



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE  
LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ ABDELHAMID IBN BADIS - MOSTAGANEM

**Faculté des Sciences Exactes et de l'Informatique**  
**Département de Mathématiques et d'Informatique**  
**Filière : Informatique**

MEMOIRE DE FIN D'ETUDES  
Pour l'Obtention du Diplôme de Master en Informatique  
Option : **Ingénierie des Systèmes d'Information**

THEME :

**Analyse d'images médicales, application à la  
catégorisation des types de tuberculose pulmonaire**

Etudiant(e) : « **YAGOUB Djamel Eddine** »

Encadrant(e) : « **Dr. HAMADI Abdelkader** »

Année Universitaire 2017-2018

## Dédicaces

Je dédie ce mémoire

À mes chers parents ma mère et mon père

À ma sœur

À toute ma famille

À mes amis

A tous ceux qui, par un mot, m'ont donné la force de  
continuer...

## Remerciements

Je souhaite manifester mes sincères remerciements à Dieu, le tout puissant, pour ses faveurs et ses grâces de m'avoir donné le courage et la patience pour achever ce modeste travail.

Je tiens à présenter de tout mon cœur mes remerciements et mes reconnaissances à mon honorable encadreur Mr « HAMADI Abdelkader » pour son aide, ses conseils précieux, sa gentillesse, son encouragement, sa disponibilité et sa confiance qui m'a permis de ne jamais faiblir et de poursuivre toujours plus loin dans mon travail.

Je remercie naturellement ma famille pour son aide, sa générosité et son soutien moral qui ont été pour moi une source de courage et de confiance.

Je remercie aussi vivement les honorables membres du jury qui ont accepté d'évaluer ce travail. Enfin, un grand merci à tous ceux qui, par un mot, m'ont donné la force de continuer à travailler afin d'atteindre mes objectifs.

## Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Taux estimé de tuberculose – 2014	3
Figure 2	La progression de la tuberculose	4
Figure 3	Exemples de types de TB. Première rangée de gauche à droite: Infiltrative, Focale et Tuberculome. Deuxième rangée: Miliary et les types Fibro-caverneux	5
Figure 4	Image médicale - représentation 3d	7
Figure 5	Exemple d'arbre de décision	12
Figure 6	Exemple - Random Forest	13
Figure 7	Principe du Séparateur à Vaste Marge (SVM)	14
Figure 8	Schéma d'un neurone à gauche et représentation d'un neurone formel à droit.	16
Figure 9	Architecture d'un réseau de neurones convolutifs	17
Figure 10	Illustration des méthodes de convolution et de Pooling. A. Méthode de convolution B. Méthode de Pooling	18
Figure 11	L'architecture AlexNet	19
Figure 12	L'architecture VGGNet	20
Figure 13	Le module Inception	20
Figure 14	L'architecture GoogleNet	20
Figure 15	l'architecture ResNet-50	21
Figure 16	Phases de construction d'un descripteur sémantique	22
Figure 17	Le module répétitif dans un RNN standards	23
Figure 18	Le module de répétition dans un LSTM	23
Figure 19	Matrice de confusion	27
Figure 20	La méthodologie de l'équipe SGeast	29
Figure 21	Résultats obtenus par les participants à la Tâche de tuberculose (Imageclef 2017).	32
Figure 22	Matrices de confusion de la meilleure exécution par groupe et la fusion des meilleurs 4 groupes.	33
Figure 23	Prétraitement des données d'entrée.	36

Figure 24	Le processus d'extraction des descripteurs sémantiques	37
Figure 25	Entraîner un modèle de classification basé sur les descripteurs sémantiques	38
Figure 26	Réorganisation hiérarchique des types TBT.	40
Figure 27	Résultats et classement en termes de coefficient Kappa sur l'ensemble de test pour la tâche TBT	41
Figure 28	Résultats et classement en termes de précision sur l'ensemble de test pour la tâche TBT.	42
Figure 29	La hiérarchie des classes considérées pour la tâche SVR	44
Figure 30	Résultats et classement en termes de RMSE sur l'ensemble de test	45
Figure 31	Résultats et classement en termes d'AUC sur l'ensemble de test.	46
Figure 32	Interface de StarUML version 5.0	47
Figure 33	Diagramme de cas d'utilisation	48
Figure 34	Diagramme de séquence de la conversion (Nifti en JPG)	49
Figure 35	Diagramme de séquence de l'extraction des descripteurs (Features Extraction)	50
Figure 36	Diagramme de séquence de l'apprentissage (Training)	51
Figure 37	Diagramme de séquence de la prédiction (Predection)	52
Figure 38	Interface graphique de la conversion des fichiers Nifti en format JPG	54
Figure 39	Interface graphique d'extraction des descripteurs.	54
Figure 40	Interface graphique d'extraction de l'apprentissage.	55
Figure 41	Interface graphique pour la prédiction.	55

## Liste des tableaux

Tableau N°	Titre du tableau	Page
Tableau 1	Résumé des caractéristiques des formats de fichiers d'images médicales.	8
Tableau 2	Résultats de certaines applications de Deep Learning dans l'analyse d'images médicales	24
Tableau 3	Les données pour la tâche TBT dans ImageClef 2017	27
Tableau 4	Données fournies pour la tâche TBT Tuberculose	39
Tableau 5	Résultats sur l'ensemble de validation pour la tâche TBT	41
Tableau 6	Résultats sur l'ensemble de test pour la tâche TBT	41
Tableau 7	Données fournies pour la tâche SVR Tuberculose.	43
Tableau 8	Résultat sur l'ensemble de validation pour la tâche SVR en termes de précision moyenne et RMSE	44
Tableau 9	Résultats sur l'ensemble de test pour la tâche SVR	45

## Liste des abréviations

Abréviation	Expression Complète	Pages
CNN	Convolutional Neural Network.	17, 18, 19, 24, 28, 29, 30 ,31
RNN	Recurrent neural network	22, 23
LSTM	Long short-term memory	23, 28, 30
TB	Tuberculosis	1, 3, 5, 6, 30, 32, 33, 39, 43
SVM	Support vector machine	11, 14
SVR	Severity scoring	35, 37, 39, 43, 44, 45, 57
TBT	Tuberculosis Type	27, 35, 37, 38, 39, 40, 41, 42, 57

## Table des Matières

Introduction générale .....	1
Chapitre 01 : Etat de l'art .....	3
1. Introduction partielle .....	3
2. Tuberculose .....	3
2.1. Tuberculose pulmonaire .....	4
2.2. Types de tuberculose pulmonaire .....	5
2.2.1. La tuberculose Infiltrative.....	5
2.2.2. Focale.....	6
2.2.3. Tuberculome .....	6
2.2.4. Miliaire .....	6
2.2.5. Fibro-cavernous .....	6
3. Imagerie médicale .....	7
3.1. Formats d'images médicales .....	7
3.1.1. Analyze .....	8
3.1.2. Nifti.....	8
3.1.3. Minc .....	9
3.1.4. Dicom .....	9
4. Apprentissage automatique .....	10
4.1. Apprentissage supervisé .....	11
4.1.1. Arbre de décision .....	11
4.1.2. Bagging.....	12
4.1.3. Forêts aléatoires (Random forest).....	13
4.1.4. Machine à Vecteurs de Support.....	14
4.2. Classification .....	15
4.2.1. Classification Hiérarchique .....	15
4.2.2. Classification Multi-classes .....	15
4.3. Apprentissage profond.....	16
4.3.1. Réseaux de neurones .....	16
4.3.2. Réseaux de neurones convolutifs .....	17
4.3.3. Réseaux de neurones convolutifs pour la classification d'images .....	18
4.3.4. Architectures de réseaux profonds .....	19
4.3.4.1. AlexNet.....	19
4.3.4.2. VGG Net .....	19
4.3.4.3. GoogleNet.....	20

4.3.4.4. ResNet.....	21
4.3.6. Descripteurs sémantiques .....	22
4.3.7. Réseau neuronal récurrent .....	22
4.3.8. Réseaux de mémoire à longue durée .....	23
4.3.9. Apprentissage profond pour l'analyse d'images médicales.....	23
4.3.10. NiftyNet: une plate-forme d'apprentissage profond pour l'imagerie médicale.....	24
5. Conclusion partielle.....	25
Chapitre 2 : Classification des types de tuberculose .....	26
1. Introduction partielle .....	26
2. ImageCLEF et la tâche tuberculoses .....	26
3. Organisation de la tâche .....	26
4. Données .....	26
5. Mesures d'évaluation .....	27
5.1. Matrice de confusion .....	27
5.2. Précision moyenne (Accuracy).....	27
5.3. Kappa de Cohen .....	28
6. Quelques travaux de classification de types de tuberculose pulmonaire .....	28
6.1. Travail de SGEast.....	28
6.1.1. Préparation des données .....	29
6.1.2. Apprentissage par transfert (Transfer Learning) .....	29
6.1.3. Apprentissage sur séquence (Séquence Learning) .....	30
6.2. Travail de MEDGIFT UPB .....	31
6.2.1. Préparation des données .....	31
6.2.2. Phase d'entraînement (Train).....	31
7. Résultats de la classification de types de tuberculose .....	32
8. Discussion .....	33
9. Conclusion partielle.....	34
Chapitre 03 : Contribution .....	35
1. Introduction partielle .....	35
2. Description de la tâche SVR .....	35
3. Contribution35	
3.1. Prétraitement des données d'entrée36	

3.2. Extraction des descripteurs sémantique .....	37
3.3. Apprendre un modèle de classification .....	38
4. Expérimentations et résultats .....	39
4.1. Tâche TBT .....	39
4.1.1. Données utilisées.....	39
4.1.2. Protocole expérimental.....	40
4.1.3. Résultats .....	41
4.2. Tâche SVR .....	43
4.2.1. Données utilisées.....	43
4.2.2. Protocole expérimental.....	43
4.2.3. Résultats .....	44
5. Conclusion partielle .....	46
Chapitre 04 : Conception et Implémentation.....	47
1. Introduction partielle.....	47
2. Conception .....	47
2.1. Diagramme de cas d'utilisation4.....	48
2.2. Diagramme d'activité.....	49
3. Implémentation .....	53
3.1. Environnement de travail .....	53
3.1.1. Environnement matériel .....	53
3.1.2. Environnement logiciel .....	53
3.1.3. Langages et outils de développement.....	53
3.2. Les interfaces graphiques principales .....	53
4. Conclusion partielle .....	56
Conclusion générale.....	57
Bibliographie .....	58
Webographie.....	60
Annexe.....	60

# Introduction générale

Au cours des dernières décennies, l'imagerie médicale est devenue indispensable dans le diagnostic et la thérapie des maladies. Avec l'enrichissement des bases d'images médicales, de nouvelles méthodes sont requises pour mieux traiter cet énorme volume de données. Cependant, en raison des grandes variations et de la complexité des données d'imagerie médicale, il est généralement difficile de déduire des solutions analytiques ou des méthodes simples pour décrire et représenter des objets tels que des lésions et des anatomies dans des données. Par conséquent, les tâches en imagerie médicale nécessitent *un apprentissage à partir d'exemples*, et cela constitue un des intérêts clés du domaine de l'apprentissage machine.

L'apprentissage machine (automatique) est devenu l'un des principaux outils pour l'analyse d'images médicales. Les techniques d'apprentissage automatique constituent des solutions pour développer des outils permettant d'aider les médecins à diagnostiquer, à prédire le risque d'atteinte de maladies et à les prévenir avant qu'il ne devienne trop tard. L'apprentissage profond (En anglais, Deep Learning) est un nouveau volet du domaine de l'apprentissage automatique qui englobe un large éventail d'architectures de réseaux élaborés pour exécuter plusieurs tâches. La première utilisation de réseaux de neurones pour l'analyse d'images médicales remonte à plus de vingt ans [1], leur utilisation a augmenté de plusieurs ordres de grandeur au cours des cinq dernières années. Des revues récentes [2] ont mis en évidence l'application de l'apprentissage profond à un large éventail de tâches d'analyse d'images médicales (segmentation, classification, détection, enregistrement, reconstruction d'image, amélioration, etc.).

La tuberculose est une maladie infectieuse causée par une bactérie appelée *Bacillus mycobacterium-tuberculosis*. Avec une estimation de 10,4 millions de nouveaux cas tuberculeux et un taux de mortalité de 1,8 million dans le monde, cette maladie est restée l'une des dix premières causes de mortalité dans le monde en 2015. La tuberculose attaque les poumons, mais peut également affecter d'autres parties du corps. Un diagnostic précis et rapide est la clé du contrôle de cette maladie, mais les tests traditionnels de TB produisent des résultats inexacts ou trop longs pour être définitifs. Des chercheurs se sont intéressés à cette maladie, notamment dans le cadre du défi international *ImageCLEF 2017* [3] où deux tâches lui ont été réservées. Des algorithmes impliquant l'apprentissage profond ont été testés pour diagnostiquer la présence ou l'absence de tuberculose. Les résultats obtenus ont été intéressants. En effet, les algorithmes ont atteint un taux de précision impressionnant de 96% [4] un résultat qui s'avère meilleur que celui de l'intervention de nombreux radiologistes.

L'objectif de notre projet consiste à détecter automatiquement les types de TB via une analyse d'images par tomographie assistée par ordinateur (CT). Une des applications possibles de cette étude est d'accélérer le diagnostic de la maladie à partir d'une image de radiologie sans avoir recours à des analyses médicales coûteuses. Ce travail s'inscrit dans le cadre de la tâche de classification

des types de tuberculose du défi *ImageCLEF* 2017 [3] qui a montré des résultats très prometteurs. Nous explorons dans ce mémoire les différents travaux effectués et les différentes notions ayant un lien avec cette problématique.

Nous proposons dans ce travail de développer un système de classification de types de tuberculose en exploitant des descripteurs sémantiques des images de Scanner des patients.

Nous pouvons résumer les objectifs de notre système à travers les points suivants:

- Aider les médecins dans le diagnostic de la tuberculose par des techniques de traitement d'image et d'apprentissage automatique
- Proposer un outil exploitant des méthodes peu coûteuses (en termes d'argent) et rapides pour une détection précoce des types de tuberculose chez les patients
- Prévoir rapidement le type de tuberculose et son degré de gravité pour aider les médecins à prendre des décisions rapides et à donner les traitements efficaces.

Le travail décrit dans ce mémoire a fait l'objet d'une participation à deux sous-tâches de la tâche de *Tuberculosis* d'*ImageCLEF* 2018 [32] : Classification des types de tuberculose et Prédiction du degré de sévérité du cas tuberculeux. Ce même travail a abouti à une soumission d'un article de conférence.

Le reste du rapport est organisé en quatre chapitres suivis d'une conclusion générale, chacun représente une partie du travail de ce projet qu'elle soit théorique ou pratique :

- **Chapitre 1** : Etat de l'art
- **Chapitre 2** : Classification types tuberculose
- **Chapitre 3** : Contributions
- **Chapitre 4** : Conception et réalisation

# **Chapitre 01 :**

# **Etat de l'art**

## 1. Introduction partielle

Nous présentons dans ce chapitre divers concepts sur la maladie de la Tuberculose ainsi que l'apprentissage machine ayant un lien avec notre problématique de recherche.

## 2. Tuberculose

La tuberculose (TB) est une maladie infectieuse chronique provoquée par une bactérie dénommée « *Mycobacterium Tuberculosis* » ou Bacille de Koch (BK). Sa forme la plus courante et la plus répandue (85% des cas) est la tuberculose pulmonaire mais il existe également des formes extra-pulmonaires comme la tuberculose osseuse, la tuberculose ganglionnaire et la tuberculose rénale.

La tuberculose peut se développer rapidement après le 1<sup>er</sup> contact avec le microbe mais elle peut aussi apparaître plusieurs années plus tard.

On peut distinguer deux cas d'atteinte de la tuberculose:

- **Infection de tuberculose latente (ITL) :**

L'ITL est la présence de bacilles tuberculeux à l'intérieure de l'organisme sans manifestation de la maladie. Les porteurs de l'ITL sont par définition non contagieux et ne représente aucun risque pour leur entourage.

- **Tuberculose active :**

La tuberculose active est un état où les microbes sont actifs et entraînent des symptômes. Cela indique que le système immunitaire d'une personne infectée par le microbe n'a pu se défendre convenablement contre ce dernier. Cette forme de tuberculose est contagieuse.

Les taux estimés d'incidence de la tuberculose sont présentés à la figure 1.

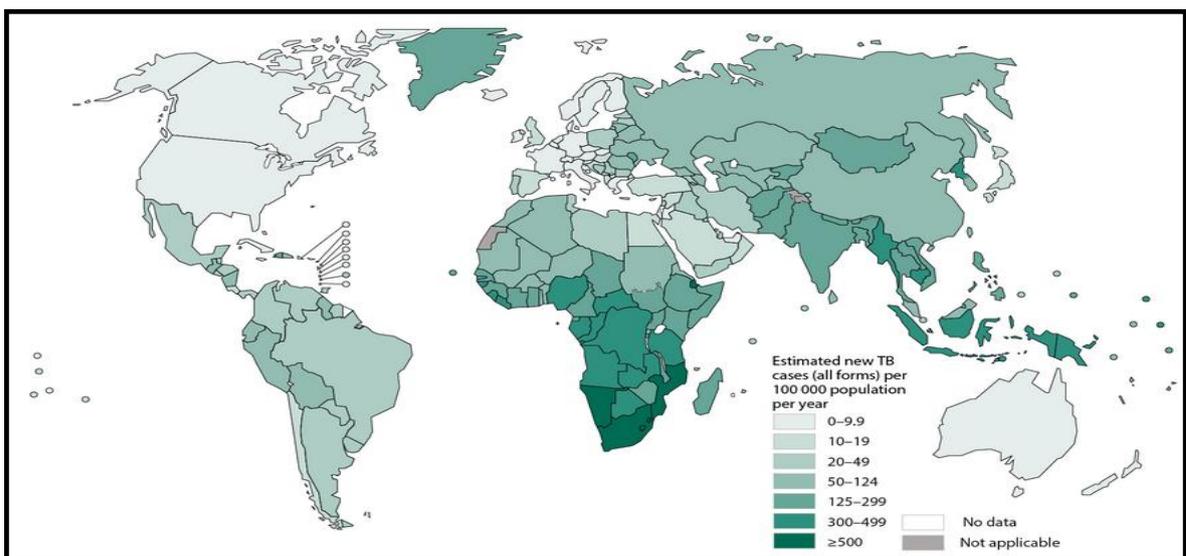


Figure 1 - Taux estimé de tuberculose – 2014 [19].

## 2.1. Tuberculose pulmonaire

La tuberculose pulmonaire est une maladie des poumons très commune. C'est la forme de tuberculose la plus contagieuse en raison de sa localisation et du nombre élevé de bacilles produits par le malade.

La bactérie responsable de la tuberculose pulmonaire est transmise par voie aérienne, via les gouttelettes qui sont suspendues dans l'air expiré par les malades, en particulier lors de la toux. Une fois qu'il a pénétré dans l'organisme, le bacille de Koch va se disséminer dans les voies respiratoires.

La figure 2 montre les étapes de l'évolution de la tuberculose dans les poumons.

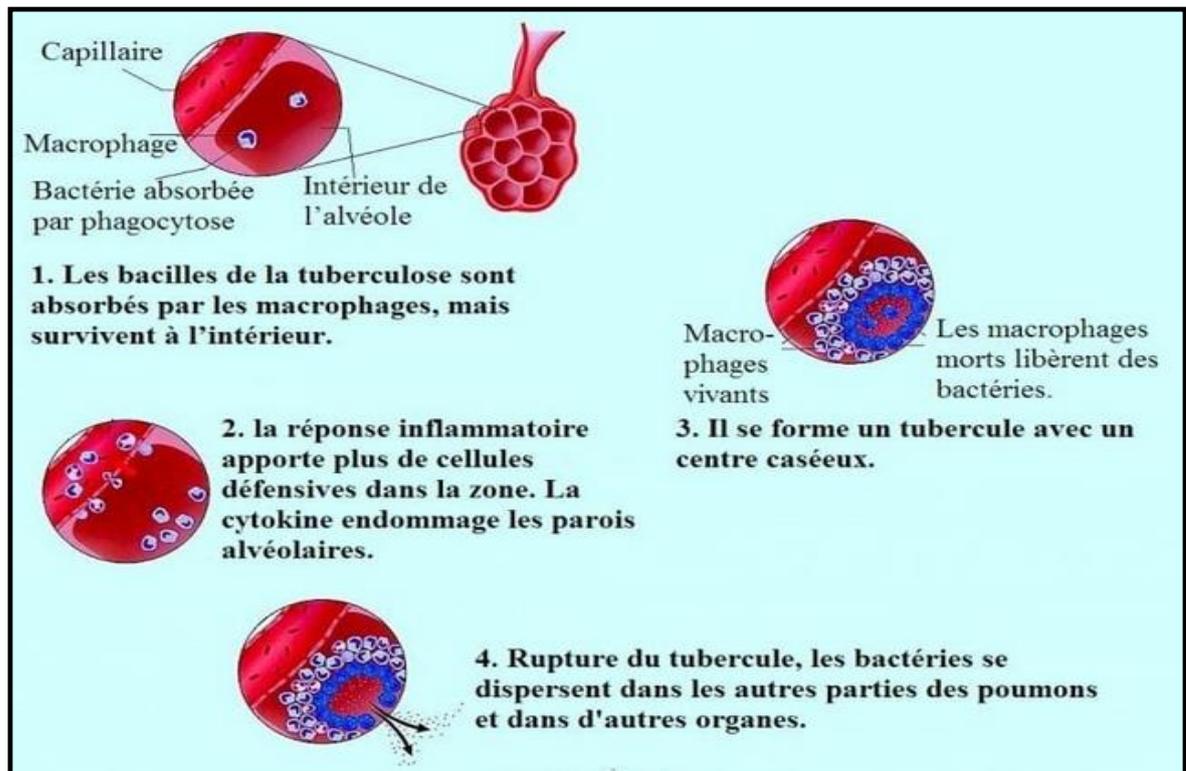


Figure 2 - La progression de la tuberculose [20].

## 2.2. Types de tuberculose pulmonaire

On distingue cinq types de tuberculose pulmonaire : Infiltrative, Focale, Tuberculome, Miliaire et Fibro-Cavernous. Une tranche 2D par type de tuberculose est représentée dans la figure 3.

