithmes.

$$F1 = 2 * \frac{RECALL * PRECISION}{RECALL + PRECISION}$$

5. Résultats

L'ensemble des résultats obtenus sont disponibles à travers notre répertoire en ligne. Dans cette section, nous présentons les "meilleurs" résultats obtenus pour les différentes méthodes de classifications que nous avons essayées.

Pour chaque approche, le dataset était divisé en données d'entraînement et données tests. Les résultats rapportés ci-dessous concernent les données tests, et donc évaluent la capacité des modèles à inférer sur de nouvelles données. On notera cependant qu'il ne s'agit pas nécessairement des mêmes répartitions entraînement / test à travers les différentes méthodes.

Enfin, on notera que, dans les résultats ci-dessous, les classifications sain / pathologique sont systématiquement réalisées séparément pour les cartilages de fémurs et les cartilages de tibias.

5.1. Réseaux de neurones

Dans cette partie, nous présentons les approches par réseaux de neurones convolutifs : MeshCNN, qui utilise les données sous forme de maillages 3D, puis le CNN que nous avons implémenté, qui traite des données sous forme de cartes d'épaisseur.

5.1.1. MeshCNN

Les résultats présentés ici correspondent à un modèle entraîné avec les paramètres décrits dans la figure ci-dessous :

Paramètres	Valeurs
<i>ninput_edges</i>	8000
<i>ncf (filtres)</i>	[64, 128, 256]
<i>pool_res (nb faces)</i>	[7000, 5000, 3000]
<i>lr (initial learning rate)</i>	0.0002
<i>batch_size</i>	16
<i>num_aug</i>	20
<i>epochs (total)</i>	457

FIGURE 22 – Paramètres d'entraînement MeshCNN

		Ground truth	
		Sain	Pathologique
Prediction	Sain	22	14
	Pathologique	9	10

TABLE 6 – Matrice de confusion pour la classe *Sain*, Mesh CNN sur Fémur

Calcul des scores pour les sains :

- Recall : 0,71
- Precision : 0,61
- F1 score : 0,66

Calcul des scores pour les pathologiques :

- Recall : 0,42
- Precision : 0,53
- F1 score : 0,47

Remarques générales :

- MeshCNN n'a été appliqué que sur les cartilages de fémur, car ceux des tibias doivent être séparés en partie latérale et partie médiale pour être utilisé, mais nous n'avons pas de labellisation distincte pour ces deux parties.
- Les entraînements ont été réalisés avec un GPU sur Google Colab.
- Les résultats du modèle ci-dessus ne permettent pas de conclure à une classification effective de sa part.

Difficultés rencontrées :

- Des temps d'entraînement relativement longs réduisent le nombre de tentatives d'entraînements et de tests.
- Difficile de tuner les hyper-paramètres et même simplement d'en sélectionner des fonctionnels.

Pistes d'amélioration :

- Il est très probablement possible d'obtenir de meilleurs hyper-paramètres.
- Rendre les maillages manifolds (préparation des données).
- De plus grands maillages en entrées (donc un meilleur hardware) pourraient améliorer les résultats.

5.1.2. CNN***Fémur***

		Ground truth	
		Sain	Pathologique
Prediction	Sain	20	7
	Pathologique	19	17

TABLE 7 – Matrice de confusion pour la classe *Sain*, CNN sur Fémur

Calcul des scores pour les sains :

- Recall : 0,51
- Precision : 0,74
- F1 score : 0,60

Calcul des scores pour les pathologiques :

- Recall : 0,71
- Precision : 0,47
- F1 score : 0,57

Tibia

		Ground truth	
		Sain	Pathologique
Prediction	Sain	4	7
	Pathologique	9	38

TABLE 8 – Matrice de confusion pour la classe *Sain*, CNN sur Tibia

Calcul des scores pour les sains :

- Recall : 0.31
- Precision : 0.36
- F1 score : 0.33

Calcul des scores pour les pathologiques :

- Recall : 0.84
- Precision : 0.81
- F1 score : 0.82

Remarques générales :

- Les entraînements ont été réalisés avec un GPU sur Google Colab avec une taille de batch de 20. (L'augmentation de la taille du batch n'améliore pas les résultats et rallonge le temps de calcul.)
- Les résultats du modèle ci-dessus ne permettent pas de conclure à une classification effective de sa part. En effet, les résultats sur les fémurs ne sont pas très concluants et au niveau des tibias, même si le modèle semble avoir une performance correcte pour les tibias pathologiques, les résultats sur les tibias sains ne sont pas satisfaisants.