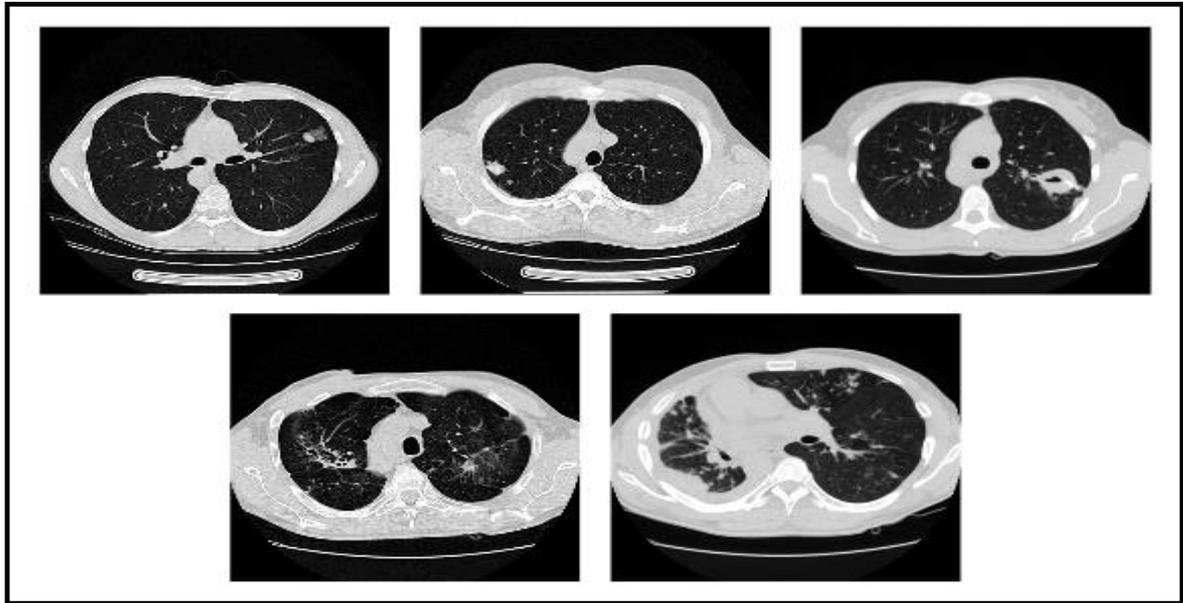


Figure 3 - Exemples de types de TB. Première rangée, de gauche à droite: Infiltrative, Focale, et Tuberculome. Deuxième rangée: Miliary, et les types Fibro-caverneux [3]

### 2.2.1. La tuberculose Infiltrative

La tuberculose Infiltrative est la forme la plus fréquemment diagnostiquée de la maladie. Les personnes atteintes de tuberculose Infiltrative sont principalement des adultes. La mortalité due à cette forme de la maladie est d'environ 1% de la tuberculose pulmonaire totale. Il existe plusieurs formes de tuberculose Infiltrative : lobulaire, ronde, nuage et Infiltration régionale.

La forme de nuage de la tuberculose Infiltrative est caractérisée par les contours flous des ombres sur la radiographie. Cela se produit souvent lors de la formation de nouvelles cavités. L'infiltration ronde apparaît sur la radiographie comme arrondie, ou ombre de structure homogène. Tous les taches sont divisées en petites, moyennes et grandes. Les petites ont une grandeur de 1 à 2 cm, les moyens 2-4 cm, les grands jusqu'à 6 cm [5].

La Diagnostic comprend: la mesure de la température et de la pression du corps, une analyse de sang générale et biochimique, un test de *Mantoux* et *Diaskintest*, un examen radiologique des poumons et la bronchoscopie.

### 2.2.2. Focale

La TB focale est diagnostiqué en cas de lésion ne dépassant pas deux segments pulmonaires. En pathologie, il se forme un petit nombre de foyers pathologiques d'un diamètre d'environ 1 cm, qui se produisent à des moments différents [6].

La Diagnostic comprend: un test tuberculitique (Test de *Mantoux*), un test sanguin biochimique et la *Radiographie*.

### 2.2.3. Tuberculome

Un tuberculome ou granulome tuberculeux est une manifestation clinique de la tuberculose qui conglomère les tubercules en une masse ferme et peut ainsi imiter les tumeurs cancéreuses.

La taille du tuberculome est généralement inférieure à 3 cm mais peut atteindre 5 cm; il siège préférentiellement dans les lobes supérieurs et sont multiples dans 20% des cas [7].

La Diagnostic comprend: un examen cyto bactériologique et la biopsie pulmonaire.

### 2.2.4. Miliaire

La tuberculose miliaire est ainsi appelée en raison d'innombrables petites taches qui se forment dans les poumons de la taille des grains de mil. La TB miliaire peut affecter un ou plusieurs organes ou se propager dans tout l'organisme [8].

La tuberculose miliaire se développe le plus souvent chez les personnes suivantes :

- Les enfants âgés de moins de 4 ans;
- Les personnes dont le système immunitaire est affaibli;
- Personnes âgées.

La Diagnostic comprend: un test cutané tuberculitique, des tests sanguins pour détecter la bactérie responsable de la tuberculose et une radiographie du thorax.

### 2.2.5. Fibro-cavernous

La tuberculose pulmonaire Fibro-Cavernous est une maladie chronique dans laquelle des cavités s'y forment. Cette formation est un résultat de la nécrose tissulaire. Parmi tous les patients tuberculeux, environ 4% souffrent de cette forme [9].

La Diagnostic comprend: Radiographie, la Bronchoscopie et un test sanguin.

### 3. Imagerie médicale

L'imagerie médicale est certainement l'un des domaines de la médecine qui a le plus progressé ces vingt dernières années. Ces récentes découvertes permettent non seulement un meilleur diagnostic mais offrent aussi de nouveaux espoirs de traitement pour de nombreuses maladies comme le cancer, épilepsie, tuberculose, etc. L'identification précise de la lésion facilite déjà le recours à la chirurgie. De telles techniques permettent également de mieux comprendre le fonctionnement de certains organes encore mystérieux, comme le cerveau.

L'image médicale est une représentation discrète d'une partie du corps qui est décrite par une matrice à 3 dimensions, comme la montre la figure 4:

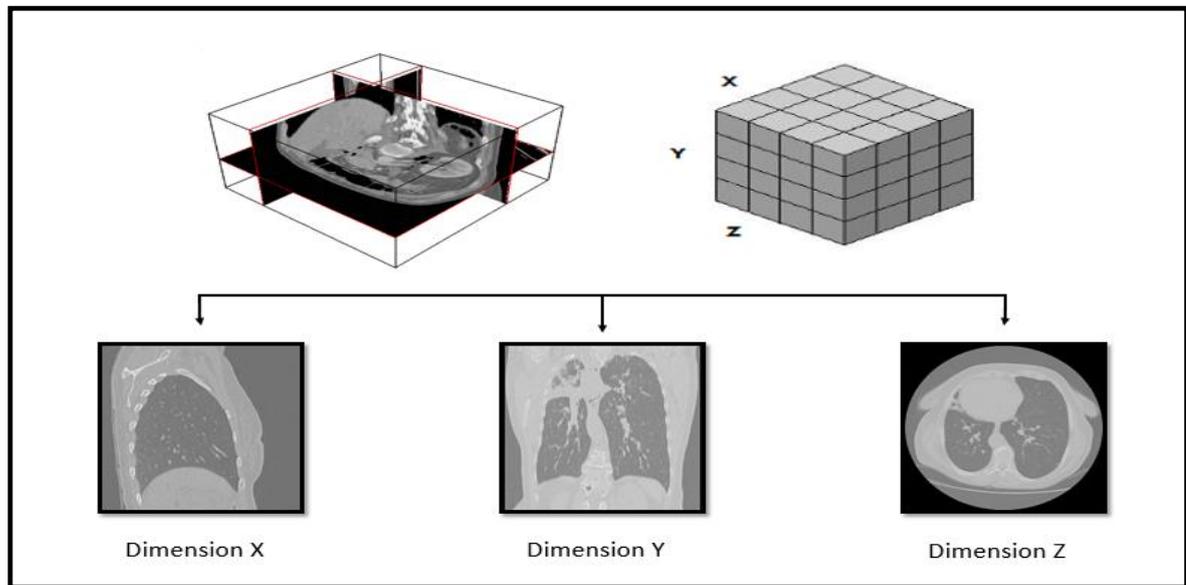


Figure 4 - Image médicale - représentation 3d

#### 3.1. Formats d'images médicales

Les formats d'images médicales peuvent être divisés en deux catégories. Le premier est celui des formats destinés à standardiser les images générées par des modalités de diagnostic, comme le format "Dicom" par exemple. Le second est celui des formats nés dans le but de faciliter et de renforcer l'analyse post-traitement, comme par exemple: Analyze, Nifti et Minc. Les fichiers d'images médicales sont généralement stockés en utilisant l'une des deux configurations suivantes. Un dans lequel un seul fichier contient à la fois les métadonnées et les données d'image, où les métadonnées sont stockées au début du fichier. Ce paradigme est utilisé par les formats de fichiers Dicom, Minc et Nifti, même si cela est autorisé par d'autres formats. La deuxième configuration stocke les métadonnées dans un fichier et les données d'image dans un second. Le format de fichier Analyze utilise le paradigme à deux fichiers (.hdr et .img) [14].

Tableau 1 - Résumé des caractéristiques des formats de fichiers d'images médicales.

Format	Header	Extension	Data types
Analyze	Longueur fixe: format binaire de 348 octets	.img et .hdr	Entier non signé (8 bits), entier signé (16, 32 bits), flottant (32, 64 bits), complexe (64 bits)
Nifti	Longueur fixe: format binaire 352 octets (348 octets dans le cas de données stockées sous .img et .hdr)	.nii	Entier signé et non signé (de 8 à 64 bits), flottant (de 32 à 128 bits), complexe (de 64 à 256 bits)
Minc	Format binaire extensible	.mnc	Entier signé et non signé (de 8 à 32 bits), flottant (32, 64 bits), complexe (32, 64 bits)
Dicom	Format binaire de longueur variable	.dcm	Entier signé et non signé, (8, 16 bits, 32 bits autorisés uniquement pour la dose de radiothérapie), non pris en charge

### 3.1.1. Analyze

*Analyze 7.5* a été créée à la fin des années 1980 sous forme de format utilisé par le logiciel commercial *Analyze* développé à la Mayo Clinic à Rochester, MN, USA. Pendant plus d'une décennie, le format était la norme pour le post-traitement de l'imagerie médicale. Le gros point de vue du format *Analyze* est qu'il a été conçu pour des données multidimensionnelles (volume). En effet, il est possible de stocker dans un fichier des données 3D ou 4D (la quatrième dimension étant typiquement l'information temporelle). Un volume *Analyze 7.5* comprend deux fichiers binaires: un fichier image avec l'extension ".img" qui contient les données brutes de voxel et un fichier d'en-tête avec l'extension ".hdr" qui contient les métadonnées (nombre de pixels dans les trois dimensions, la taille du voxel et le type de données). L'en-tête a une taille fixe de 348 octets et est décrit comme une structure dans le langage de programmation C. La lecture et l'édition de l'en-tête nécessitent un utilitaire. Le format est aujourd'hui considéré comme «ancien» mais il est encore largement utilisé et pris en charge par de nombreux logiciels de traitement, visionneuses et utilitaires de conversion [14]

### 3.1.2. Nifti

Nifti est un format de fichier créé au début des années 2000 avec l'intention de créer un format pour la neuro-imagerie en conservant les avantages du format *Analyze*, mais en résolvant ses faiblesses. Le Nifti peut en fait être considéré comme un format "Analyze" révisé. Le format remplit certains des champs inutilisés / peu utilisés présents dans l'en-tête *Analyze 7.5* pour stocker de nouvelles informations comme l'orientation de l'image avec l'intention d'éviter l'ambiguïté gauche-droite dans l'étude du cerveau. De plus, Nifti inclut la prise en charge du type de données non envisagé dans le format *Analyze* comme le format 16 bits non signé. Bien

que le format permette également le stockage des données d'en-tête et de pixel dans des fichiers séparés, les images sont généralement enregistrées sous la forme d'un fichier ".nii" unique dans lequel l'en-tête et les données de pixel sont fusionnées. L'en-tête a une taille de 348 octets dans le cas de stockage de données ".hdr" et ".img", et une taille de 352 octets dans le cas d'un seul fichier ".nii". Cette différence de taille est due à la présence de quatre octets supplémentaires à la fin, essentiellement pour faire de la taille un multiple de 16, et aussi pour fournir un moyen de stocker des métadonnées supplémentaires, auquel cas ces 4 octets sont différents de zéro [14].

### **3.1.3. Minc**

Le format de fichier Minc a été développé en 1992 pour fournir un format de données flexible pour l'imagerie médicale. La première version du format Minc (Minc1) était basée sur le format standard de réseau commun (NetCDF). Par la suite, pour surmonter la limite de prise en charge de fichiers de données volumineux et pour fournir d'autres nouvelles fonctionnalités, l'équipe de développement de Minc a choisi de passer de NetCDF à Hierarchical Data Format version 5 (HDF5). Cette nouvelle version, non compatible avec la précédente, s'appelait Minc2 [14].

### **3.1.4. Dicom**

DICOM (Digital Imaging and Communications in Medicine), est la norme internationale pour transmettre, stocker, récupérer, imprimer, traiter et afficher des informations d'imagerie médicale.

Dicom ne peut stocker que des valeurs de pixels en tant qu'entier. Dicom ne peut actuellement pas enregistrer les données de pixel en virgule flottante alors qu'il prend en charge divers types de données, y compris les flottants, pour stocker les métadonnées. Chaque fois que les valeurs stockées dans chaque voxel doivent être mises à l'échelle, Dicom utilise un facteur d'échelle utilisant deux champs dans l'en-tête définissant la pente et l'ordonnée à l'origine de la transformation linéaire à convertir en valeurs réelles. Dicom prend en charge les données d'image compressées via un mécanisme qui permet d'encapsuler un document non-Dicom dans un fichier Dicom. Les systèmes de compression pris en charge par Dicom sont les suivants: JPEG, RLE (Run-Length Encoding), JPEG-LS, JPEG-2000 et MPEG2 / MPEG4 [14].

## 4. Apprentissage automatique

Le « **machine Learning** », ou littéralement l'apprentissage automatique, fait partie de l'une des approches de l'intelligence artificielle. C'est une discipline consacrée à l'analyse des données. Le but de cette discipline est de créer de la connaissance de manière automatique à partir de données brutes (échantillons). Cette connaissance (ou bien modèle) peut alors être exploitée pour prendre des décisions [26].

En fonction de la nature des échantillons (labellisés ou pas), il existe trois types d'apprentissage:

- **Apprentissage supervisé.** Cette approche a pour objectif la conception d'un modèle reliant des données d'apprentissage à un ensemble de valeurs de sortie (Les données d'entrée sont labellisées).
- **Apprentissage non supervisé.** [27] : Il vise à concevoir un modèle structurant l'information. La différence ici est que les comportements (ou catégories ou encore les classes) des données d'apprentissage ne sont pas connus, c'est ce que l'on cherche à trouver.
- **Apprentissage semi-supervisé.** [28] : Les données d'entrée sont constituées d'exemples étiquetés et non étiquetés. Ce qui peut être très utile quand on a deux types de données, car cela permet de ne pas en laisser de côté et d'utiliser toute l'information.
- **Autres.** Il existe de nombreuses méthodes d'apprentissage (apprentissage par renforcement, transfer learning, self-taught learning)

En fonction de la nature de la sortie (catégorisée ou pas), il existe deux types de modèles :

- **Classifier.** lorsque l'ensemble des valeurs de sortie est discret. Ceci revient à attribuer une classe (aussi appelée étiquette ou label) pour chaque vecteur d'entrée. (on dit que le modèle effectue une **classification**)
- **Régresseur.** lorsque le résultat supervisé que l'on cherche à estimer est une valeur dans un ensemble continu de réels. (on dit que le modèle effectue une **régression**)

## 4.1. Apprentissage supervisé

Nous nous intéressons plus particulièrement à l'apprentissage supervisé, qu'est simplement une formalisation de l'idée d'apprendre à partir d'exemples. Dans l'apprentissage supervisé, l'apprenant (généralement un programme d'ordinateur) reçoit deux ensembles de données, un ensemble d'apprentissage et un ensemble de tests. L'idée est que l'apprenant «apprenne» à partir d'un ensemble d'exemples labellisés dans l'ensemble d'apprentissage afin qu'il puisse identifier les exemples non-labellisés dans l'ensemble de test avec la plus grande précision possible. C'est-à-dire que le but de l'apprenant est de développer une règle, un programme ou une procédure qui classifie de nouveaux exemples (dans l'ensemble de test) en analysant des exemples qui lui ont déjà été attribués.

En apprentissage supervisé, l'ensemble d'apprentissage est constitué de  $n$  paires ordonnées  $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$ , où chaque  $x_i$  est une mesure ou un ensemble de mesures d'un seul point de données, et  $y_i$  est l'étiquette pour ce point de données.

Les données de test en apprentissage supervisé sont un autre ensemble de  $m$  mesures sans étiquettes:  $(X_{n+1}, X_{n+2}, \dots, X_{n+m})$ .

Il existe plusieurs algorithmes d'apprentissage supervisé, on présente quelques-uns des plus connus parmi eux, il s'agit de machines à vecteurs de support (SVM), les arbres de décision et les random forests :

### 4.1.1. Arbre de décision

Les arbres de décision sont un outil très populaire de classification. Leur principe repose sur la construction d'un arbre de taille limitée. La racine constitue le point de départ de l'arbre et représente l'ensemble des données d'apprentissage. Puis ces données sont segmentées en plusieurs sous-groupes, en fonction d'une variable discriminante (un des attributs).

Une fois l'arbre construit à partir des données d'apprentissage, on peut prédire un nouveau cas en le faisant descendre le long de l'arbre, jusqu'à une feuille. Comme la feuille correspond à une classe, l'exemple sera prédit comme faisant partie de cette classe.

Exemple, sur la Figure 5, la première variable discriminante est l'outlook. Elle divise la population en trois sous-groupes : Sunny, Overcast et Rain. Le processus est ensuite réitéré au deuxième niveau de l'arbre, où les sous-populations sont segmentées à leur tour en fonction d'une autre valeur discriminante.

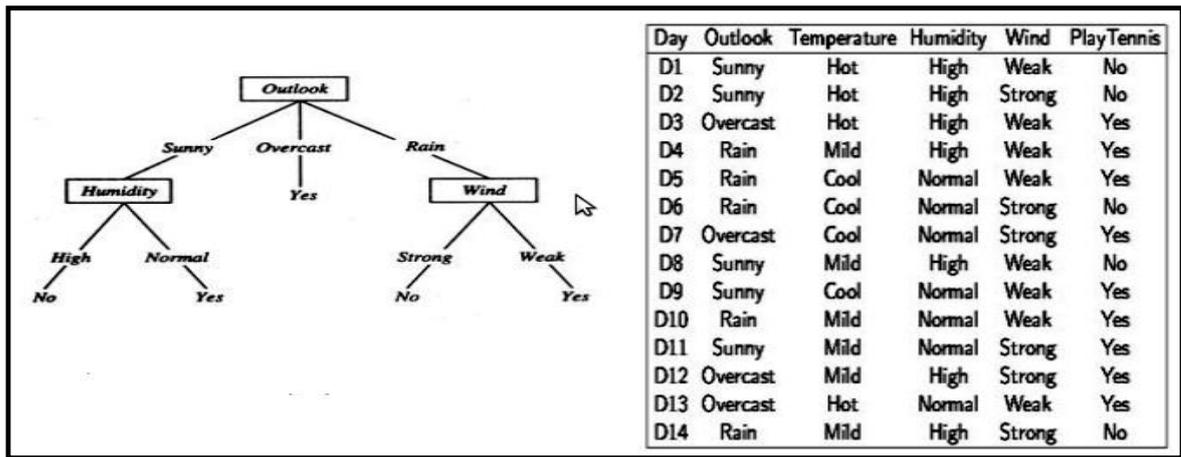


Figure 5 - Exemple d'arbre de décision

Pour construire un arbre de décision, nous utilisons:

- Les critères de choix l'entropie et l'indice de Gini, pour choisir le meilleur couple (attribut; test) pour créer un nœud :

-Entropie =  $-\sum_{i=1}^k (P(W_i) \cdot \log_2(P(W_i)))$ , avec  $P(W_k)$  la probabilité de la class  $W_i$

-L'indice de Gini =  $1 - \sum_{i=1}^k P^2(W_i)$

- Le gain d'information est basé sur la diminution de l'entropie après la division d'un ensemble de données sur un attribut. Construire un arbre de décision est une question de trouver un attribut qui renvoie le gain d'information le plus élevé :

-Gain (N, T) =  $I(N) - \sum_{i=1}^m P(N_i) \cdot I(N_i)$ , ou T: le test,  $N_i$  : le nœud N en m sous-nœud  $N_i$ ,  $I(N_i)$  : Critère de choix.

#### 4.1.2. Bagging

Il consiste à sous-échantillonner (ou ré-échantillonner au hasard avec doublons) le training set et de faire générer à l'algorithme voulu un modèle pour chaque sous-échantillon. On obtient ainsi un ensemble de modèles dont il convient de moyenner (lorsqu'il s'agit d'une régression) ou de faire voter (pour une classification) les différentes prédictions.

Cette technique concerne les apprenants dont l'erreur est due en plus grande partie à la variance, c'est-à-dire des apprenants qui réagissent à de petites modifications de l'ensemble d'apprentissage en produisant des hypothèses différentes à chaque fois.

La solution proposée est justement d'utiliser plusieurs fois l'apprenant pour obtenir autant d'hypothèses et les combiner au sein d'un vote. Pour obtenir des hypothèses différentes, le bagging propose de construire de nouveaux échantillons à partir des données initiales.

**Algorithme :**

1- Échantillonnage :

- Entrée : A un ensemble des données d'apprentissage contenant n exemples ;
- Sortie : A' un nouvel ensemble de données de même taille ;
- Tirer avec remise n exemples de A et les placer dans A'.

2- Bagging :

- Entrées : un ensemble d'exemples  $A = \{(X_i, Y_i)\}$ , un nombre d'itérations T à effectuer, un apprenant L ;
- Sortie : H le classifieur final ;
- Pour t allant de 1 à T
  - $A_t = \text{Échantillonnage}(A)$
  - $h_t = L(A_t)$
- Retourner (la décision) :  $H(x) = \text{sign}(\sum_{t=1}^T (h_t(x)))$

**4.1.3. Forêts aléatoires (Random forest)**

La forêt aléatoire ou la forêt de décision aléatoire [29] est une méthode qui fonctionne en construisant plusieurs arbres de décision pendant la phase d'apprentissage. La décision de la majorité des arbres est choisie par la forêt aléatoire comme décision finale.

L'algorithme de la forêt aléatoire utilise une approche de Bagging pour créer un groupe d'arbres de décision avec un sous-ensemble aléatoire de données.

L'algorithme de la forêt aléatoire peut se diviser en deux étapes. Premièrement, dans lequel 'n' arbres aléatoires sont créés, cela forme la forêt aléatoire. Dans la deuxième étape, le résultat pour la même caractéristique de test de tous les arbres de décision est combiné. Ensuite, la prédiction finale est dérivée en évaluant les résultats de chaque arbre de décision ou simplement en faisant une prédiction qui apparaît le plus souvent dans les arbres de décision. (Figure 06)

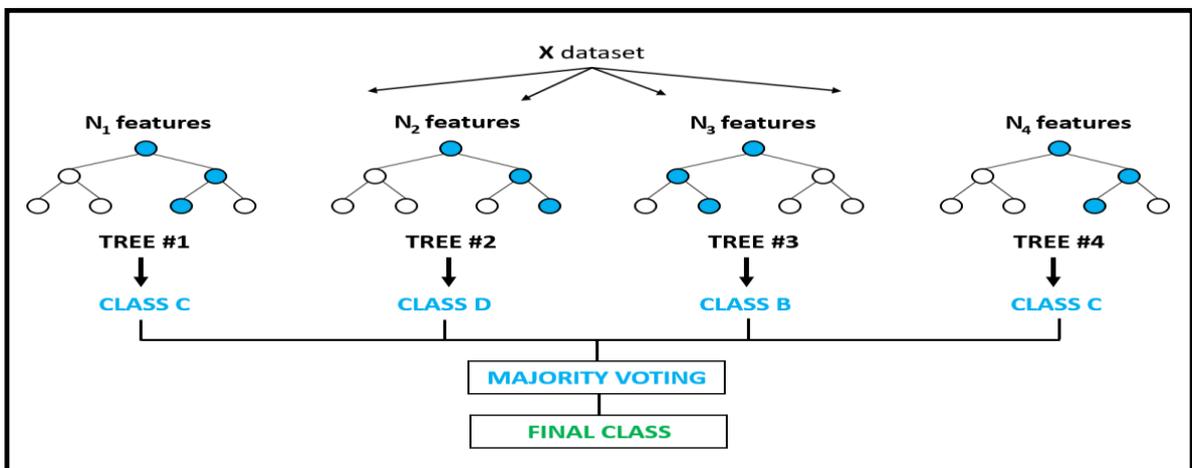


Figure 6 - Exemple - Random Forest [29].

Le pseudo code de l'algorithme Random Forest est défini comme suite :

Entrée : l'ensemble d'apprentissage T, Nombre d'arbres N.

Sortie : Ensemble d'arbres E

Processus :

$E \leftarrow \emptyset$

Pour  $i = 1 \rightarrow N$  faire :

$T_i \leftarrow \text{Bagging}(T)$

$C_i \leftarrow \text{ConstruireArbre}(T_i)$  ou à chaque nœud :

$E \leftarrow E \cup \{C_i\}$

Fin pour

Retourne E

#### 4.1.4. Machine à Vecteurs de Support

Les machines à vecteurs de support ou séparateurs à vastes marges (notées SVM pour Support Vector Machines) ont été introduites lors de la conférence COLT en 1992 [30]. C'est un algorithme d'apprentissage automatique supervisé qui peut être utilisé à la fois pour la classification et la régression.

En effet, l'idée principale des SVM est de reconsidérer le problème dans un espace de dimension supérieure, éventuellement de dimension infinie. Dans ce nouvel espace, il est alors probable qu'il existe un hyperplan séparateur linéaire. Si c'est le cas, les SVM cherchent parmi l'infinité des hyperplans séparateurs celui qui maximise la marge entre les classes.

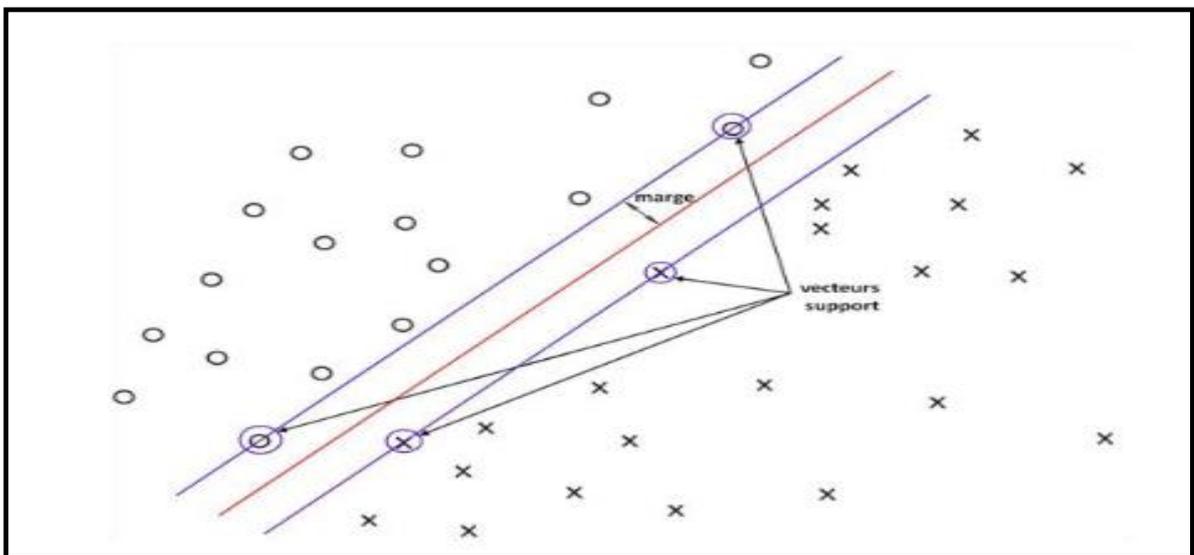


Figure 7 - Principe du Séparateur à Vaste Marge (SVM) [30].

## 4.2. Classification

### 4.2.1. Classification Hiérarchique

La classification hiérarchique est une méthode de classification itérative dont le principe est simple :

- Considérer une hiérarchie de classes;
- Classer les échantillons dans les classes du niveau ancêtre (le plus haut)  
Descendre dans la hiérarchie pour classer les échantillons dans les classes du niveau courant. Le niveau suivant sera celui atteint en passant par la classe prédite dans le niveau actuel.
- Itérer jusqu'à arriver au niveau le plus bas de la hiérarchie.
- Il existe toutefois, des méthodes qui effectue la classification dans le sens inverse, en commençant par les classes du niveau le plus bas puis continuent le traitement en allant vers les niveaux de haut: Down-Top stratégie.

L'ensemble des classifications successives réalisées produisent un arbre où chaque nœud d'un niveau correspond à la classe prédite dans ce niveau.

### 4.2.2. Classification Multi-classes

La classification multi-classes désigne une tâche de classification avec plus de deux classes ( $k$  classe ou  $k > 2$ ). Dans la classification multi-classes, le but est d'affecter des instances à l'une des  $k$  classes possibles.

La classification hiérarchique s'attaque au problème de classification multi-classes en divisant l'espace de sortie, c'est-à-dire en un arbre. Chaque nœud parent est divisé en plusieurs nœuds enfants et le processus se poursuit jusqu'à ce que chaque nœud enfant ne représente qu'une seule classe.

### 4.3. Apprentissage profond

Deep-Learning ou l'apprentissage profond est un sous-domaine de l'apprentissage automatique qui s'intéresse aux algorithmes inspirés par la structure et la fonction du cerveau, appelés réseaux de neurones artificiels. Elle implique un type particulier de modèle mathématique qui peut être considéré comme une composition de blocs simples d'un certain type, dans une structure multicouche et où certains de ces blocs peuvent être ajustés pour mieux prédire le résultat final.

Dans l'apprentissage en profondeur, un modèle informatique apprend à effectuer des tâches de classification directement à partir d'images, de texte ou de son. Les modèles d'apprentissage en profondeur peuvent atteindre une précision de pointe, dépassant parfois les performances d'un niveau humain. Les modèles sont formés en utilisant une grande masse de données étiquetées et des architectures de réseau neuronal qui contiennent de nombreuses couches.

#### 4.3.1. Réseaux de neurones

Les méthodes d'apprentissage profond utilisent des architectures de réseaux de neurones, ce qui explique pourquoi les modèles d'apprentissage profond sont souvent appelés réseaux de neurones profonds (Deep Neural Networks – DNN).

Les premières formulations du réseau de neurones remontent à 1943. L'idée est de reproduire le fonctionnement d'un neurone humain, comme représenté sur la figure 4. Le fonctionnement d'un neurone formel est simple : c'est une somme pondérée d'entrées à laquelle on applique une fonction d'activation. Les coefficients de pondération sont appelés poids synaptiques et la fonction d'activation utilise un seuil : la sortie valant 1 si la somme pondérée dépasse le seuil et 0 sinon. On note  $b$  la valeur du seuil :

$$Y = \begin{cases} 1 & \text{si } \sum_k W_k i_k > b \\ 0 & \text{sinon} \end{cases}$$

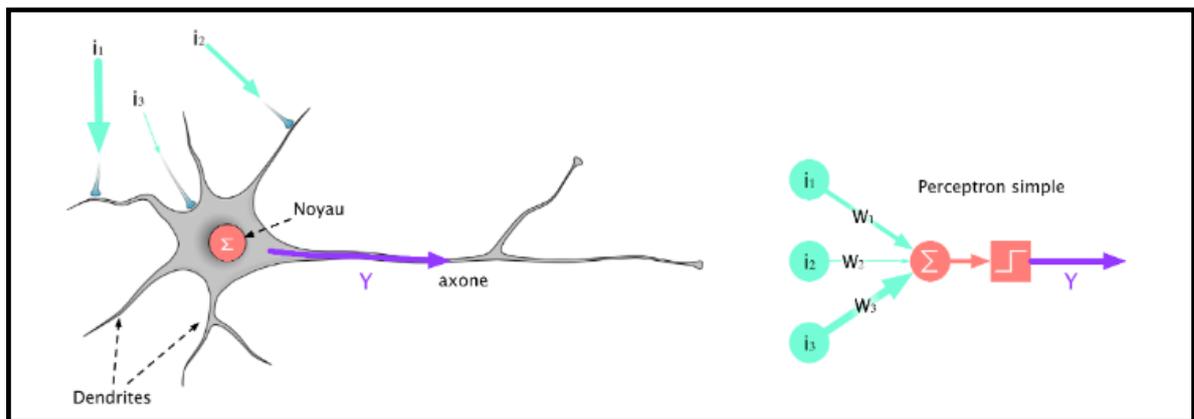


Figure 8 - Schéma d'un neurone à gauche et représentation d'un neurone formel à droite. [12].

La fonction d'entrée-sortie peut varier. Donc un neurone peut être :

- une **unité linéaire** : la sortie est proportionnelle à l'entrée pondérée totale;
- une **unité de seuil** : la sortie est fixée à l'un des deux niveaux, dépendamment de la puissance totale, si elle est supérieure à une valeur déterminée ou non;
- une **unité sigmoïde**, la sortie varie de façon continue, mais pas de façon linéaire.

### 4.3.2. Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN, ou ConvNet) est l'un des algorithmes les plus populaires dans le domaine de l'apprentissage en profondeur. Comme les autres réseaux de neurones, un CNN est composé d'une couche d'entrée, d'une couche de sortie et de nombreuses couches cachées (Figure 9). La distribution hiérarchique la plus courante des couches CNN contient :

- **Couche d'entrée**: le volume d'entrée est une image dont les dimensions sont les suivantes: [largeur x hauteur x profondeur]. Il s'agit d'une matrice de valeurs de pixels;
- **Les couches convolutives**: l'objectif d'une couche convolutif est d'extraire les caractéristiques du volume d'entrée;
- **Couches non linéaires**: la plupart du temps utilisant une fonction d'activation ReLU qui applique une activation élémentaire par seuillage à zéro;
- **Couches Pooling** (Figure 10): dans cette couche un sous-échantillonnage spatial est appliqué;
- **Couche entièrement connectée (FC)**: Les couches entièrement connectées connectent chaque neurone d'une couche à chaque neurone d'une autre couche.

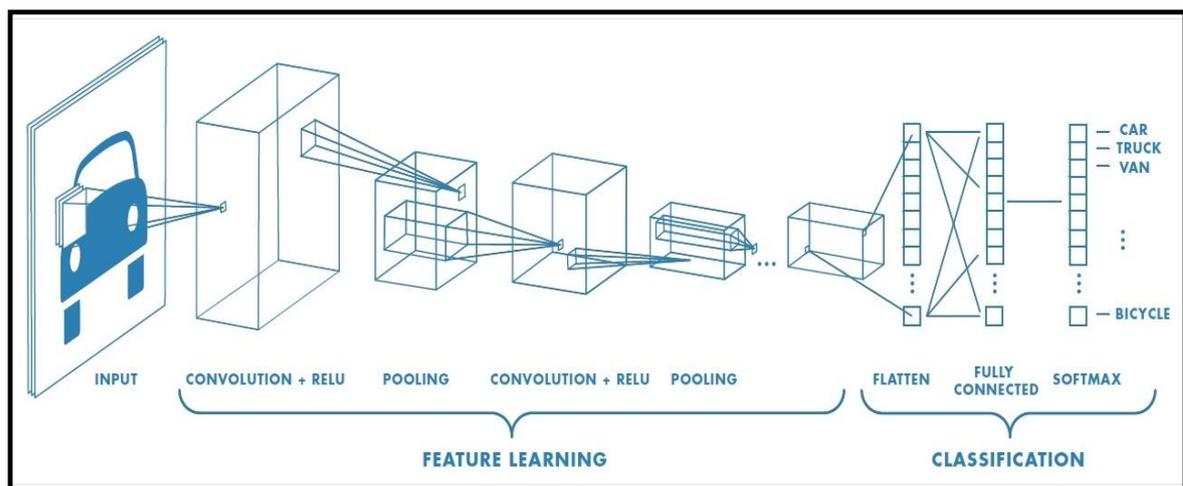


Figure 9 - Architecture d'un réseau de neurones convolutifs [24].

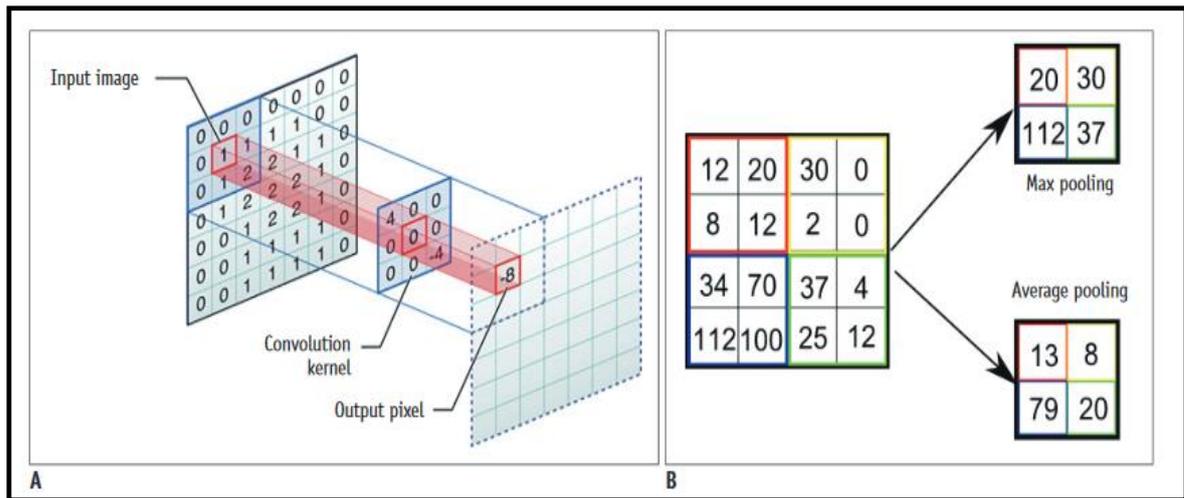


Figure 10 - Illustration des méthodes de convolution et de Pooling. A. Méthode de convolution B. Méthode de Pooling [24].

### 4.3.3. Réseaux de neurones convolutifs pour la classification d'images

La classification d'images a été l'un des sujets les plus importants dans le domaine de la vision par ordinateur, visant à donner aux machines la capacité de reconnaître des modèles et de s'adapter à différents environnements d'image. La reconnaissance faciale, la détection des véhicules, le diagnostic médical et la reconnaissance des chiffres sont des exemples typiques. Les réseaux de neurones convolutifs ont été largement utilisés dans ce domaine. Voici quelques exemples exploitant les CNNs pour la classification d'images:

- Le travail d'Alex Krizhevsky et al. [10] visant la création d'un «grand réseau de neurones convolutifs profonds» pour la classification d'images. Le modèle résultant est le premier modèle performant sur l'ensemble de données ImageNet.
- LeCun Yann, et al. [11] passent en revue diverses méthodes appliquées à la reconnaissance de caractères manuscrits en utilisant un CNN. Ce travail a exploité la base de données MNIST contenant un grand ensemble de chiffres manuscrits.
- S. Lawrence et al. [12] explorent l'idée d'utiliser un ConvNet pour la reconnaissance faciale afin d'extraire des fonctionnalités successivement plus grandes dans un ensemble hiérarchique de couches.
- Chuan Zhou et al [13] ont appliqué les CNNs à l'imagerie médicale en classant les régions d'intérêt sur les mammographies en masse ou en tissu normal.

### 4.3.4. Architectures de réseaux profonds

Il existe différents types d'architectures de CNNs tels que GoogleNet, Alexnet, ResNet, VGGNet... etc.

#### 4.3.4.1. AlexNet

AlexNet est la première architecture profonde qui a été introduite par l'un des pionniers de l'apprentissage en profondeur (Geoffrey Hinton et ses collègues). Il s'agit d'une architecture de réseau simple mais puissante, qui a ouvert la voie à une recherche révolutionnaire dans le Deep Learning telle qu'elle soit actuellement.

Alexnet dispose de huit couches. Les cinq premières sont convolutives et les trois dernières sont des couches entièrement connectées. Entre les deux, nous avons également des «couches» appelées Pooling et Activation. L'architecture AlexNet peut être vue sur la figure 11.

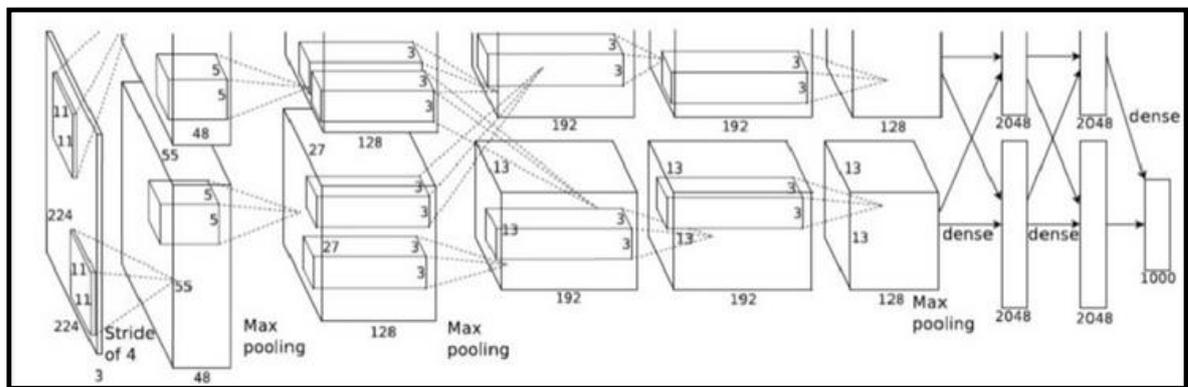


Figure 11 - L'architecture AlexNet [24].

#### 4.3.4.2. VGG Net

Le réseau VGG a été présenté par les chercheurs du Visual Graphics Group à Oxford (d'où le nom VGG). Ce réseau est spécialement caractérisé par sa forme pyramidale, où les couches inférieures les plus proches de l'image sont larges, tandis que les couches supérieures sont profondes.

Comme montré dans la figure 12, VGG contient des couches convolutives ultérieures suivies par des couches de regroupement. Les couches de regroupement sont responsables de rendre les couches plus étroites.

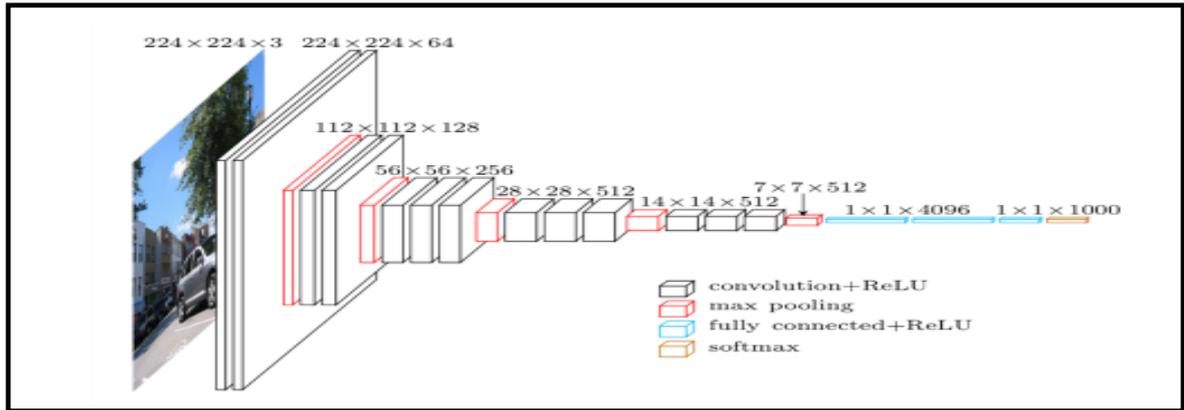


Figure 12 - L'architecture VGGNet [24].

#### 4.3.4.3. GoogleNet

GoogleNet (ou Inception Network) est une architecture conçue par des chercheurs de Google. Elle contient 22 couches par rapport à VGG qui dispose de 19 couches. Les chercheurs ont également développé une nouvelle approche appelée le module Inception (figure 13).