Pistes d'amélioration :

- Comme l'utilisation de réseau de neurones nécessite un grand nombre de données d'entraînement pour que ces derniers soient performants, avec un nombre de données plus grand, il serait possible d'améliorer les performances de notre modèle.
- De plus, une optimisation de l'architecture de notre CNN ou l'utilisation de réseaux de type U-Net pourraient s'avérer utiles.

5.2. Machine learning "classique"

Dans cette partie, nous présentons les résultats avec une approche de classification par machine learning plus classique. En effet, nous avons profité de nos tests avec l'approche SVM pour tester d'autres méthodes de machine learning dit "classique", par opposition aux méthodes de la partie précédente. À cette fin, nous avons utilisé la librairie python Scikit-Learn pour implémenter les approches SVM, KNN, SGD classifieur, Naive Bayes et Random Forest.

Nous avons appliqué ces approches de classification avec les données sous forme de cartes d'épaisseur, mais aussi sur les cartes d'épaisseur après les avoir réduites, pour chacune des méthodes de réduction : ACP, NMF et NTF.

De nombreux tests ont été réalisés en faisant varier différents éléments, et notamment les cartes d'épaisseur : sans recalage, avec recalage rigide et avec recalage affine. Les figures ci-dessous résument les éléments les plus pertinents concernant les entraînements réalisés. Les scores affichés correspondent à une moyenne des scores F1 des deux classes. De plus, le calcul du Training score est obtenu par une moyenne sur plusieurs itérations de cross-validation (donc pas au score sur l'ensemble des données d'entraînement).

Remarques générales :

- Les classifieurs obtenus sont globalement médiocres.
- On ne note pas de discriminations significatives pour le recalage ou non des données, pour le choix de la méthode de réduction, ni pour d'autres hyper-paramètres tels que la taille des données réduites ou le ratio des données test / train par exemple.
- On constate que le SVM et le SGD classifieur ressortent globalement vis-à-vis des autres méthodes.

Difficultés rencontrées :

- Les méthodes de réductions NMF et TNF demandent un temps de calcul considérable et croissant avec le nombre de composantes souhaitées. Ainsi, cela a très vite borné le nombre de composantes que l'on a pu tester, et de plus, ces méthodes n'ont pas toujours convergé (par manque d'itérations) dans les résultats présentés.

Pistes d'amélioration :

- Se concentrer sur une seule méthode et sur l'optimisation de ses hyper-paramètres pourrait permettre d'améliorer nos résultats.