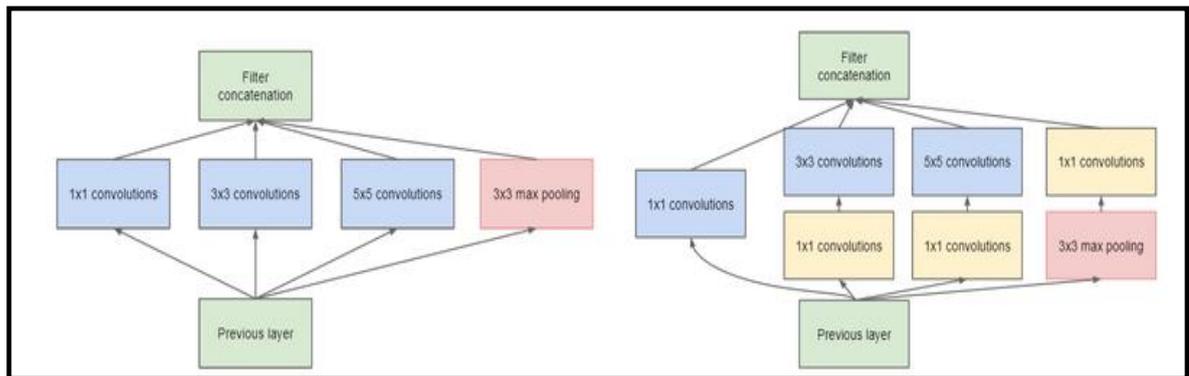


Figure 13 - Le module Inception [24].

L'architecture finale contient plusieurs modules d'Inception, comme le montre la figure 14 :

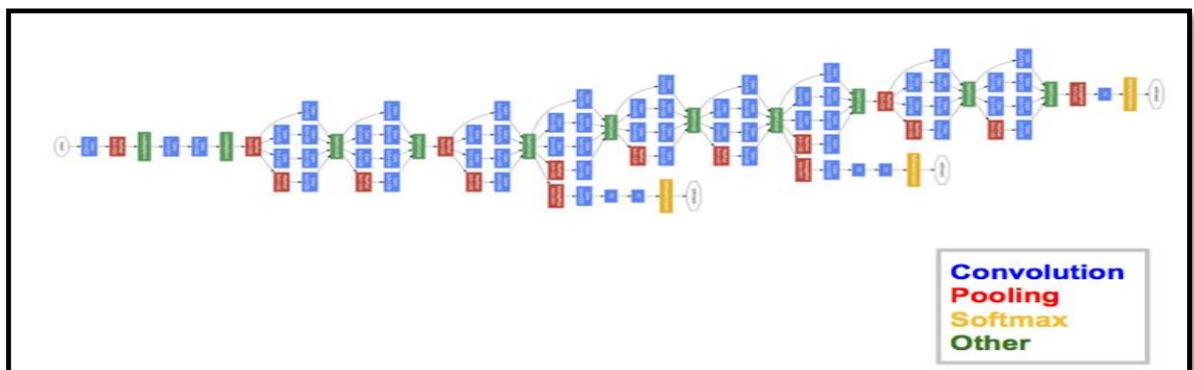


Figure 14 - L'architecture GoogleNet [24].

Les avantages de GoogleNet sont:

- GoogleNet s'entraîne plus vite que VGG.
- La taille d'un GoogleNet pré-entraîné est comparativement plus petite que celle de VGG.

#### 4.3.4.4. ResNet

ResNet est l'abréviation du nom composé *RESidual NETwork* (Réseaux résiduels). ResNet est l'une des architectures qui définissent véritablement la profondeur d'une architecture d'apprentissage en profondeur. Les ResNets sont constitués de plusieurs modules résiduels subséquents, qui constituent le bloc de construction de base de l'architecture ResNet. Ces modules résiduels sont empilés les uns sur les autres pour former un réseau complet.

Une représentation de l'architecture ResNet-50 est la suivante :

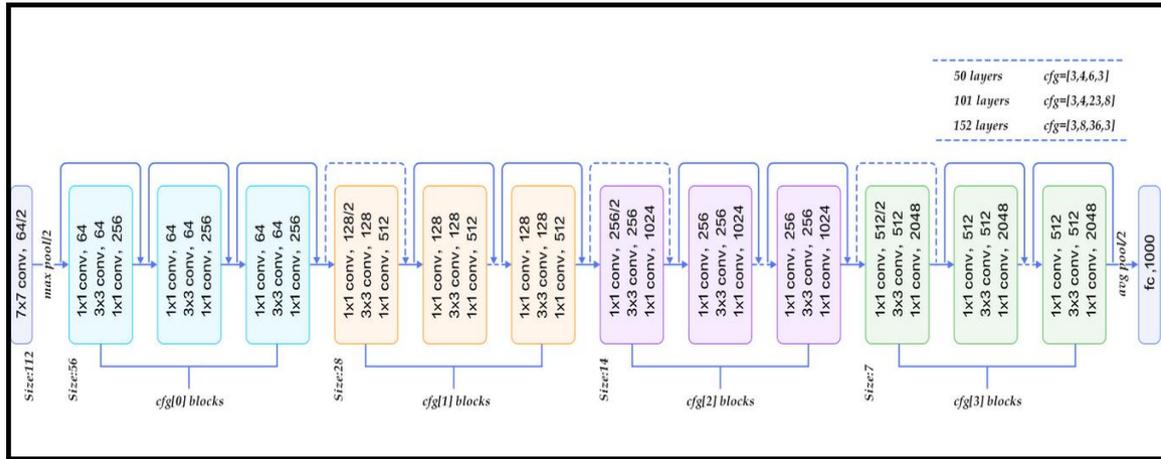


Figure 15 - l'architecture ResNet-50 [19].

#### 4.3.5. Apprentissage par transfert

L'apprentissage par transfert est une méthode d'apprentissage automatique où un modèle développé pour une tâche est réutilisé comme point de départ pour un modèle sur une deuxième tâche. C'est une approche populaire dans l'apprentissage en profondeur. Cette notion est sollicitée très souvent pour l'extraction de descripteurs de haut niveau. En effet, au lieu d'exploiter les prédictions finales des réseaux profonds, dans le transfert learning, on prend la sorties d'une couche intermédiaire proche de la dernière couche de sortie. Ce genre de sortie fournit généralement plusieurs valeurs qui sont pris comme une vectrice représentation le descripteur de l'image en question. Ces descripteurs sont exploités dans un nouveau stage d'apprentissage en utilisant généralement un autre type d'apprenants. Ce processus s'est montré plus efficace que l'idée de considérer les sorties finales des réseaux profonds via un seul stage d'apprentissage.

L'apprentissage par transfert s'est avéré utile dans les tâches d'analyse d'image et de vidéo. De plus, cette approche donne très souvent des résultats satisfaisants en particulier lorsque l'ensemble de données d'entraînement est relativement riche.

### 4.3.6. Descripteurs sémantiques

Les descripteurs sémantiques expriment des propriétés qui ont une signification forte. Ce sont des concepts qui peuvent être décrits en termes d'autres concepts. Leur construction nécessite le calcul de modèles d'apprentissage utilisant des descripteurs de bas niveau.

Dans un système de détection de concepts, le traitement utilisant des descripteurs de bas niveau est fait d'une manière complètement indépendante pour tous les échantillons et pour tous les concepts cibles. Une telle approche ne prend pas en compte les relations sémantiques et/ou statistiques entre les concepts cibles. Comme alternative à cette approche, on a une autre approche qui utilise les résultats de détection de plusieurs concepts pour construire un descripteur de détection d'un concept cible. Dans cette dernière, la phase d'apprentissage est faite sur plusieurs concepts différents du concept cible à détecter, ensuite on prend les résultats de détection de ces concepts pour construire un descripteur qui sera utilisé dans la détection du concept cible. Ce qui donne à cette approche la prise en considération des relations sémantiques et/ou statistiques entre les concepts. Cette approche de génération de descripteurs sémantiques est illustrée dans la Figure 16.

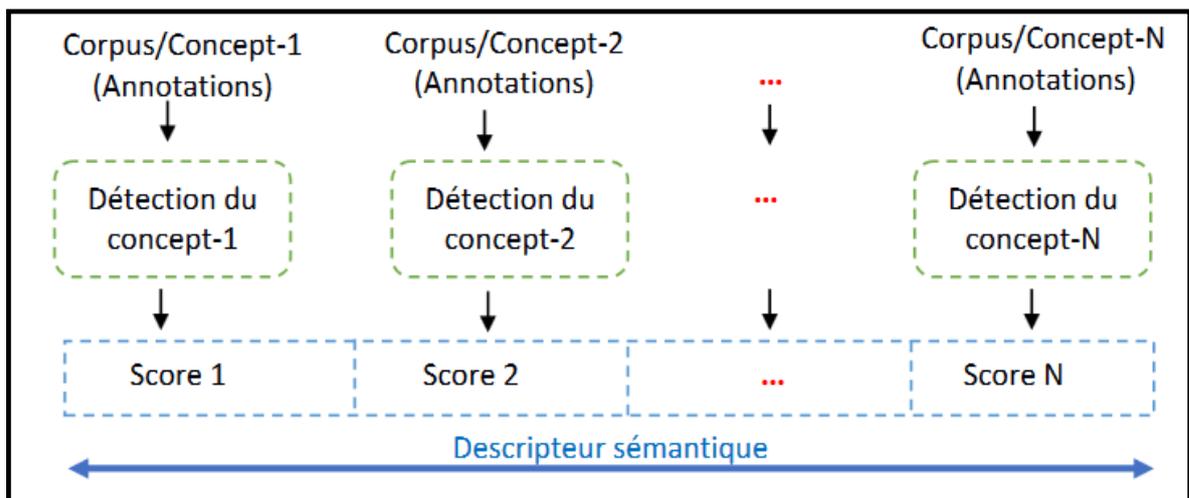


Figure 16 - Phases de construction d'un descripteur sémantique

### 4.3.7. Réseau neuronal récurrent

Un réseau neuronal récurrent (RNN) est une classe de réseaux de neurones qui comprend des connexions pondérées à l'intérieur d'une couche. Comme les RNN incluent des boucles, ils peuvent stocker des informations lors du traitement de nouvelles entrées. Cette mémorisation est rendue idéale pour les tâches de traitement où des entrées préalables doivent être prises en compte (telles que des données de séries chronologiques).

Les réseaux récurrents permettent de traiter des données séquentielles une suite d'entrées  $x$ . En effet au temps  $t$  ils calculent leur sortie en fonction de l'entrée  $x(t)$  mais aussi de l'état de la couche cachée au temps précédent.

Tous les réseaux neuronaux récurrents ont la forme d'une chaîne de modules répétitifs de réseau neuronal. Dans les RNN standard, ce module répétitif a une structure très simple, telle qu'une couche de « **tanh** » unique (figure 12).

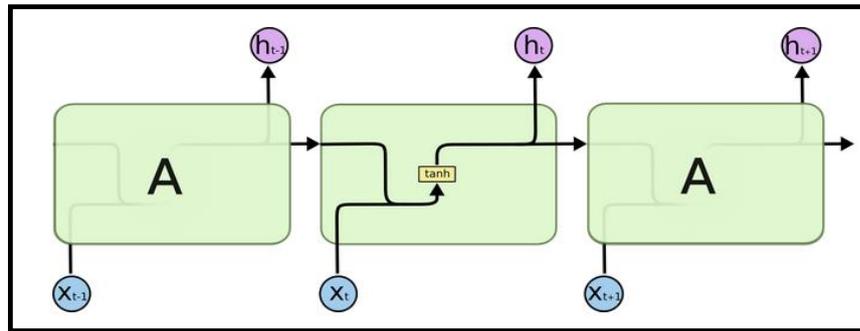


Figure 17 - Le module répétitif dans un RNN standards [25].

### 4.3.8. Réseaux de mémoire à longue durée

Les réseaux de mémoire à longue durée - généralement appelés "LSTM" - sont un type spécial de RNN, capable d'apprendre les dépendances à long terme. Ils ont été introduits par Hochreiter & Schmidhuber (1997). Ils fonctionnent très bien sur une grande variété de problèmes et sont maintenant largement utilisés. Les LSTM sont explicitement conçus pour éviter le problème de dépendance à long terme.

Les LSTM ont également la forme d'une chaîne de modules répétitifs, mais avec une structure différente de celle des RNN. Au lieu d'avoir une seule couche de réseau neuronal, il y en a quatre, interagissant d'une manière très spéciale.

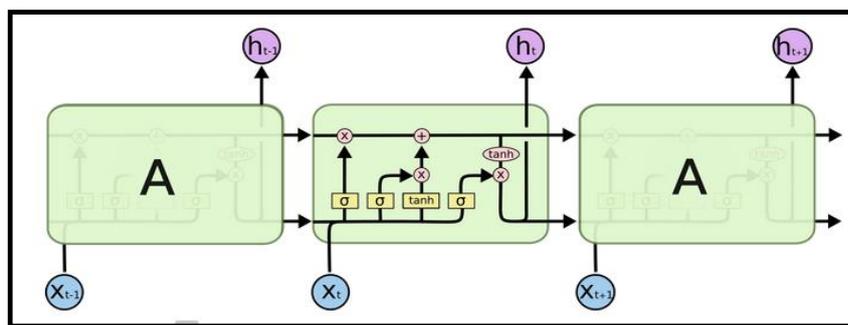


Figure 18 - Le module de répétition dans un LSTM [25].

### 4.3.9. Apprentissage profond pour l'analyse d'images médicales

Les chercheurs estiment que les images médicales représentent actuellement au moins 90% de toutes les données médicales, ce qui en fait la plus importante source de données dans l'industrie des soins de santé. Cela devient une quantité écrasante à l'échelle humaine, quand on considère que les radiologues dans les salles d'urgence de certains hôpitaux reçoivent

quotidiennement des milliers d'images. De nouvelles méthodes sont donc nécessaires pour extraire et représenter les données de ces images plus efficacement.

Bien que l'une des applications les plus courantes d'apprentissage machine dans le domaine médicale ait été en réalité l'imagerie médicale, ce n'est que récemment que des algorithmes d'apprentissage profond ont été introduits [15] Le tableau ci-dessous présente des résultats de certaines applications de Deep Learning dans l'analyse d'images médicales. La première colonne contient les tâches traitées, la seconde contient les noms des auteurs ayant réalisé le travail en question. La troisième colonne présente la performance atteinte par les différents travaux en termes de précision.

**Tableau 2 - Résultats de certaines applications de Deep Learning dans l'analyse d'images médicales**

	Auteurs des travaux	Précision (%)
Rétinopathie diabétique	Kathirvel	96
	Pratt et al.	75
	Haloï et al.	98
	Alban et al.	45
Détection d'éléments histologiques et microscopiques	Quinn et al.	100 pour le paludisme; 99 pour la tuberculose
	Dong et al.	98.66
Alzheimer et Parkinson	Suk	95.35
	Sarraf et al.	98.84
	Liu et al.	87.76

#### 4.3.10. NiftyNet: une plate-forme d'apprentissage profond pour l'imagerie médicale

NiftyNet [16] est une plate-forme de réseaux de neurones convolutifs open-source (CNN) basée sur TensorFlow dédiée à la recherche dans le domaine de l'analyse d'images médicales et la thérapie guidée par l'image. NiftyNet fournit une architecture d'apprentissage en profondeur de haut niveau avec des composants optimisés pour les applications d'imagerie médicale et des interfaces spécifiques pour la segmentation d'images médicales, la classification, la régression et la représentation. Niftynet a été appliqué dans l'état de l'art, surtout dans le domaine de la segmentation des images médicales et a donné des résultats très intéressants. L'avantage avec ce type de réseaux convolutifs est leur capacité à traiter des images médicales (Nifty, Dicom.) en entrée sans devoir les convertir aux formats classiques (JPG, ..). Cela constitue un point fort puisque dans la solution de convertir les images médicales en un format standard et de les traiter ensuite implique une perte considérable d'information et impose des prétraitements nécessaires et le traitement manuel des images de chaque dimension séparément.

## **5. Conclusion partielle**

Nous avons présenté dans ce chapitre la maladie de Tuberculose ainsi que ses différents types. Nous avons exploré également les différents prérequis en apprentissage machine pour le traitement de notre problématique. Nous présenterons dans le chapitre suivant un état de l'art sur les travaux ayant été réalisés pour la classification des types de tuberculose.

## **Chapitre 2 :**

### **Classification des types de tuberculose**

### 1. Introduction partielle

Dans cette section, nous présentons des travaux récents qui ont été réalisés dans la campagne d'évaluation *ImageCLEF*, en se focalisant sur la tâche de classification des types de tuberculose.

### 2. ImageCLEF et la tâche tuberculosis

*ImageCLEF*<sup>1</sup> est une campagne d'évaluation qui a été initialement lancée en 2003 comme une partie du forum d'évaluation CLEF (CrossLanguage Evaluation Forum). *ImageCLEF* s'intéresse principalement aux tâches d'analyse et de recherche d'images. La tâche sur la tuberculose (tuberculosis) s'est tenue pour la première fois en 2017 et a connu une participation très encourageante avec la présentation de résultats par neuf groupes [3].

Le but de la tâche de classification de types de tuberculose est de classer automatiquement chaque cas tuberculeux dans l'une des cinq catégories suivantes correspondant aux cinq types de tuberculose pulmonaire: Infiltrative, Focale, Tuberculome, Miliaire et Fibro-caverneux.

### 3. Organisation de la tâche

Dans la tâche de la classification des types de tuberculose, les organisateurs mettent à disposition un corpus d'apprentissage (Les étiquettes des données d'apprentissage sont fournies également) et définissent une mesure d'évaluation à utiliser pour mesurer la performance des systèmes. Pour participer, une équipe doit faire partie d'une institution officielle et signer une charte d'utilisation de données. Une fois inscrits, Les participants peuvent accéder et exploiter le corpus d'apprentissage pour entraîner leurs modèles. Ensuite, les organisateurs fournissent un corpus de test. Les classes des exemples de test ne sont pas fournies bien évidemment. Les participants doivent interroger leurs modèles afin de prédire les classes des différents exemples de test. Ces prédictions sont soumises sur la plateforme du défi avant un délai défini au préalable par les organisateurs. Les participants ont le droit de soumettre jusqu'à 10 soumissions. Les organisateurs évaluent les résultats et les publient avant d'organiser une conférence pour la présentation de l'ensemble des travaux effectués dans ce défi.

### 4. Données

L'ensemble de données utilisé comprend des images tomodensitométriques thoraciques (format de fichier NIFTI avec extension de fichier .nii.gz) de patients tuberculeux. Le type de tuberculose pour chaque patient est fourni comme donnée uniquement dans le corpus d'apprentissage. Le tableau 3 présente la composition des deux corpus d'apprentissage et de test par rapport aux cinq types de tuberculoses considérés. Ces corpus ont été fournis dans *ImageCLEF*<sup>2</sup> 2017 (tableau 3).

---

<sup>1</sup> : <http://www.imageclef.org>

<sup>2</sup> : <http://www.imageclef.org/2017/tuberculosis>

**Tableau 3 - Les données pour la tâche TBT dans ImageClef 2017 [3]**

Tupes de tuberculose	Corpus d'apprentissage	Corpus de Test
Type 1 (T1) - Infiltratif	140	80
Type 2 (T2) – Focal	120	70
Type 3 (T3) - Tuberculome	100	60
Type 4 (T4) - Miliaire	80	50
Type 5 (T5) - Fibro-caverneux	60	40

## 5. Mesures d'évaluation

Dans cette campagne, le coefficient kappa de Cohen et la précision moyenne (accuracy) sont utilisés comme mesures d'évaluation.

### 5.1. Matrice de confusion

Une matrice de confusion est une table qui est souvent utilisée pour décrire la performance d'un modèle de classification (ou «classificateur») sur un ensemble de données de test pour lesquelles les vraies valeurs sont connues. Dans le cas d'un classificateur binaire, on obtient :

		Classe prédite	
		Classe 0	Classe 1
Classe réelle	Classe 0	VN	FN
	Classe 1	FP	VP

Figure 19 - Matrice de confusion

Où:

- Vrai positif VP. Elément de la classe 1 correctement prédit
- Vrai négatif VN. Elément de la classe 0 correctement prédit
- Faux positif FP. Elément de la classe 1 mal prédit
- Faux négatif FN. Elément de la classe 0 mal prédit

### 5.2. Précision moyenne (Accuracy)

L'accuracy (ACC) est calculée divisant tous les prédictions correctes par le nombre total de l'ensemble de données :

$$ACC = \frac{VP + VN}{VP + VN + FP + FN}$$

La valeur de l'accuracy varie dans l'intervalle [0,1].

### 5.3. Kappa de Cohen

Le kappa de Cohen est un coefficient destiné à mesurer l'accord entre deux variables qualitatives ayant les mêmes modalités. Le calcul de cette mesure se fait de la manière suivante:

$$K = \frac{P_a - P_e}{1 - P_e}$$

Où :

- K: la valeur de la mesure Kappa de Cohen;
- $P_a$  est l'accord relatif entre codeurs et  $P_e$  la probabilité d'un accord aléatoire :

$$P_a = \frac{1}{n^2} \sum_{i=1}^p n_{i.} \cdot n_{.i} \qquad P_e = \frac{1}{n} \sum_{i=1}^p n_{ii}$$

$n$  : nombre d'unités statistiques.

$p$  : le nombre de catégories.

## 6. Quelques travaux de classification de types de tuberculose pulmonaire

Dans cette section, nous présenterons les méthodes proposées par les deux équipes SGEast et MEDGIFT UPB. Ce choix est justifié d'une part, par les bons résultats obtenus par ces deux équipes et d'autres parts, par la disponibilité de suffisamment de détails concernant leurs travaux par rapport aux autres équipes participantes.

### 6.1. Travail de SGEast

L'équipe SGEast [14] a adopté à la combinaison d'un CNN avec LSTM, qui sont bien connus pour leurs performances dans les images et les tâches de classification de séquences. Les auteurs proposent de transformer d'abord les scans en tranches (Slices en anglais, ce sont des images résultant de la conversion des images .nii vers le format jpg) et réalisent ensuite un prétraitement comme décrit dans la section (1.4.1.1). Ensuite, chaque slice prétraité d'un patient est utilisé pour entraîner un CNN. Pour le choix de l'architecture du réseau profond, les auteurs ont testé plusieurs choix. Leurs premières expériences ont montré que ResNet-50 surpassait GoogLeNet et Caffe-reference en termes de précision. Ensuite, LSTM est mis en œuvre en utilisant les fonctionnalités de ResNet-50. La figure 20 montre un aperçu de la méthodologie.

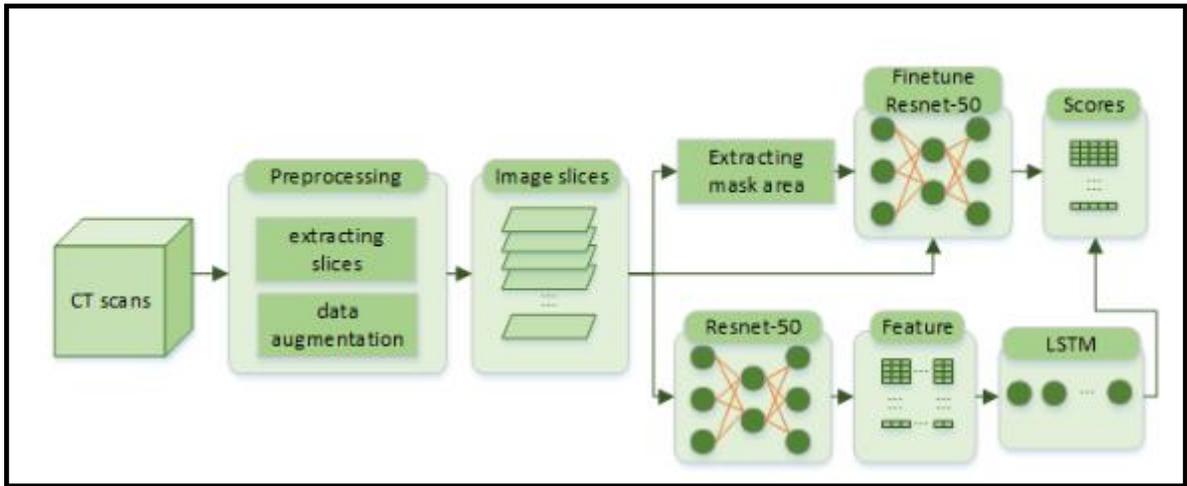


Figure 20 - La méthodologie de l'équipe SGeast [17].

### 6.1.1. Préparation des données

Pour la préparation de données, leurs auteurs ont imité ce que les médecins font habituellement lorsqu'ils interprètent des scanographies. Les tranches ont été extraites le long de la dimension de haut en bas de sorte qu'elles soient de taille  $512 \times 512$ . Afin de rassembler plus de données d'entraînement, chaque tranche d'image a été clonée en changeant le contraste et la luminosité, le flou, ainsi que la rotation. Le contraste et la luminosité ont été améliorés par le module Python Image-Enhance. Le paramètre d'amélioration est fixé à 1,5, afin de ne pas introduire beaucoup de dégradation. Un filtre en mode  $3 \times 3$  est utilisé pour rendre l'image floue, où la taille de la fenêtre est relativement petite. En considérant que les positions des poumons peuvent varier un peu d'un patient à l'autre, ils ont également exploité des images en leur appliquant une rotation de 5 degrés pour tenir compte des légères différences entre les patients. Visuellement, les tranches étaient toujours les mêmes après les améliorations.

Un autre ensemble d'entraînement est généré appelé ensemble d'entraînement basé sur les des fichiers appelés *masques*. Puisque seules les zones dans les poumons contiennent des informations pertinentes, ils ont utilisé les fichiers de masque fournis par *ImageCLEF* pour extraire la zone pulmonaire. Les fichiers de masque sont également des données 3D avec une valeur de 0, 1 et 2. La valeur 0 représente la zone non pertinente; 1 et 2 représentent respectivement le côté gauche et le côté droit du poumon. L'objectif de l'utilisation des fichiers de masque est de former uniquement la zone pulmonaire dans les images [14].

### 6.1.2. Apprentissage par transfert (Transfer Learning)

Dans L'apprentissage par transfert, les auteurs ont choisi le modèle CNN ResNet-50. L'ensemble d'entraînement original et l'ensemble d'entraînement masqué ont été utilisés pour entraîner les modèles de réseaux profonds ResNet-50.

Pour la phase d'entraînement (Training), les auteurs ont utilisé deux ensembles d'entraînement, l'ensemble original et l'ensemble masqué. Puisque les deux ensembles d'apprentissage sont constitués d'images 2D, les tranches d'image de tous les patients ont été mélangées et entraînées en tant qu'images individuelles. Toutes les tranches ont été redimensionnées à  $256 \times 256$  afin d'ajuster les tailles du lot et d'accélérer l'entraînement. Ils ont utilisé le Max-pooling au lieu de l'average-pooling pour que la couche de regroupement (pooling) puisse extraire la caractéristique des tranches pertinentes et réduire l'influence des données d'apprentissage bruyantes. Ils ont utilisé la descente de gradient stochastique (SGD) comme optimiseur. Pour affiner le CNN (fine-tuning), les couches après la dernière couche de regroupement (pooling) sont connues pour être affinées davantage par rapport aux autres couches du réseau de neurones, car les premières couches contiennent des caractéristiques génériques telles que des bords ou des blobs pertinents pour de nombreuses tâches de traitement d'image.

À l'étape du test, les tranches d'un patient sont introduites dans le réseau neuronal une par une et pour chaque tranche le réseau renvoie une sortie. Les auteurs ont calculé la moyenne des sorties des tranches d'un patient comme sortie finale. Il est à rappeler que l'entraînement sur des tranches individuelles introduit du bruit dans les données d'entraînement, car certaines tranches ne contiennent pas de tuberculose. Par conséquent, l'approche de la moyenne des scores au moment du test peut réduire l'impact des données d'entraînement bruyantes pour une sortie plus fiable [14].

### 6.1.3. Apprentissage sur séquence (Sequence Learning)

Les images extraites peuvent former une séquence d'images du début à la fin des poumons. De plus, comme les zones touchées par la tuberculose n'apparaîtront pas dans le poumon entier, certaines images seront considérées comme exemptes de traces de tuberculose. Par conséquent, les auteurs ont exploré l'idée d'utiliser une séquence d'images, qui montrent potentiellement différentes parties de la zone TB, mélangées avec des images sans TB pour rendre la classification plus robuste. Ainsi, ils ont utilisé le CNN pour générer les caractéristiques de l'image, puis LSTM pour la classification, que nous appellerons CNN-LSTM. L'architecture CNN-LSTM peut être considérée comme un processus en deux étapes. Premièrement, les images seront transmises dans le modèle CNN pour générer des vecteurs caractéristiques d'image. Ces vecteurs décrivent les images qui ont été transmises dans le CNN sous une forme numérique. Ensuite, ces descripteurs sont passés dans le modèle LSTM pour la classification. Ce processus a été illustré dans la figure 12 [14].

### 6.2. Travail de MEDGIFT UPB

L'équipe MEDGIFT UPB [15] ont extrait les tranches en utilisant une stratégie d'échantillonnage structurel clairsemée. Ensuite, ils ont utilisé un CNN (GoogleNet) pour effectuer l'extraction des caractéristiques de chaque tranche du volume CT des patients. Enfin, ils ont pris la moyenne des scores et utilisent une classification Softmax pour obtenir la classification de nouvelles instances.

#### 6.2.1. Préparation des données

Les auteurs ont réalisé une transformation des données en des images 2D RGB sur lesquelles ils ont appliqué une normalisation des valeurs et un mapping vers l'espace de couleur HSV. Après l'étape de prétraitement, les auteurs ont exploité une observation clé: les tranches consécutives sont très redondantes. Ils ont donc utilisé une stratégie d'échantillonnage structurel clairsemée pour améliorer la qualité de l'ensemble de données.

#### 6.2.2. Phase d'entraînement (Train)

Les auteurs ont utilisé un réseau GoogleNet. Les poids de réseau sont appris en utilisant la descente de gradient stochastique. À chaque itération, un mini-lot de 32 échantillons est construit en échantillonnant 32 tranches d'apprentissage à partir de l'ensemble de données.

## 7. Résultats de la classification de types de tuberculose

La figure 21 montre les résultats obtenus pour la sous-tâche de classification de types de tuberculose. L'évaluation a été réalisée sur l'ensemble d'images de Test en utilisant le coefficient Kappa de Cohen non pondéré et la précision de classification comme mesure d'évaluation. Les résultats sont triés par coefficient Kappa de Cohen dans l'ordre décroissant.

Group Name	Run	Kappa	ACC	Rank
SGEast	TBT_resnet_full.txt	0.2438	0.4033	1
SGEast	TBT_LSTM_17_wcrop.txt	0.2374	0.3900	2
MEDGIFT UPB	TBT_T_GNet.txt	0.2329	0.3867	3
SGEast	TBT_LSTM_13_wcrop.txt	0.2291	0.3833	4
Image Processing	TBT-testSet-label-Apr26-XGao-1.txt	0.2187	0.4067	5
SGEast	TBT_LSTM_46_wcrop.txt	0.2174	0.3900	6
UIIP	TBT_iiggad_PCA_RF_run_1.txt	0.1956	0.3900	7
MEDGIFT UPB	TBT_....GoogleNet_10crops_at_different_scales_.txt	0.1900	0.3733	8
SGEast	TBT_resnet_partial.txt	0.1729	0.3567	9
MedGIFT	TBT_Top1_correct.csv	0.1623	0.3600	10
SGEast	TBT_LSTM_25_wcrop.txt	0.1548	0.3400	11
MedGIFT	TBT_submitted_topBest3_correct.csv	0.1548	0.3500	12
BatmanLab	TBT_SuperVx_Hist_FHOG_lr_0.414000.csv	0.1533	0.3433	13
SGEast	TBT_LSTM_37_wcrop.txt	0.1431	0.3333	14
MedGIFT	TBT_submitted_topBest5_correct.csv	0.1410	0.3367	15
MedGIFT	TBT_Top4_correct.csv	0.1352	0.3300	16
MedGIFT	TBT_Top2_correct.csv	0.1235	0.3200	17
BatmanLab	TBT_submitted_bootstrap.csv	0.1057	0.3033	18
BatmanLab	TBT_submitted_top3_0.490000.csv	0.1057	0.3033	19
BatmanLab	TBT_SuperVx_Hist_FHOG_Reisz_lr_0.426000.csv	0.0478	0.2567	20
BatmanLab	TBT_submitted_top2_0.430000.csv	0.0437	0.2533	21
BioinformaticsUA	TBT_run0.txt	0.0222	0.2400	22
BioinformaticsUA	TBT_run1.txt	0.0093	0.1233	23

Figure 21 - Résultats obtenus par les participants à la Tâche de tuberculose (Imageclef 2017). [3]

Nous pouvons remarquer que SGEast a obtenu les meilleures performances dans cette tâche en termes de coefficient kappa de Cohen. Le meilleur résultat en termes de précision moyenne est obtenue par l'équipe "Image Processing" classée cinquième dans le classement de la figure 13.

La figure 22 contient les matrices de confusion de la meilleure série de chaque équipe. Ces matrices montrent le pourcentage de patients de chaque classe classés dans chaque type de TB. Lors de l'analyse de ces matrices, nous pouvons constater que la meilleure performance par type de TB se trouve dans les 4 premières équipes (SGEast, MEDGIFT UPB, traitement d'image et UIIP). Ces méthodes semblent être complémentaires. Les organisateurs de la tâche ont effectué une fusion tardive avec la meilleure soumission de ces 4 meilleures équipes. L'évaluation de cette approche a atteint une valeur Kappa de 0,3263 Cohen. Ce résultat est une amélioration importante par rapport à la meilleure soumission des participants et montre à quel point les approches de ces quatre participants sont complémentaires. Cependant, en analysant profondément les résultats, il a été constaté que méthode de fusion a principalement amélioré la performance pour le premier type T1. Dans toutes les autres classes, la performance est inférieure à la meilleure performance obtenue dans la classe correspondante par au moins l'une

des méthodes fusionnées. Dans tous les cas, la précision moyenne obtenue par la fusion est supérieure (0,4867) à celles de toutes les soumissions avec une marge importante.

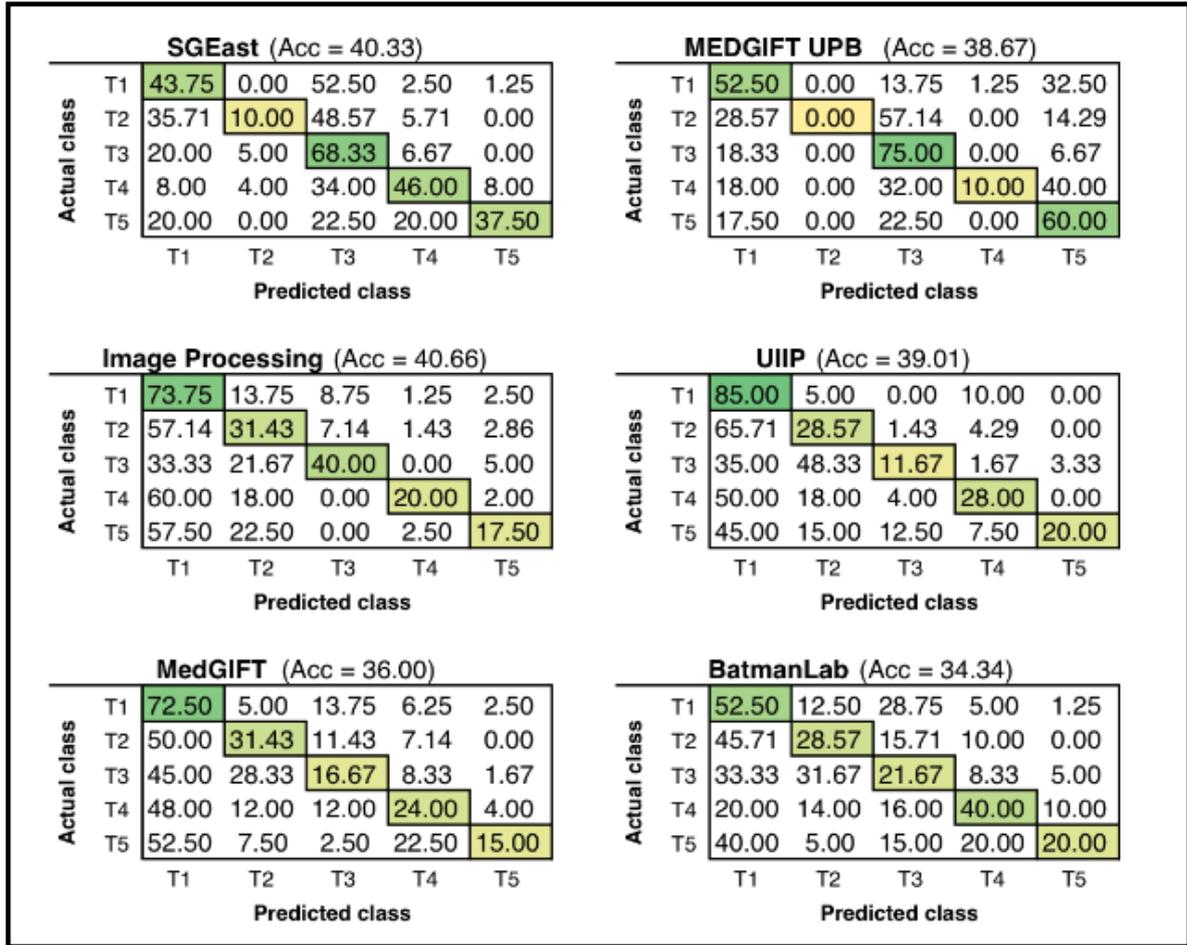


Figure 22 - Matrices de confusion de la meilleure exécution par groupe et la fusion des meilleurs 4 groupes. [3]

## 8. Discussion

Les résultats obtenus pour la classification de Tuberculose dans la tâche imageclef 2017 sont clairement basés sur l'adéquation des techniques d'imagerie. Les meilleures soumissions semblent exploiter des aspects très différents et chacun d'entre elles s'avère la meilleure pour une classe spécifique. Cependant, il y a encore place à l'amélioration, notamment dans les types Focal (T2), Tuberculome (T3), Miliary (T4) et Fibro-caverneux (T5). Pour cette tâche particulière, les méthodes d'apprentissage en profondeur ont mieux fonctionné que les autres approches. Néanmoins, la fusion effectuée par les organisateurs combinant la prédiction de trois techniques d'apprentissage en profondeur et celle basée sur des caractéristiques de texture 3D a dépassé la meilleure soumission des participants. Ceci suggère que différents types de TB peuvent être décrits par des approches différentes et qu'une telle approche peut conduire à des résultats optimaux [3].

## **9. Conclusion partielle**

Nous avons présenté dans ce chapitre la tâche Tuberculosis du défi international ImageClef 2017 dans laquelle plusieurs travaux de classification des types de tuberculose ont été considérés. Les résultats obtenus ont été intéressants, mais il est toujours possible d'améliorer les performances en adoptant différentes stratégies permettant de contourner les inconvénients des approches proposées. Nous présenterons dans le chapitre suivant nos contributions dans le cadre de ce travail.

# **Chapitre 03 :**

# **Contribution**

## 1. Introduction partielle

Dans le présent chapitre, nous allons présenter notre contribution en décrivant l'approche utilisée.

L'ensemble de ces expérimentations réalisées ont fait l'objet d'une participation aux deux tâches de tuberculose du défi international *ImageCLEF 2018*: Classification des types de tuberculose (TBT) et score de gravité de la tuberculose (SVR). En outre, le travail a abouti à une soumission d'un article de conférence (Annexe 1).

Nous décrivons dans ce chapitre ces deux tâches et nous détaillerons les expérimentations réalisées ainsi qu'une discussion des résultats d'évaluation obtenus.

## 2. Description de la tâche SVR

La tâche SVR vise à prédire le degré de sévérité des cas de tuberculose. Dans le cas d'un patient tuberculeux, l'objectif principal est de prédire son score de sévérité en fonction de son scanner 3D. Le degré de sévérité est modélisé selon 5 valeurs discrètes: de 1 («critique / très mauvais») à 5 («très bien»). La valeur du score est simplifiée de telle sorte que les valeurs 1, 2 et 3 correspondent à la classe " haute sévérité ", et les valeurs 4 et 5 correspondent à " faible sévérité ".

Le problème de classification est évalué à l'aide de courbes ROC (AUC) produites à partir des probabilités fournies par les participants. Pour le problème de régression, l'erreur quadratique moyenne (RMSE) est utilisée.

## 3. Contribution

Nous avons proposé d'extraire des descripteurs sémantiques à partir d'images par tomodensitométrie (images CT). 3D. Nous avons remarqué que les participants de la tâche TBT (*ImageCLEF 2017*) utilisaient chaque tranche (Slices en anglais) extraite comme un échantillon séparé. Ainsi, des centaines de slices sont considérées comme des échantillons d'apprentissage distincts alors que ces slices représentent le même patient. Cela introduit beaucoup de bruit. De plus, chaque slice recevra l'étiquette du patient (son type) même ceux dont le contenu ne présente aucune information permettant d'identifier le type de cas de tuberculose. Cela introduit plus de bruit. La majorité des participants [17] d'*ImageCLEF 2017* ont mis en évidence ce problème et son impact sur les résultats.

Pour surmonter ce problème, nous pensons que la solution la plus simple consiste à trouver un seul descripteur pour chaque patient. Ceci constitue l'idée clé de notre contribution.

Notre système proposé passe par trois étapes principales:

- Prétraitement des données d'entrée
- Extraction des descripteurs sémantique
- Apprendre un modèle de classification

### 3.1. Prétraitement des données d'entrée

Nous rappelons que dans les deux tâches, les images CT 3D sont fournies au format Nifti compressé. Premièrement, nous décompressons les fichiers et extrayons les tranches. À la fin, nous avons trois ensembles de slices correspondant aux trois dimensions de l'image 3D. Pour chaque dimension et pour chaque image Nifti, nous obtenons un nombre de slices allant de 40 à 400 images JPEG.

Le contenu visuel des images extraites des différentes dimensions n'est pas similaire. En effet, les images de chaque dimension sont prises sous un angle de vue différent. Nous avons remarqué, à partir de nos expériences, que les slices de la dimension «Y» donnent de meilleurs résultats que les deux autres (X et Z). Cependant, les étapes suivantes peuvent être appliquées aux slices de l'une des trois dimensions.

D'un autre côté, certain slices ne contiennent pas des informations pertinentes qui peuvent être utiles pour identifier les types de tuberculose. C'est pourquoi il est essentiel de filtrer les slices en ne conservant que celles qui peuvent être informatives et peuvent contenir des informations pertinentes. De plus, comme nous voulons extraire un seul descripteur par patient, il est essentiel de conserver le même nombre de slices pour chaque patient. Nous avons constaté qu'il y a habituellement un maximum de 60 slices visuellement informatives. Puisque les slices sont ordonnées, les 60 les plus informatives sont habituellement au centre de la liste. Nous proposons ensuite de conserver les 60 slices du milieu. Ce n'est pas optimal mais nous avons opté pour ce choix pour une approche entièrement automatique. Ce choix peut être amélioré en effectuant un filtrage manuel avec l'intervention d'un expert humain, de préférence avec des compétences médicales sur la maladie de tuberculose. La figure 23 résume le processus.

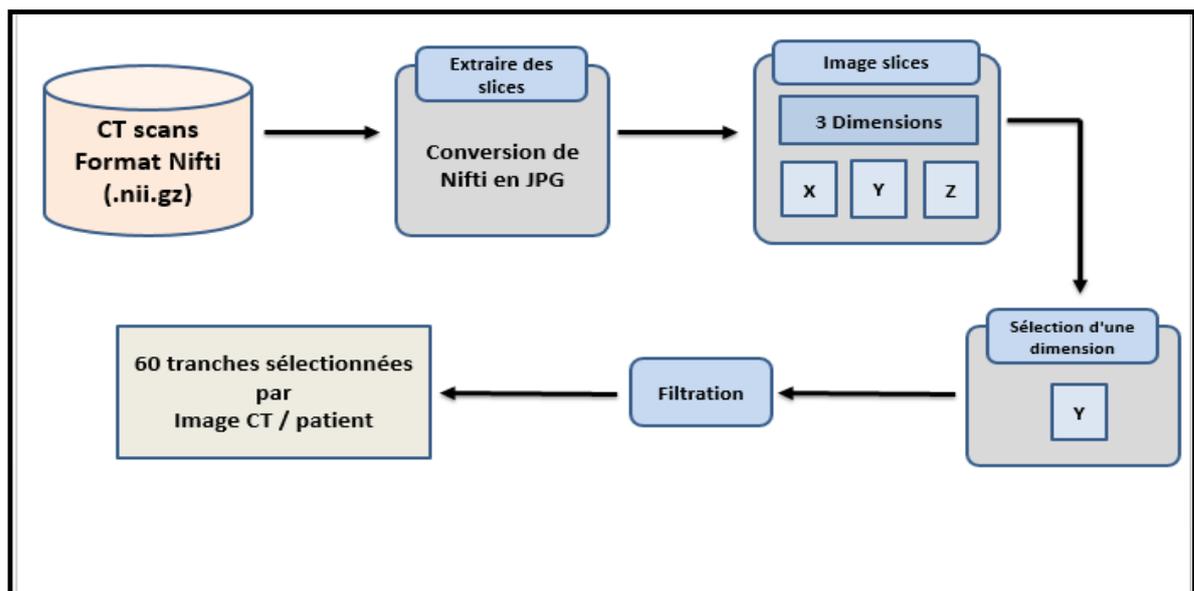


Figure 23 - Prétraitement des données d'entrée.