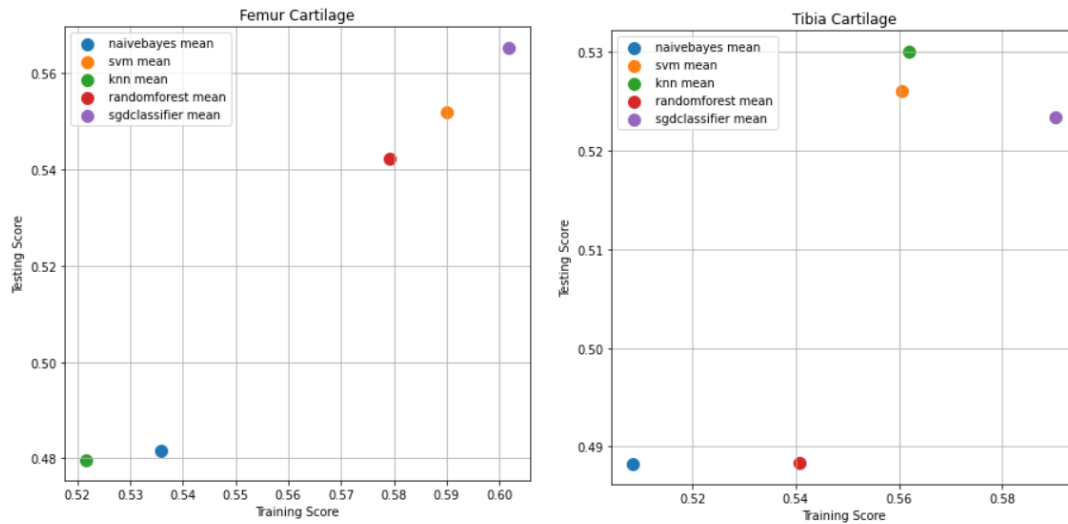


FIGURE 23 – Résultats moyens des entraînements des différents classifieurs

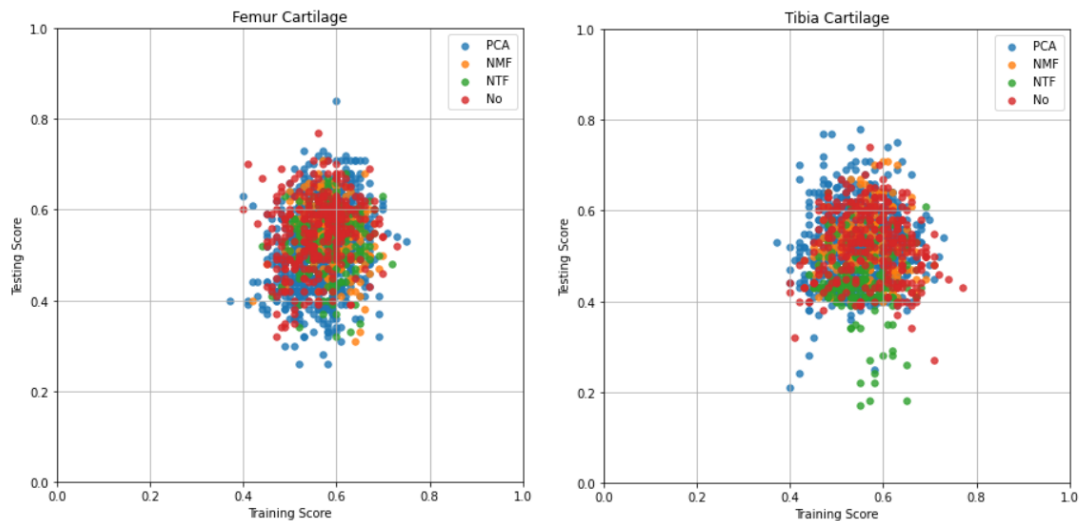


FIGURE 24 – Résultats des entraînements labellisés par méthode de réduction utilisée

- Analyser les résultats du point de vue des données (à quelle fréquence chacune est bien classifiée par exemple) pourrait permettre d'obtenir un insight pertinent pour analyser le rôle des données et de leur labellisation dans l'échec de nos méthodes.

Dans les parties ci-dessous, de "bons" classifieurs ont été sélectionnés pour illustrer les résultats obtenus avec les différentes méthodes de réductions.

5.2.1. ACP***Fémur***

		Ground truth	
		Sain	Pathologique
Prediction	Sain	17	3
	Pathologique	7	11

TABLE 9 – Matrice de confusion pour la classe *Sain*, ACP sur Fémur

Calcul des scores pour les sains :

- Recall : 0,71
- Precision : 0,85
- F1 score : 0,77

Calcul des scores pour les pathologiques :

- Recall : 0,79
- Precision : 0,61
- F1 score : 0,69

Tibia

		Ground truth	
		Sain	Pathologique
Prediction	Sain	32	8
	Pathologique	10	26

TABLE 10 – Matrice de confusion pour la classe *Sain*, ACP sur Tibia

Calcul des scores pour les sains :