### 3.2. Extraction des descripteurs sémantiques

Après l'extraction et le filtrage des slices, nous proposons d'extraire un seul descripteur par patient. L'apprentissage par transfert présente dans ce contexte une piste intéressante qui peut être exploitée. Les résultats de SGEast [17] et même d'autres équipes dans la même tâche d'*ImageCLEF 2017* prouvent l'efficacité de cette approche [3, 17]. En effet, SGEast a opté pour l'apprentissage par transfert où ils ont exploité la sortie de l'architecture d'apprentissage en profondeur Resnet-50[19]. Cependant, cette idée présente un problème de taille de descripteur. En effet, par exemple, SGeast a considéré un descripteur par slice et non par patient. Cependant, puisque nous voulons avoir un seul descripteur, il est important que l'information extraite de chaque slice ne soit pas très grande. Par conséquent, nous proposons de décrire chaque slice par des informations sémantiques. Cette idée est inspirée par le travail présenté dans [31].

Nous choisissons d'exploiter les probabilités prédites par un apprenant profond formé sur l'ensemble des slices. Si  $K$  est le nombre de classes considérées, cette information correspond typiquement aux  $K$  valeurs de probabilité prédites pour les classes  $K$  (cinq probabilités des cinq types pour la tâche TBT, ou les cinq degrés de gravité pour la tâche SVR). On obtient alors pour chaque slice  $K$  des valeurs correspondant au nombre des classes considérées.

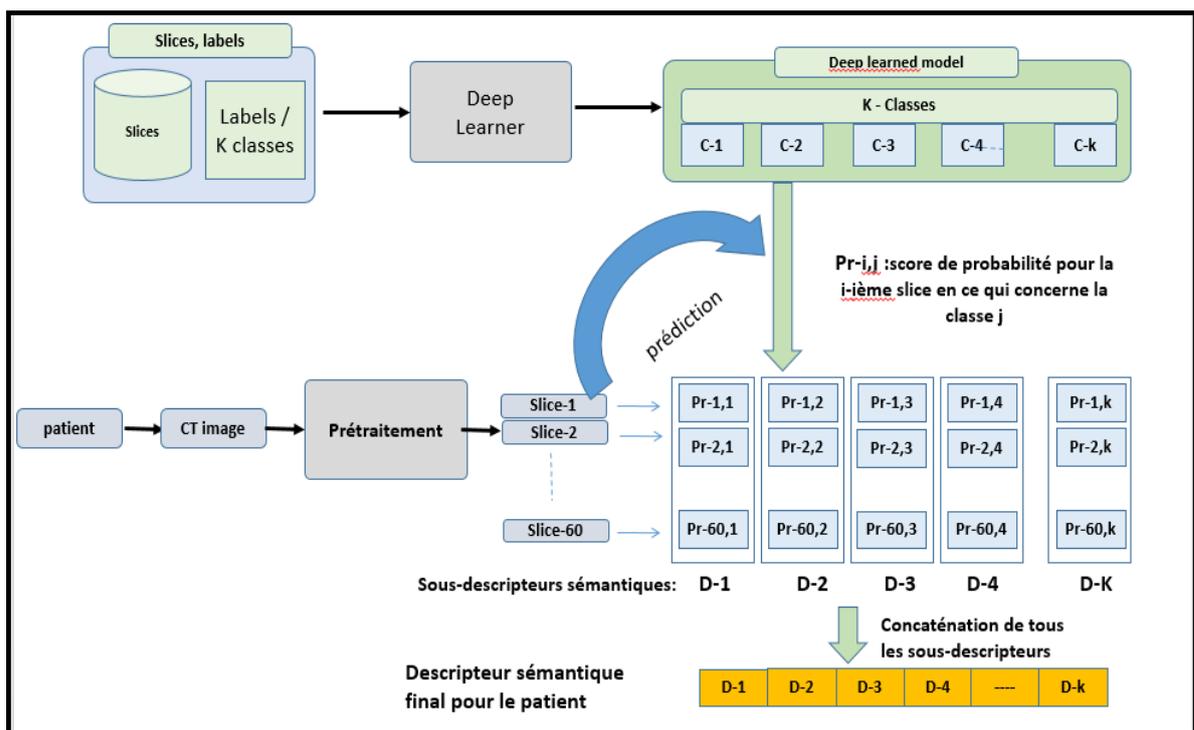


Figure 24 - Le processus d'extraction des descripteurs sémantiques

De plus,  $K$  sous-descripteurs sont générés:  $D_1, D_2, D_3, D_4, \dots, D_k$ . Chaque sous-descripteur  $D_i$  contient les probabilités prédites pour la classe  $i$  pour toutes les tranches du patient. Un descripteur sémantique final est construit en concaténant les sous-descripteurs  $K$ . La figure 24 détaille le processus d'extraction des caractéristiques sémantiques pour un patient.

### 3.3. Apprentissage d'un modèle de classification

Dans cette étape, nous proposons d'exploiter les descripteurs sémantiques des patients obtenus à l'étape précédente. Toute approche de classification supervisée peut être appliquée comme indiqué dans la figure 25.

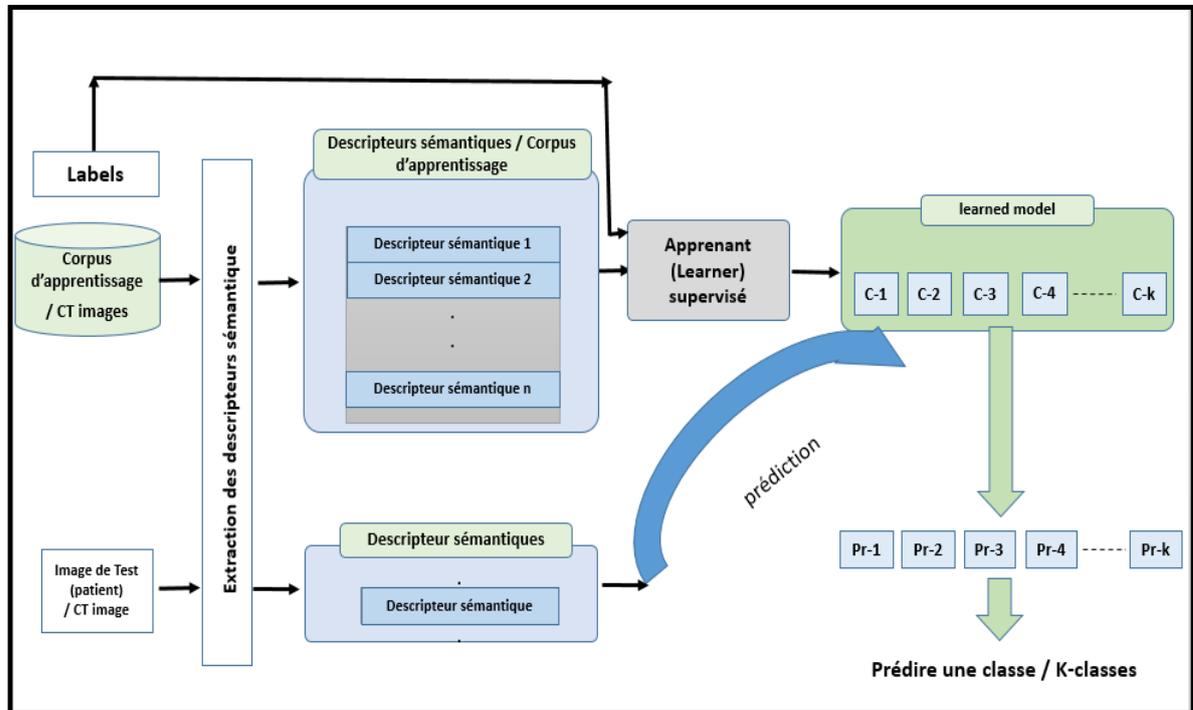


Figure 25 - Entraîner un modèle de classification basé sur les descripteurs sémantiques

Nous recommandons pour cette étape quelques idées:

- Utiliser un apprenant profond ayant comme entrée les descripteurs sémantiques des patients et leurs étiquettes (labels). Comme alternative, nous proposons d'utiliser une méthode d'ensachage qui collabore avec plusieurs apprenants et sous-échantillonne la collection de trains. Cela conduirait à de meilleurs résultats que nos expériences ont montrés.
- Pour appliquer une sélection d'échantillons, en particulier dans la tâche TBT où plusieurs images CT ont été fournies pour certains patients. Nous avons remarqué dans nos expériences que l'utilisation de toutes les images pour chaque patient introduisait beaucoup de bruit et donnait moins de bons résultats que l'utilisation d'une seule image par patient. Une alternative consiste à créer plusieurs sous-collections où chacune contient une image CT unique par patient, puis à générer un apprenant sur chaque sous-collection pour agréger leurs résultats. Cela conduirait probablement à un modèle beaucoup plus robuste.

## 4. Expérimentations et résultats

Nous décrivons dans les sections suivantes nos soumissions (Runs) soumis aux tâches TBT et SVR.

Nous avons implémenté l'approche du descripteur sémantique décrite dans la section 3. Nous avons utilisé pour cela les outils suivants:

- Le Framework Caffe pour l'apprentissage en profondeur;
- Weka pour tester plusieurs algorithmes d'apprentissage et de classification;
- med2image pour la conversion des images médicales Nifti au format JPEG.

Nous avons choisi d'utiliser des slices de dimension «Y» car nos expériences ont montré qu'elles sont plus adaptées que celles des deux autres et ont obtenu de meilleurs résultats.

Pour l'extraction des descripteurs, notre approche consiste à former un modèle profond pour générer des informations sémantiques. Malheureusement, nous avons eu des problèmes avec nos machines pour former notre apprenant en profondeur. En raison du manque de temps, nous n'avons pas pu réaliser le processus d'apprentissage. Comme alternative à cette étape, nous avons déployé le même modèle que celui proposé par l'équipe SGeast [17] lors de la tâche TBT *ImageCLEF 2017*. Le modèle est basé sur un Resnet50 [19] et a obtenu les meilleurs résultats à la tâche TBT de l'édition 2017. Nous avons donc exploité les sorties de la dernière couche (nommée prob) du Resnet-50 correspondant aux probabilités des 5 classes considérées.

### 4.1. Tâche TBT

#### 4.1.1. Données utilisées

L'ensemble de données utilisé dans la tâche TBT comprend des images tomodensitométriques des patients tuberculeux avec le type TB. Certains patients comprennent plus d'un scanner. Tous les scanners appartenant au même patient présentent le même type de TB. Le tableau 4 résume la répartition d'images CT selon les cinq types de tuberculose considérés.

Tableau 4 - Données fournies pour la tâche TBT

TB Types	Apprentissage		Test	
	#Patients	#CTs	#Patients	#CTs
Type 1	228	376	89	176
Type 2	210	273	80	115
Type 3	100	154	60	86
Type 4	79	106	50	71
Type 5	60	99	38	57
Total	677	1008	317	505

### 4.1.2. Protocole expérimental

Nous avons utilisé la collection d'apprentissage fournie par les organisateurs et nous l'avons divisée en deux sous-collections: 80% pour l'apprentissage et 20% pour la validation. Nous avons exploité dans toutes nos exécutions les descripteurs sémantiques générés comme décrit précédemment. Nous avons testé plusieurs algorithmes dans l'étape de classification. Nous avons finalement soumis trois soumissions principales (les trois meilleurs résultats obtenus sur le corpus de validation). Les autres soumissions sont des variantes ou sont générées par la fusion de certaines de ces trois soumissions (Runs):

- Soumission 1: Random Forest (forêts aléatoires) en tant que classificateur supervisé. Nous avons ajusté les deux paramètres en fonction du nombre d'itérations effectuées et du nombre d'entités sélectionnées de façon aléatoire.
- Soumission 2 : « Bagging + Random Forest » en tant que classificateur supervisé. Nous avons ajusté le nombre d'apprenants pour le bagging et les deux mêmes paramètres que « Soumission 1 » pour les forêts aléatoires;
- Soumission 3 : Une classification hiérarchique. Nous avons organisé les cinq classes 5 dans une structure hiérarchique comme décrit dans la figure 26. Nous avons créé deux nouvelles classes virtuelles V-1 et V-2. V-2 regroupe les deux classes Type 1, Type 2, et V-2 contient les classes Type 3, Type 4 et Type 5. Nous avons réorganisé nos collections afin d'obtenir une classification à deux niveaux différents. Dans la première étape, nous classons les échantillons dans deux classes virtuelles V-1 et V-2. Dans le deuxième niveau de classification, nous avons effectué une classification des échantillons concernant l'ensemble des classes de la classe prédite à l'étape précédente. Dans deux processus de classification, nous avons utilisé l'algorithme random forest en ajustant ses deux paramètres comme décrit pour « Soumission 1 ».

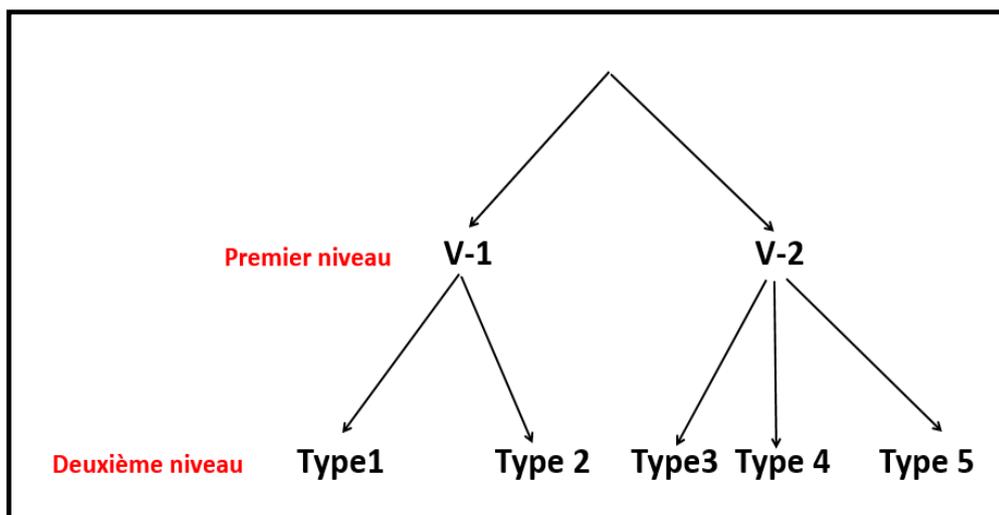


Figure 26 - Réorganisation hiérarchique des types TBT.

### 4.1.3. Résultats

Le tableau 5 montre les résultats obtenus par nos soumissions sur l'ensemble de validation.

**Tableau 5 - Résultats sur l'ensemble de validation pour la tâche TBT**

Soumissions	Kappa	Accuracy
Soumission 1	0,21	0,38
Soumission 2	0,25	0,41
Soumission 3	0,26	0,52

Le tableau 6 montre les résultats obtenus par nos soumissions sur l'évaluation réalisée par le comité *ImageCLEF*.

**Tableau 6 - Résultats sur l'ensemble de test pour la tâche TBT**

Soumissions	Kappa	Classement	Accuracy	Classement
Soumission 1	0,0412	28	0,265	29
Soumission 2	0,0275	29	0,2555	32
Soumission 3	0,0629	25	0,2744	27

Comme indiqué sur les résultats de validation, 3eme soumission obtient les meilleurs résultats sur l'ensemble de test (et de validation) par rapport aux 1<sup>er</sup> et 2eme soumissions.

Les figures 27 et 28 décrivent les résultats et le classement de toutes les soumissions sur la tâche TBT en termes de coefficient kappa et de précision, respectivement.

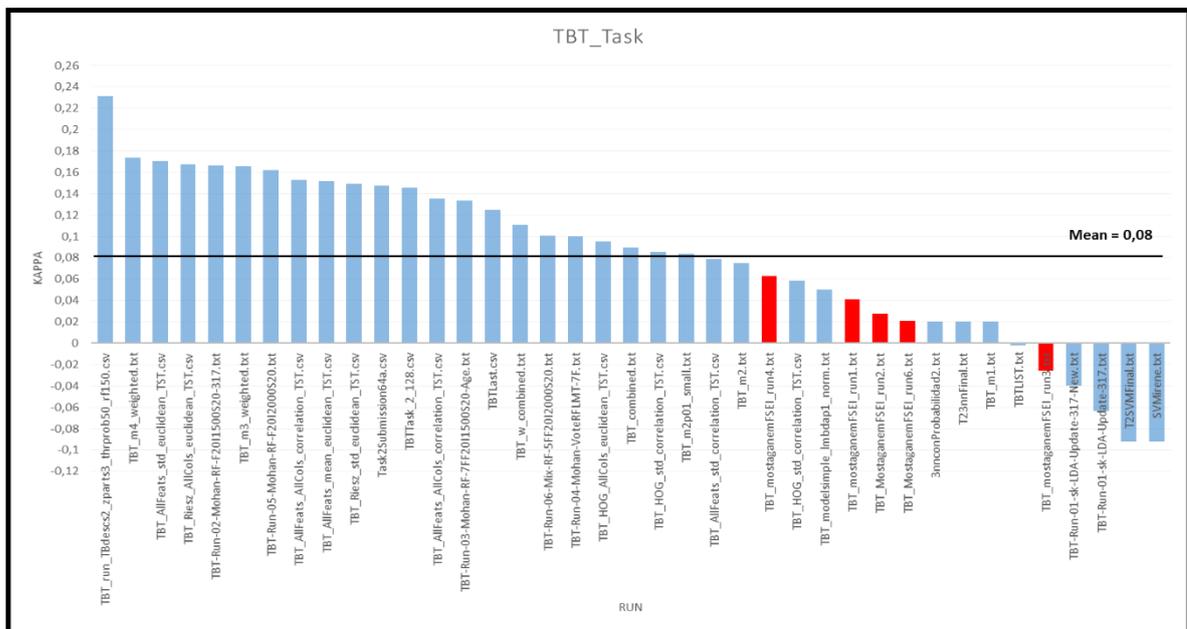


Figure 27 - Résultats et classement en termes de coefficient Kappa sur l'ensemble de test pour la tâche TBT

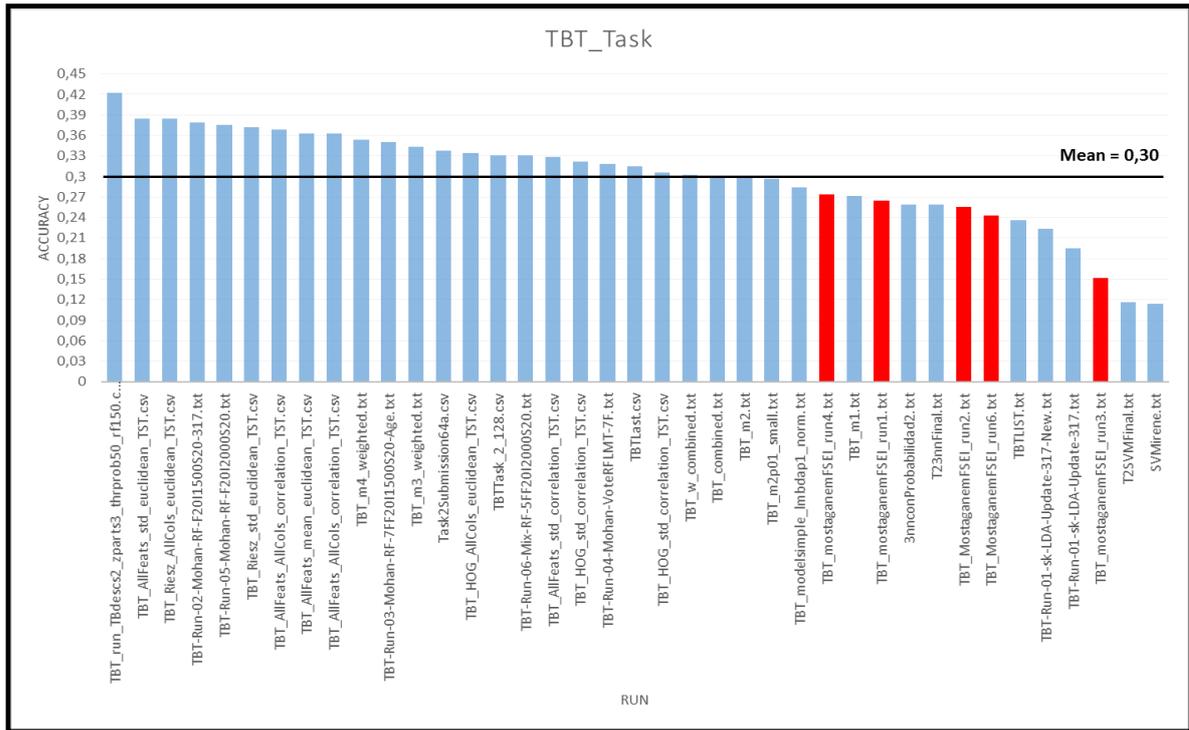


Figure 28 - Résultats et classement en termes de précision sur l'ensemble de test pour la tâche TBT.

Nous rappelons que nos descripteurs sémantiques ont été extraits en utilisant un modèle qui n'était pas très bien formé. En fait, nous avons rencontré des problèmes avec nos machines pendant la phase d'apprentissage de notre modèle de deep learning. En effet, bien que le modèle déployé de SGEast ait obtenu les meilleurs résultats à la tâche TBT ImageCLEF 2017, nous n'avons pas la possibilité d'effectuer exactement le même prétraitement effectué par cette équipe comme décrit dans [17]. Nous pensons que nos descripteurs sémantiques pourraient donner de meilleurs résultats s'ils sont extraits d'un modèle plus profond et mieux adapté.