- Recall : 0,76
- Precision : 0,8
- F1 score : 0,78

Calcul des scores pour les pathologiques :

- Recall : 0,76
- Precision : 0,72
- F1 score : 0,74

5.2.2. NMF***Fémur***

La méthode ayant donné les meilleurs résultats pour le fémur avec une NMF est le random forest et donne la matrice de confusion suivante :

		Ground truth	
		Sain	Pathologique
Prediction	Sain	28	7
	Pathologique	6	15

TABLE 11 – Matrice de confusion pour la classe *Sain*, NMF sur Fémur

Calcul des scores pour les sains :

- Recall : 0.82
- Precision : 0.8
- F1 score : 0.81

Calcul des scores pour les pathologiques :

- Recall : 0.68
- Precision : 0.71
- F1 score : 0.70

Tibia

La méthode ayant donné les meilleurs résultats pour le tibia avec une NMF est le random forest et donne la matrice de confusion suivante :

		Ground truth	
		Sain	Pathologique
Prediction	Sain	71	11
	Pathologique	3	10

TABLE 12 – Matrice de confusion pour la classe *Sain*, NMF sur Tibia

Calcul des scores pour les sains :

- Recall : 0,96
- Precision : 0,87
- F1 score : 0,91

Calcul des scores pour les pathologiques :

- Recall : 0,48
- Precision : 0,77
- F1 score : 0,59

5.2.3. TNF**Fémur**

Les meilleurs résultats avec le TMF ont été obtenus avec sgdcclassifier :

		Ground truth	
		Sain	Pathologique
Prediction	Sain	40	15
	Pathologique	10	30

TABLE 13 – Matrice de confusion pour la classe *Sain*, TNF sur Fémur

Calcul des scores pour les sains :

- Recall : 0.8
- Precision : 0.73
- F1 score : 0.76

Calcul des scores pour les pathologiques :

- Recall : 0.67
- Precision : 0.75
- F1 score : 0.71

Tibia

		Ground truth	
		Sain	Pathologique
Prediction	Sain	38	8
	Pathologique	0	10

TABLE 14 – Matrice de confusion pour la classe *Sain*, TNF sur Tibia

Calcul des scores pour les sains :

- Recall : 1
- Precision : 0.83
- F1 score : 0.91

Calcul des scores pour les pathologiques :

- Recall : 0.56
- Precision : 1
- F1 score : 0.71

5.2.4. Sans méthode de réduction

Fémur

La méthode ayant donné les meilleurs résultats sans méthode de réduction de dimension est le sgdcclassifier et donne la matrice de confusion suivante :

		Ground truth	
		Sain	Pathologique
Prediction	Sain	32	8
	Pathologique	10	26

TABLE 15 – Matrice de confusion pour la classe *Sain*, Sans réduction sur Fémur

Calcul des scores pour les sains :

- Recall : 0.76
- Precision : 0.80
- F1 score : 0.78

Calcul des scores pour les pathologiques :

- Recall : 0.76
- Precision : 0.72
- F1 score : 0.74

Tibia

La méthode ayant donné les meilleurs résultats sans méthode de réduction de dimension est le svm et donne la matrice de confusion suivante :

		Ground truth	
		Sain	Pathologique
Prediction	Sain	24	5
	Pathologique	3	6

TABLE 16 – Matrice de confusion pour la classe *Sain*, Sans réduction sur Tibia

Calcul des scores pour les sains :

- Recall : 0.89
- Precision : 0.83
- F1 score : 0.86

Calcul des scores pour les pathologiques :

- Recall : 0.54
- Precision : 0.67
- F1 score : 0.6

5.3. Récapitulatif des résultats

F1 score	MeshCNN	CNN	ACP	NMF	TNF	Sans réduction
Fémurs sains	0.66	0.60	0.77	0.81	0.76	0.78
Fémurs pathologiques	0.47	0.57	0.69	0.70	0.71	0.74
Tibias sains	/	0.33	0.78	0.91	0.91	0.86
Tibias pathologiques	/	0.82	0.74	0.59	0.71	0.60

Remarques générales :

- Les méthodes de machine learning "classiques" semblent plus performantes que celles de deep learning avec MeshCNN et CNN.
- Les différentes méthodes testées semblent aussi plus performantes pour la correcte classification des genoux sains que pathologiques.