## 4.2. Tâche SVR

### 4.2.1. Données utilisées

L'ensemble de données pour la tâche de SVR comprend des tomodensitométries thoraciques de patients TB ainsi que le score de sévérité correspondant (1 à 5). Les scores de 1 à 3 correspondent à la sévérité "Haute" alors que les deux scores 4 et 5 se réfèrent au degré de sévérité "Faible". Le tableau 7 résume la distribution d'images CT selon deux classes de gravité.

**Tableau 7 - Données fournies pour la tâche SVR.**

	Apprentissage	Test
sévérité Faible	90	62
sévérité Haute	80	47
Total	170	109

### 4.2.2. Protocole expérimental

Nous avons généré dans un premier temps les descripteurs sémantiques suivant l'approche décrite dans la section 3. Pour la prédiction des scores de sévérité TB, nous avons traité le problème comme un problème de classification. Nous avons utilisé pour cela, deux approches:

- Problème de classification multi-classes: nous avons considéré les cinq scores comme des classes séparées. Nous avons ensuite testé plusieurs classificateurs. Nous en avons sélectionné deux qui ont été les plus efficaces comparés à ceux qui ont été testés: Random Forest, Bagging d'un ensemble d'apprenants Random Forest.
- Classification hiérarchique: Nous avons organisé nos données afin d'effectuer une classification hiérarchique. Nous avons considéré la hiérarchie décrite dans la figure 29. Ensuite, une classification hiérarchique à deux niveaux est effectuée. Au premier niveau, les échantillons sont classés en classes "High" ou "Low". Au deuxième niveau, les échantillons sont reclassés dans les classes décroissantes de celui prévu dans le premier niveau.

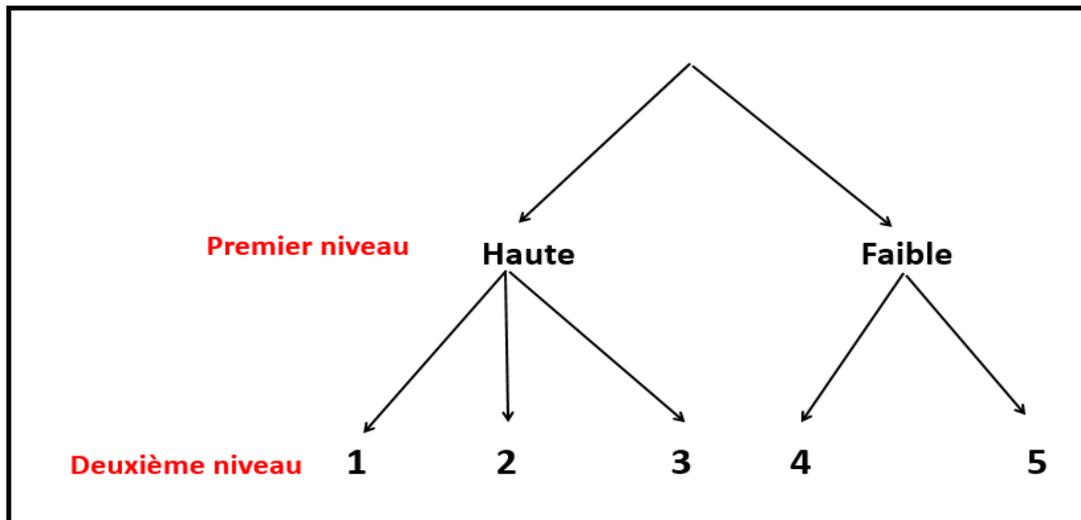


Figure 29 - La hiérarchie des classes considérées pour la tâche SVR

- Soumission 1 : Modèle multi-classe utilisant la forêt aléatoire comme classificateur. Nous avons ajusté les deux paramètres: le nombre d'itérations effectuées et le nombre de caractéristiques choisies au hasard;
- Soumission 2 : Modèle multi-classe utilisant un Bagging d'un ensemble d'apprenants Random Forest avec sous-échantillonnage de l'ensemble d'apprentissage principale. Nous avons créé deux sous-ensembles en équilibrant le nombre d'échantillons pour les 5 classes. Nous avons ensuite fusionné les résultats obtenus par les deux sous-ensembles;
- Soumission 3 : Classification hiérarchique en utilisant un Bagging d'un ensemble d'apprenants type "Forets aléatoires" dans chaque niveau du processus de classification hiérarchique ;
- Soumission 4 : fusion de Soumission 1 et Soumission 2 ;
- Soumission 5 : fusion de Soumission 1 et Soumission 3.

### 4.2.3. Résultats

Le tableau 8 montre les résultats obtenus par nos soumissions sur la collecte de validation.

Tableau 8 - Résultat sur l'ensemble de validation pour la tâche SVR en termes de precision moyenne et RMSE

Soumissions	Accuracy	RMSE
Soumission 1	0,41	0,37
Soumission 2	0,36	0,45
Soumission 3	0,56	0,3
Soumission 4	0,42	0,36
Soumission 5	0,48	0,34

Le tableau 6 montre les résultats obtenus par nos soumissions sur l'évaluation réalisée par le comité *ImageCLEF*.

Tableau 9 - Résultats sur l'ensemble de test pour la tâche SVR

Soumissions	RMSE	Classement	AUC	Classement
Soumission 1	1,0227	19	0,5971	26
Soumission 2	1,0837	22	0,6127	22
Soumission 3	0,9721	12	0,5987	25
Soumission 4	1,0137	18	0,6107	24
Soumission 5	1,0046	16	0,6119	23

Nous pouvons voir que notre 3eme soumission a obtenu les meilleurs résultats en termes de RMSE par rapport à nos autres soumissions sur l'ensemble de validation et même sur l'ensemble de test. Cependant, en termes d'AUC, la 2eme soumission semble être plus efficace.

Les figures 30 et 31 décrivent les résultats et le classement de toutes les soumissions sur la tâche SVR en termes de valeurs RMSE et AUC, respectivement.

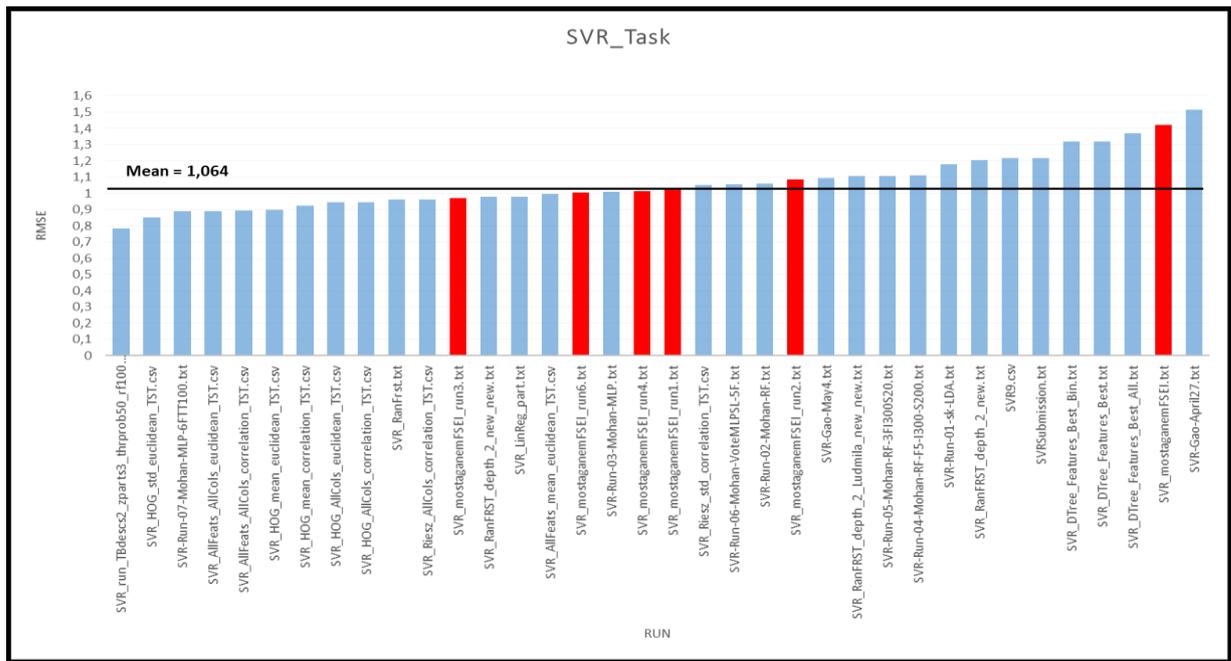


Figure 30 - Résultats et classement en termes de RMSE sur l'ensemble de test

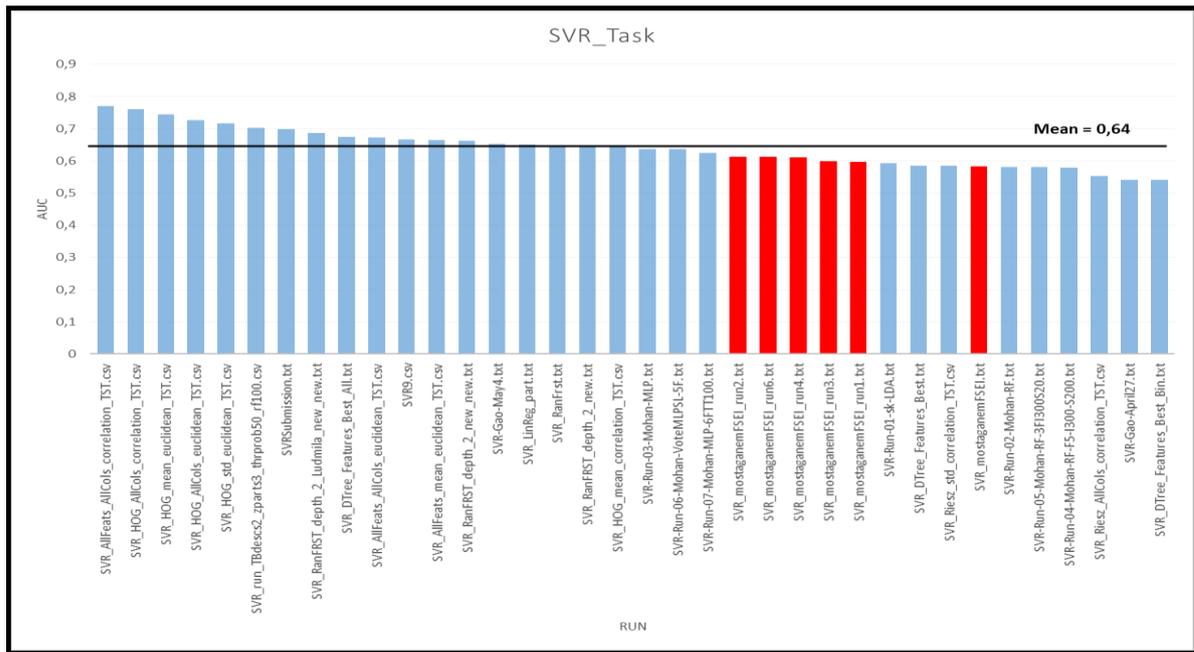


Figure 31 - Résultats et classement en termes d'AUC sur l'ensemble de test.

Nous pouvons voir que notre meilleur résultat est classé 12e sur 36 soumissions. Cependant, la différence entre les performances des 12 meilleures soumissions n'est pas très significative. Nous rappelons que notre meilleur résultat est obtenu par une approche de classification hiérarchique utilisant un Bagging d'un ensemble d'apprenants Random Forest à chaque niveau de la hiérarchie. Nous croyons que notre approche pourrait donner de meilleurs résultats en utilisant un modèle profond bien entraîné dans l'étape d'extraction des caractéristiques sémantiques

## 5. Conclusion partielle

Dans ce chapitre, nous avons décrit nos contributions dans le cadre de ce projet. Nous avons proposé une approche qui consiste à extraire un seul descripteur sémantique pour chaque « image CT » / Patient pour faire l'apprentissage. Ensuite nous avons détaillé les différentes expérimentations ainsi que les résultats obtenus sur l'ensemble de validation et sur l'évaluation réalisée par le comité *ImageCLEF*. Ces résultats montrent que cette approche pourrait donner des résultats plus intéressants si elle est appliquée de manière optimale. Dans ce qui suit nous allons présenter les différentes étapes de conception et d'implémentation de notre système.

**Chapitre 04 :**

**Conception et Implémentation**

## 1. Introduction partielle

Dans ce chapitre nous abordons les aspects de conception de notre système, en montrant les différents diagrammes de conception, qui décrit l'architecture et le fonctionnement de notre système de classification de types de tuberculose. Nous entamons ensuite la présentation de l'environnement de travail, en détaillant les différents outils utilisés. Nous allons présenter aussi, les différentes interfaces principales du système.

## 2. Conception

Nous avons modélisé les besoins de notre système de classification de types de tuberculose, en se basant sur le langage de modélisation UML «Unified Modeling Language». Nous avons choisi pour cette modélisation deux diagrammes principaux, le diagramme de cas d'utilisation et le diagramme d'activité, qui représentent les besoins fonctionnels ainsi que l'architecture du système.

Nous avons utilisé comme outils de modélisation UML, l'outil StarUML qui est un logiciel de modélisation, qui aide à tracer les différents diagrammes UML. Cet outil donne la possibilité d'exporter les diagrammes réalisés en images de format «png» ou «jpg» afin de les insérer au sein du document. La Figure montre l'interface principale de l'outil de modélisation StarUML version 5.0.

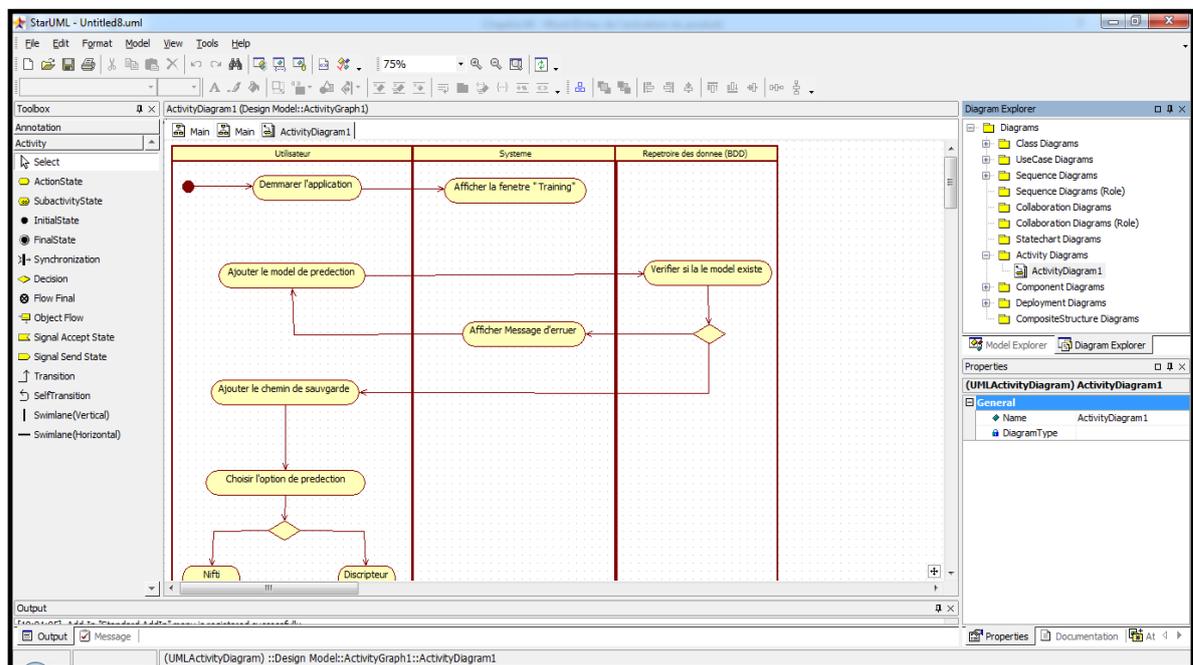


Figure 32 - Interface de StarUML version 5.0

## 2.1. Diagramme de cas d'utilisation

Nous avons commencé la modélisation par le diagramme de cas d'utilisation, qui permet de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités du système.

La figure 33 représente le diagramme de cas d'utilisation générale qui fait preuve d'un acteur appelé utilisateur qui admet les fonctionnalités suivantes :

- Convertir les fichiers Nifti en jpg ;
- Extraire des descripteurs sémantique ;
- L'apprentissage ;
- La prédiction.

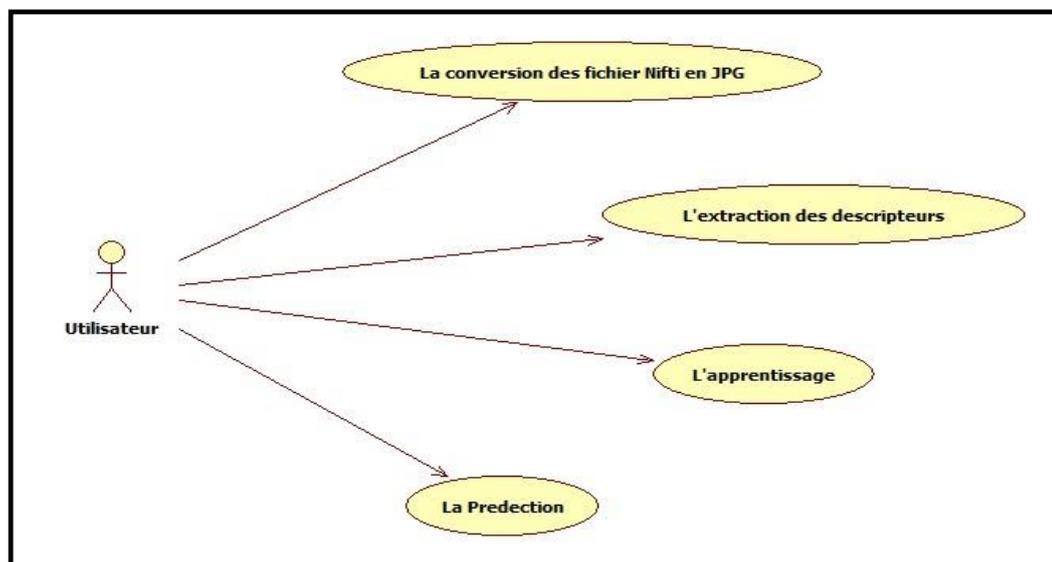


Figure 33 - Diagramme de cas d'utilisation

Dans la section suivante, nous allons détailler le fonctionnement de chaque cas à travers les diagrammes d'activités.

## 2.2. Diagramme d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Les figures suivantes montrent les diagrammes d'activités de chaque cas (ou bien interface de notre application) :

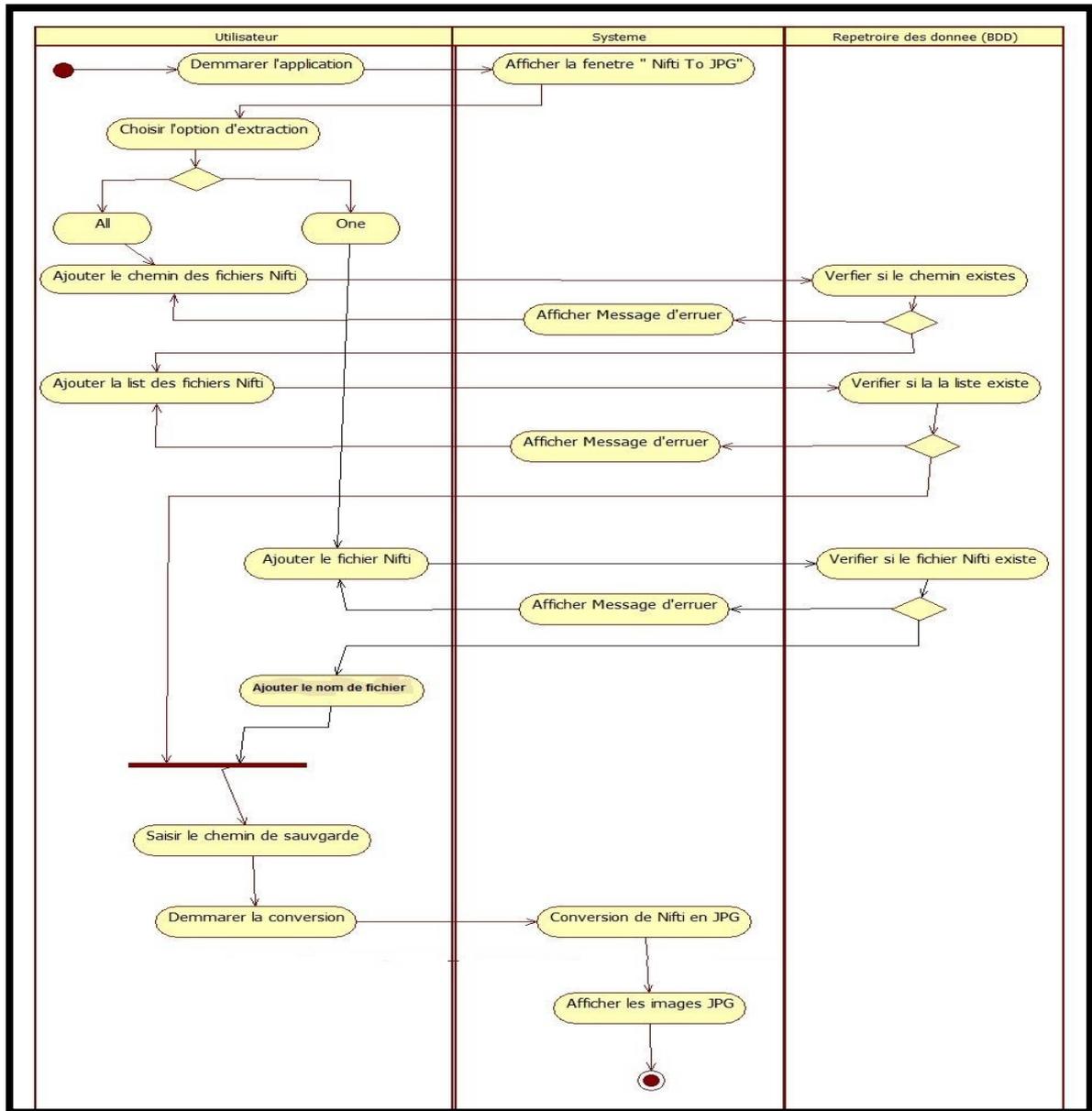


Figure 34 - Diagramme de séquence de la conversion (Nifti en JPG)