6. Conclusions

Ainsi, à travers ce projet, nous nous sommes intéressés à la mise en place de méthodes de machine learning sur les images médicales de cartilages fémoraux-tibiaux, afin d'obtenir une classification de la forme sain/pathologique. Pour fonctionner au mieux, ce genre d'approches nécessitent un maximum de données. Ainsi, après de premiers tests infructueux sur le petit dataset fourni par le CHU de Brest, nous l'avons complété par les données du SKM-TEA dataset. Nous avons donc créé un dataset commun, lui avons associé un fichier de métadonnées, et nous avons réalisé différentes étapes de traitement de données qui ont abouti à deux formes finales des données : maillages 3D et cartes d'épaisseur. Enfin, nous avons appliqué et comparé différentes méthodes de classification dont certaines combinées à des méthodes de réduction / décomposition de données.

Les résultats que nous avons obtenus ne nous ont pas permis de mettre en évidence ce que nous recherchions : un classifieur sain/pathologique performant. Cependant, les tests de classification par machine learning "classique" nous ont permis de confirmer que l'approche SVM semble être l'une des plus prometteuses à creuser. De plus, à travers ce type de méthodes, les résultats avec et sans réduction de dimensions n'ont pas montré de différences significatives. Ainsi, l'ajout de méthodes de réduction de dimension ne semble pas améliorer les performances, mais augmente considérablement le temps de calcul pour les approches NMF et TNF.

Parmi les difficultés principales que nous avons rencontrées, il y a le recalage 3D qui nous aurait permis de tester davantage de méthodes de classification sans passer par les cartes d'épaisseur, mais que nous n'avons pas réussi à mettre en place. De plus, la labellisation des cartilages de tibias ne nous indiquait pas, dans le cas pathologique, quelle partie parmi la latérale et la médiale est pathologique. Cela nous a empêché d'utiliser MeshCNN avec ces données.

Nous pouvons identifier plusieurs pistes d'améliorations potentielles dans notre démarche, dans l'optique d'obtenir de meilleurs résultats. Tout d'abord, une amélioration assez directe serait l'optimisation des hyper-paramètres de nos modèles, en particulier pour les approches de type réseau de neurones, pour lesquelles nous avons réalisé relativement peu de tests. Ensuite, au niveau des données, on peut remarquer que nous n'avons appliqué aucune méthode sur les données "complètes", qu'il s'agisse des projections 2d (cartes d'épaisseur) ou des maillages 3d qui ont été réduits. Ainsi, nos résultats médiocres peuvent potentiellement être reliés à une perte d'informations sur les données manipulées. Une meilleure puissance de calcul pourrait permettre de manipuler des données de plus grandes tailles, et donc, potentiellement, de pallier ce problème. Enfin, on peut se demander si le mode de classification choisie sain/pathologique est tout à fait pertinent, et si une labellisation sur l'intensité de la pathologie, par exemple, ne serait pas plus adaptée.

7. Management de projet

7.1. Organisation, planification et répartition des tâches

Nous avons défini plusieurs rôles et objectifs dès le début du projet en accord avec les tuteurs. Nous avons donc pris en compte les compétences et les envies de chacun pour désigner des responsables de tâches. Bien sûr ces responsables ne sont en aucun cas les seuls effecteurs des tâches mais simplement des référents vers qui se tourner en cas de questions ou de remarques sur une partie spécifique du projet.

Pour planifier les tâches, nous avons utilisé l'application **Trello** qui permet grâce à son interface intuitive (Figure 25) de créer des tâches et de les classer dans les différentes colonnes (à faire, en cours, bloqué, terminé, ...) et d'attribuer un responsable à chacune de ces tâches. Sur la gestion à plus long terme nous avons réalisé un diagramme de Gantt (Figure 26) que nous actualisons chaque mois. Cela nous permet de grader une vue sur ce qui a été fait et sur les objectifs à réaliser dans les prochains mois.

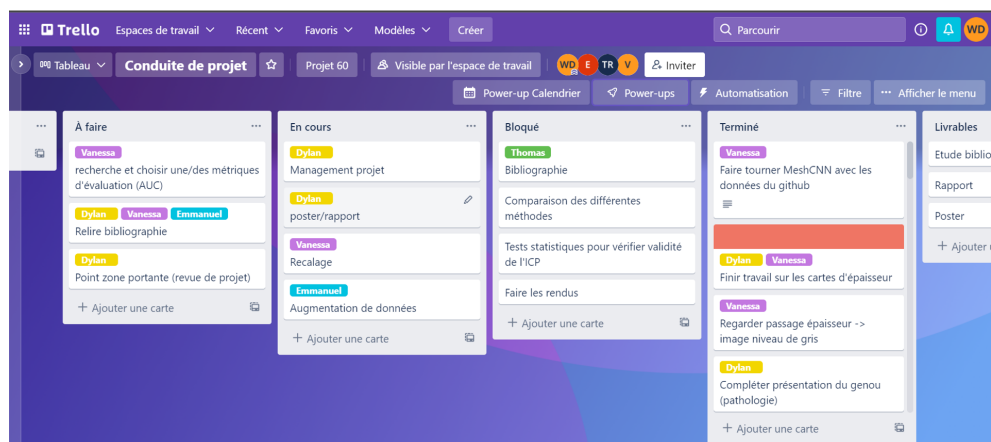


FIGURE 25 – Exemple du trello utilisé par l'équipe

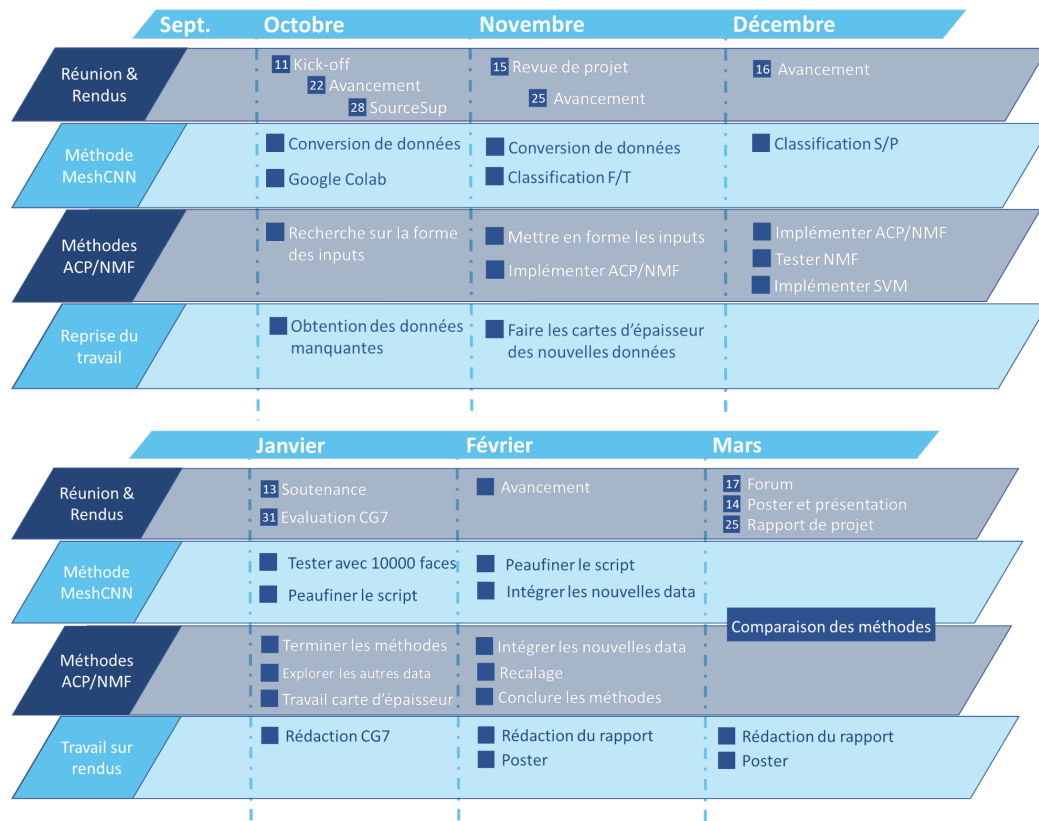


FIGURE 26 – Diagramme de Gantt

7.2. Identification des risques

Dès le début du projet nous avons identifié certains risques qui auraient pu survenir ou qui sont survenus lors de son déroulement :

- Inefficacité des méthodes utilisées (MeshCNN ou SVM par exemple)
- Manque de communication
- Démotivation d'un membre de l'équipe
- Dépassement du planning

Dresser cette liste nous a permis d'identifier les difficultés qui sont survenues et de les traiter les plus rapidement et efficacement que possible.

7.3. Communication interne et externe

Nous avons cherché à utiliser les outils les plus performants pour permettre une communication la plus efficace possible entre les membres de l'équipe mais aussi avec les tuteurs du projet :



Une conversation **Messenger** a été mise en place pour la communication informelle, et l'échange d'information rapide (lieu et heure de réunion, update d'avancement, échange en distanciel,...)



Un Drive **Google** a été mis en place pour le stockage des préparations de réunions, de certaines données, des comptes rendus de réunions, des codes développés, de bibliographies ... uniquement utilisé par les membres de l'équipe



Un **Gitlab** a été utilisé pour échanger les codes avec les tuteurs projet



Pour faciliter les échanges et permettre à tous les membres d'obtenir les informations, nous avons mis en place une mailing list : projet3a-60@imt-atlantique.net



Pour délivrer tous les rendus finaux du projet, nous avons utilisé **SourceSup** qui permet comme Gitlab de déposer des codes et autres documents importants du projet.

Références

- [1] Thibault Lagorce, Jacques Buxeraud, Xavier Guillot, *Comprendre l'arthrose*, Actualités Pharmaceutiques, Volume 55, Issue 555, Pages 18-22, Avril 2016
- [2] Thibault Lagorce, Jacques Buxeraud, Xavier Guillot, *Les traitements de l'arthrose*, Actualités Pharmaceutiques, Volume 55, Issue 555, Pages 23-29, Avril 2016
- [3] Noble, P. C., Conditt, M. A., Cook, K. F., Mathis, K. B. (2006). The John Insall Award : Patient Expectations Affect Satisfaction with Total Knee Arthroplasty. *Clinical Orthopaedics and Related Research*, 452, 35-43. <https://doi.org/10.1097/01.blo.0000238825.63648.1e>
- [4] Guillaume D., Valérie B., Latim, *FollowKnee : Prothèses du genou spécifique patient*, url : <https://followknee.com>, lastchecked : 18.11.2021, originalyear : 18.02.2011
- [5] Clinique Parisienne de la Hanche et du Genou. (2021, août 19). Anatomie du genou : os, cartilage, muscles, tendons, ménisques, nerfs. <https://www.chirurgie-hanchegenou.fr/anatomie-genou>
- [6] Qu'est-ce que l'arthrose du genou? (2021, 29 juin). ameli.fr | Assuré. <https://www.ameli.fr/assure/sante/themes/arthrose-genou/definition-facteurs-favorisants>
- [7] Qu'est ce que l'arthrose du genou? (2021). IML - IML. <https://www.iml.care/details-qu+est+ce+que+l+arthrose+du+genou-178.html>
- [8] Dupaire, F. (2020, 14 juin). La chirurgie du genou : quand opérer et quelle prothèse ? Chirurgie orthopédique et traumatologie à Rennes. <https://www.chirurgie-orthopedique-rennes.fr/la-chirurgie-du-genou-quand-operer-et-quelle-prothese/>
- [9] Wikipedia contributors. (2022, 8 mars). Analyse en composantes principales. Wikipedia. https://fr.wikipedia.org/wiki/Analyse_en_composantes_principales
- [10] Wikipedia contributors. (2022b, mars 12). Non-negative matrix factorization. Wikipedia. https://en.wikipedia.org/wiki/Non-negative_matrix_factorization
- [11] Andreas Janecek, A. J., Ying Tan, Y. T. (2011). Iterative improvement of the Multiplicative Update NMF algorithm using nature-inspired optimization. *Seventh International Conference on Natural Computation*. <https://doi.org/10.1109/ICNC.2011.6022356>
- [12] Huang, C., Chen, X., Cao, S., Zhang, X. (2016). Muscle-tendon units localization and activation level analysis based on high-density surface EMG array and NMF algorithm. *Journal of Neural Engineering*, 13(6), 066001. <https://doi.org/10.1088/1741-2560/13/6/066001>
- [13] Padilla, P., Lopez, M., Gorriz, J. M., Ramirez, J., Salas-Gonzalez, D., Alvarez, I. (2012). NMF-SVM Based CAD Tool Applied to Functional Brain Images for the Diagnosis of Alzheimer's Disease. *IEEE Transactions on Medical Imaging*, 31(2), 207-216. <https://doi.org/10.1109/tmi.2011.2167628>
- [14] TensorLy : Tensor Learning in Python — TensorLy : Tensor Learning in Python. (s. d.). Tensorly. Consulté le 24 février 2022, à l'adresse <http://tensorly.org/stable/index.html>

- [15] Saad Albawi, Tareq Abed Mohammed, Saad Al-Zawi, *Understanding of a convolutional neural network*, ICET International Conference on Engineering and Technology, pp. 1-6, Août 2017
- [16] Rana Hanocka, R. H. (2019). MeshCNN. MeshCNN. <https://ranahanocka.github.io/MeshCNN>
- [17] 1.4. Support Vector Machines. (2007). Scikit-Learn. <https://scikit-learn.org/stable/modules/svm.html>
- [18] Haifeng Wang, Dejin Hu. (2005). Comparison of SVM and LS-SVM for Regression. 2005 International Conference on Neural Networks and Brain. <https://doi.org/10.1109/icnnb.2005.1614615>
- [19] Stanford AIMI Shared Datasets. (2021, 9 septembre). Stanford AIMI. Consulté le 1 février 2022, à l'adresse <https://stanfordaimi.azurewebsites.net/datasets/4aaeafb9-c6e6-4e3c-9188-3aaaf0e0a9e7>
- [20] MeshCNN Github - Data pre-processing. Consulté le 20 mars 2022, à l'adresse <https://github.com/ranahanocka/MeshCNN/wiki/Data-Processing>
- [21] Keldenich, T. (2021, 11 septembre). Recall, Precision, F1 Score – Explication Simple Métrique en ML. Inside Machine Learning. https://inside-machinelearning.com/recall-precision-f1-score/F1_Score
- [22] Kenta-Tanaka et al. (2019, 29 Septembre). Probreg. <https://probreg.readthedocs.io/en/latest/>
<https://github.com/neka-nat/probreg>
- [23] Andriy Myronenko, Xubo Song. (2009, 15 Mai). Point-Set Registration : Coherent Point Drift. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 32, issue 12, pp. 2262-2275

IMT Atlantique Bretagne-Pays de la Loire – <http://www.imt-atlantique.fr/>

Campus de Brest

Technopôle Brest-Iroise

CS 83818

29238 Brest Cedex 3

France

T +33 (0)2 29 00 11 11

F +33 (0)2 29 00 10 00



IMT Atlantique

Bretagne-Pays de la Loire
École Mines-Télécom

Rédigé à IMT Atlantique
Mars2022