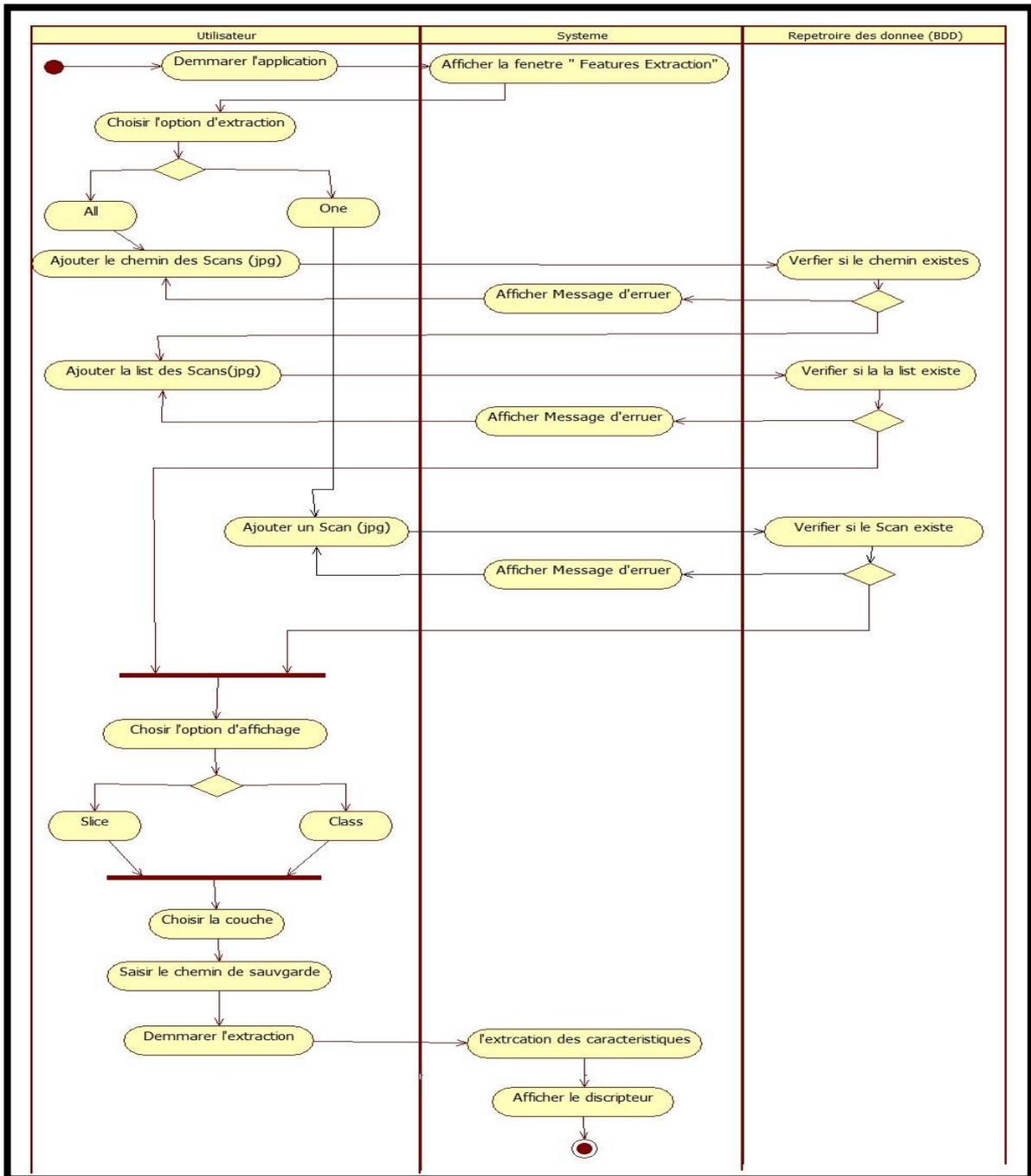


Figure 35 - Diagramme de séquence de l'extraction des descripteurs (Features Extraction)

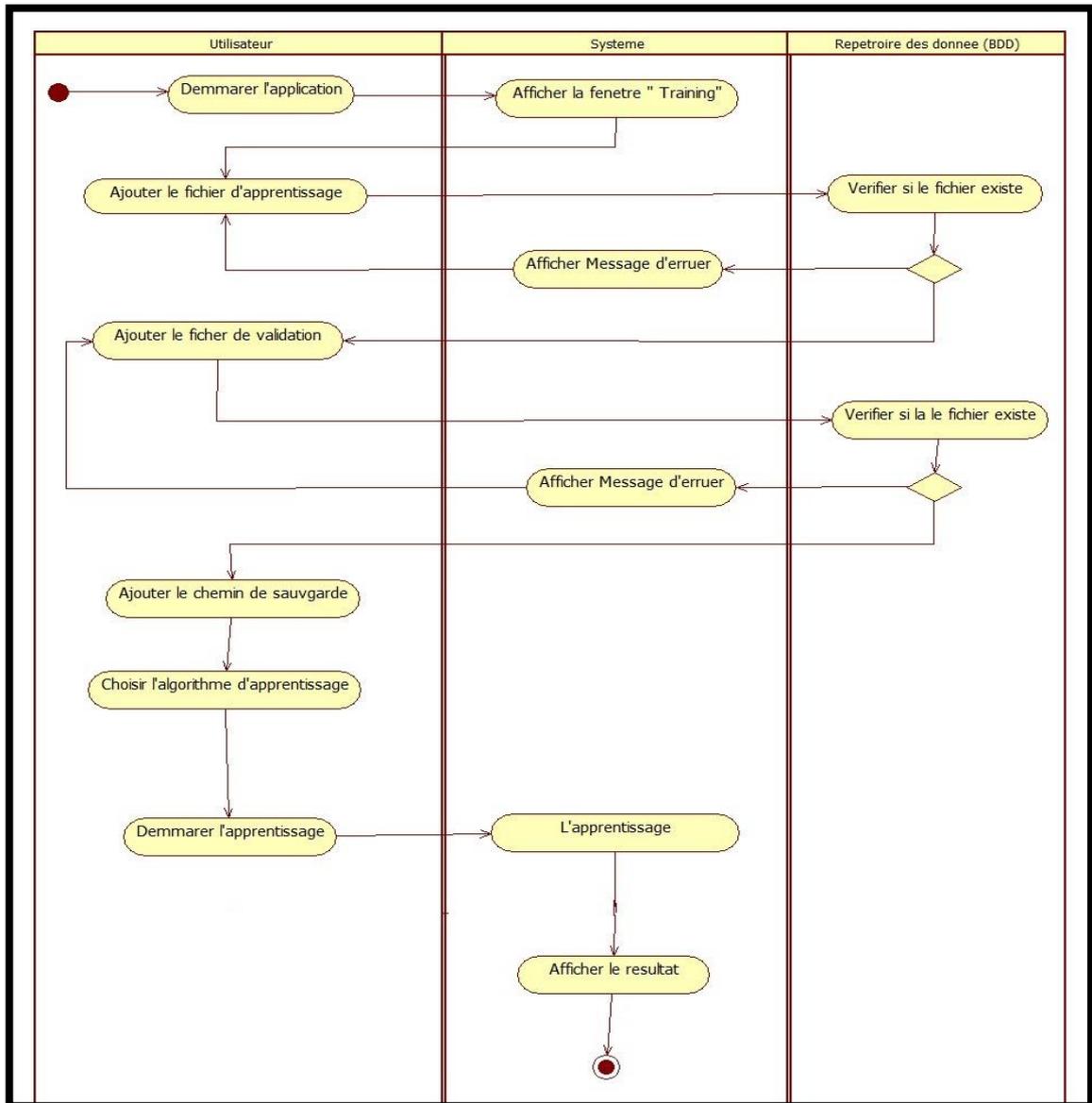


Figure 36 - Diagramme de séquence de l'apprentissage (Training)

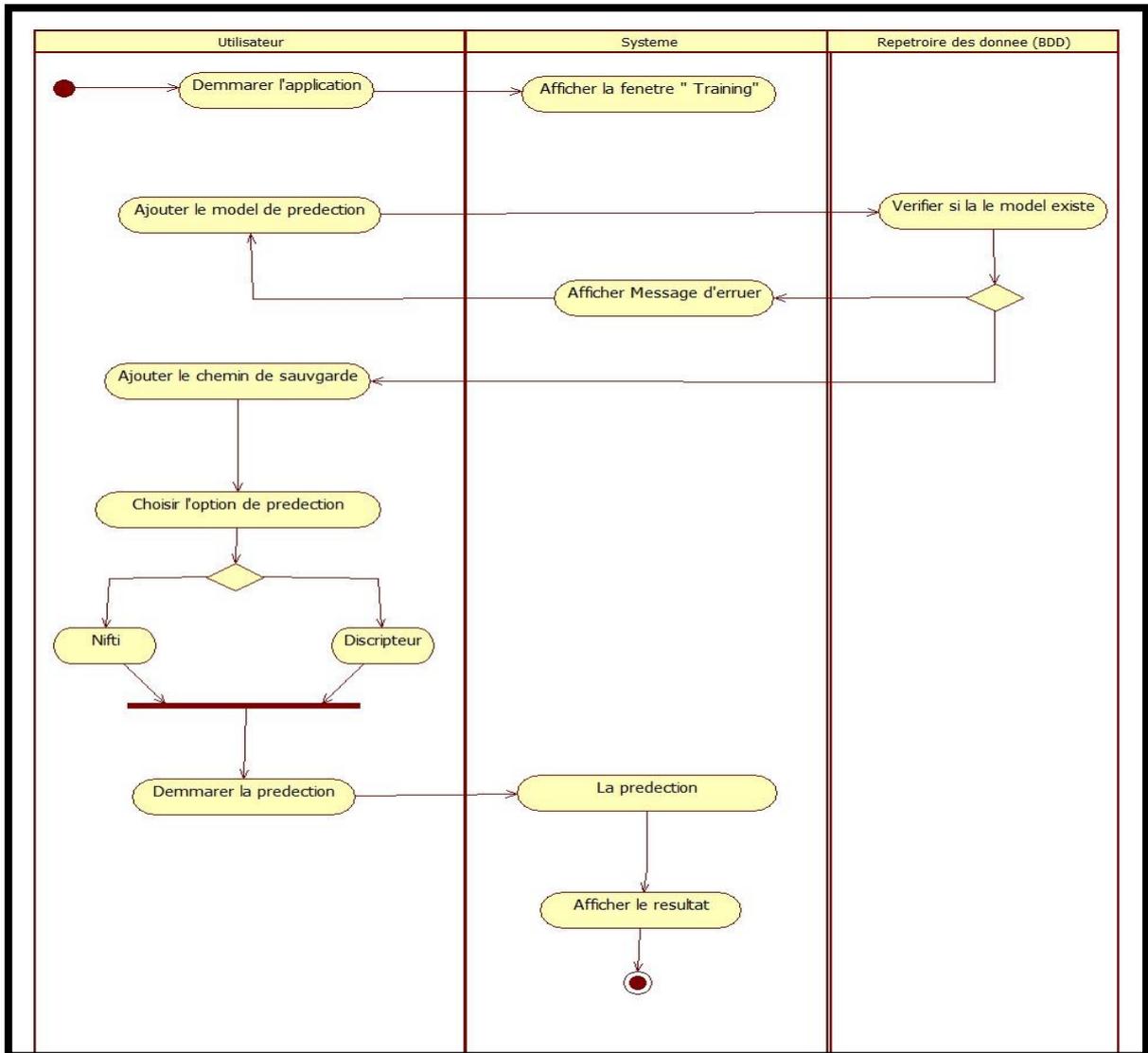


Figure 37 - Diagramme de séquence de la prédiction (Predection)

## 3. Implémentation

### 3.1. Environnement de travail

Pour l'implémentation du système, plusieurs bibliothèques et technologies ont été nécessaires.

#### 3.1.1. Environnement matériel

Pour le développement, un ordinateur portable Lenovo avec les caractéristiques suivantes :

1. Processeur : Intel Core i3-4030U CPU @ 1.90 GHz
2. RAM : 8 GO

#### 3.1.2. Environnement logiciel

3. Système d'exploitation : Ubuntu Mate 16.04 LTS 64-bits.
4. Outil de modélisation: StarUML version 2.8.0
5. Outils de développement: Qt Designer, Sublime Text

#### 3.1.3. Langages et outils de développement

Nous avons utilisé plusieurs langages et outils pour le développement de notre système. Nous avons utilisé ces langages et outils comme suit:

- Nous avons utilisé l'outil Qt Designer pour réaliser les interfaces graphiques de notre système.
- Nous avons utilisé le langage de programmation Python, utilisant l'outil Sublime Text et le framework de l'apprentissage profond Caffe, pour extraire les descripteurs.
- Nous avons utilisé l'outil med2image<sup>3</sup> et le langage de script shell (.sh), utilisant l'outil gedit pour convertir les fichiers Nifti en format jpg.

### 3.2. Les interfaces graphiques principales

Dans cette partie nous allons présenter les différentes parties de notre système, du côté interface graphique, en montrant les interfaces graphiques principales qui représentent quelques fonctionnalités du système

---

<sup>3</sup> : <https://github.com/FNNDSC/med2image>

- La figure 38 montre l'interface graphique de la conversion des fichiers Nifti en format JPG :

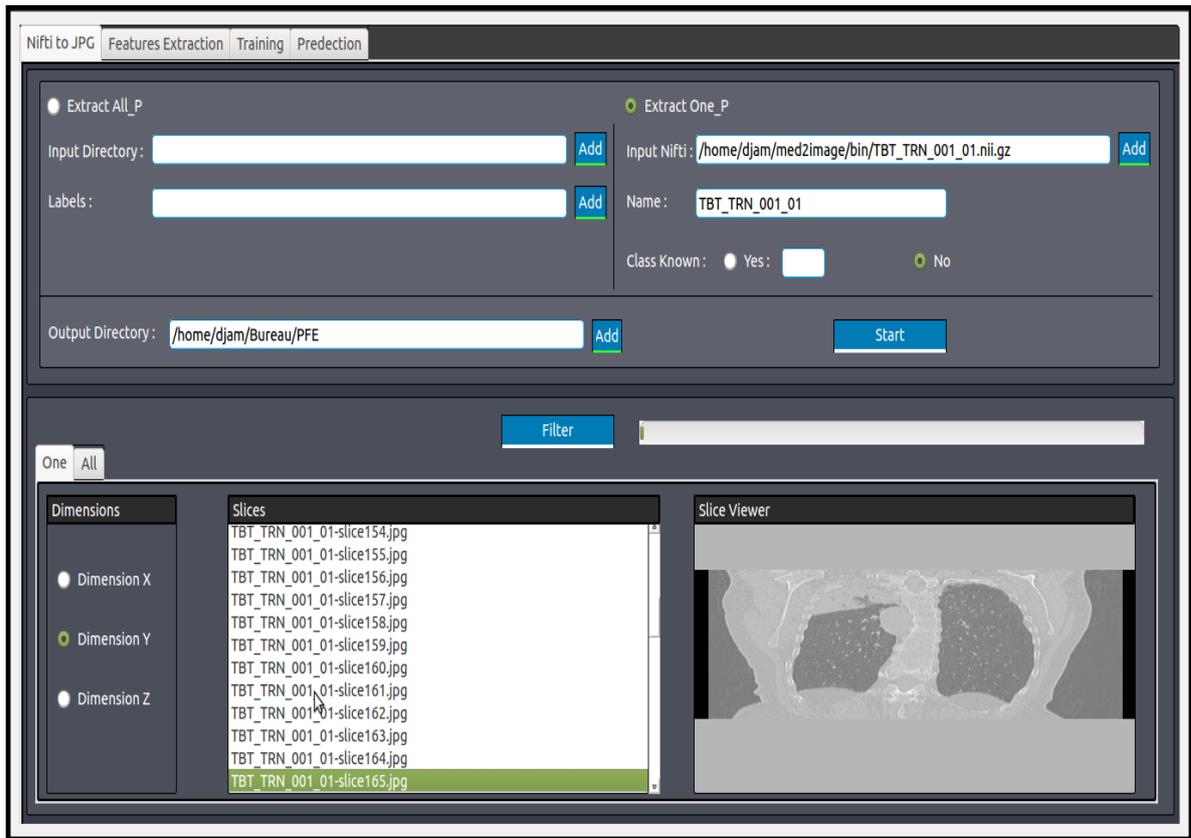


Figure 38 - Interface graphique de la conversion des fichiers Nifti en format JPG

Cette interface représente la partie de « prétraitement des données » de notre contribution. L'utilisateur a le choix d'extraire les slices d'une seule image Nifti ou bien d'un ensemble des images Nifti. La partie en bas est réservée à l'affichage des résultats de l'extraction (dimension, slices, l'affichage des images).

- La figure 39 montre l'interface graphique d'extraction des descripteurs sémantiques :

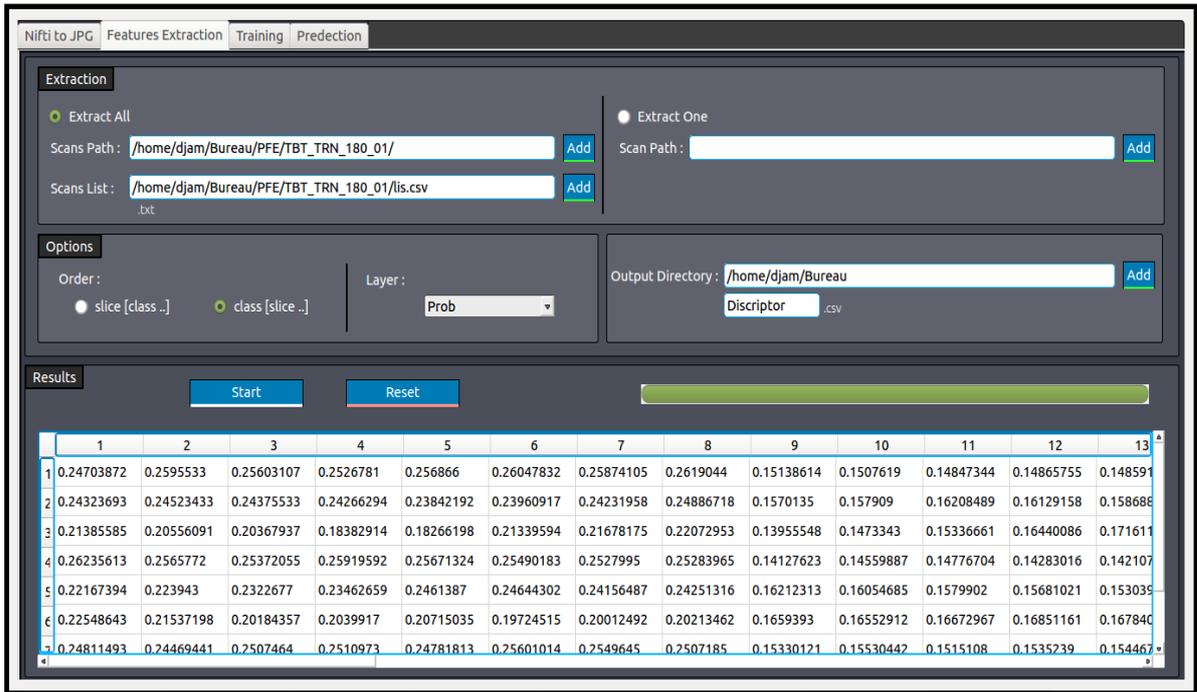


Figure 39 - Interface graphique d'extraction des descripteurs.

A partir de cette interface l'utilisateur peut extraire un descripteur sémantique pour un seul patient ou bien pour un ensemble de patients

- La figure 40 montre l'interface graphique de l'apprentissage :

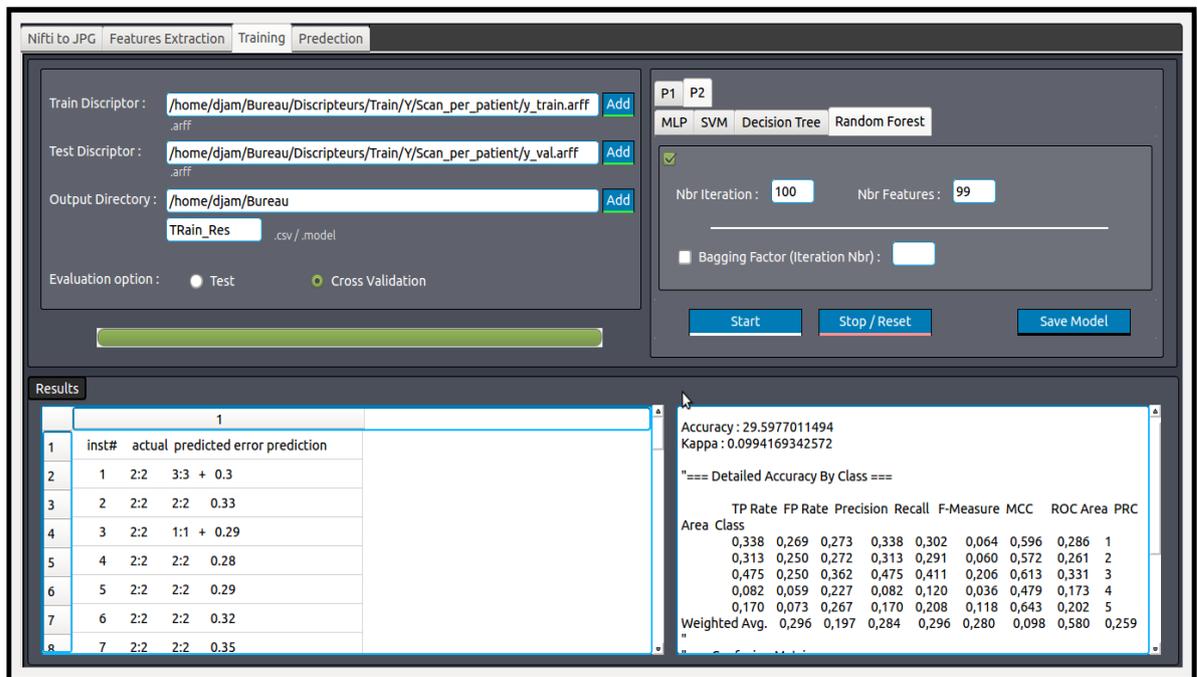


Figure 40 - Interface graphique d'extraction de l'apprentissage.

- La figure 41 montre l'interface graphique pour la prédiction :

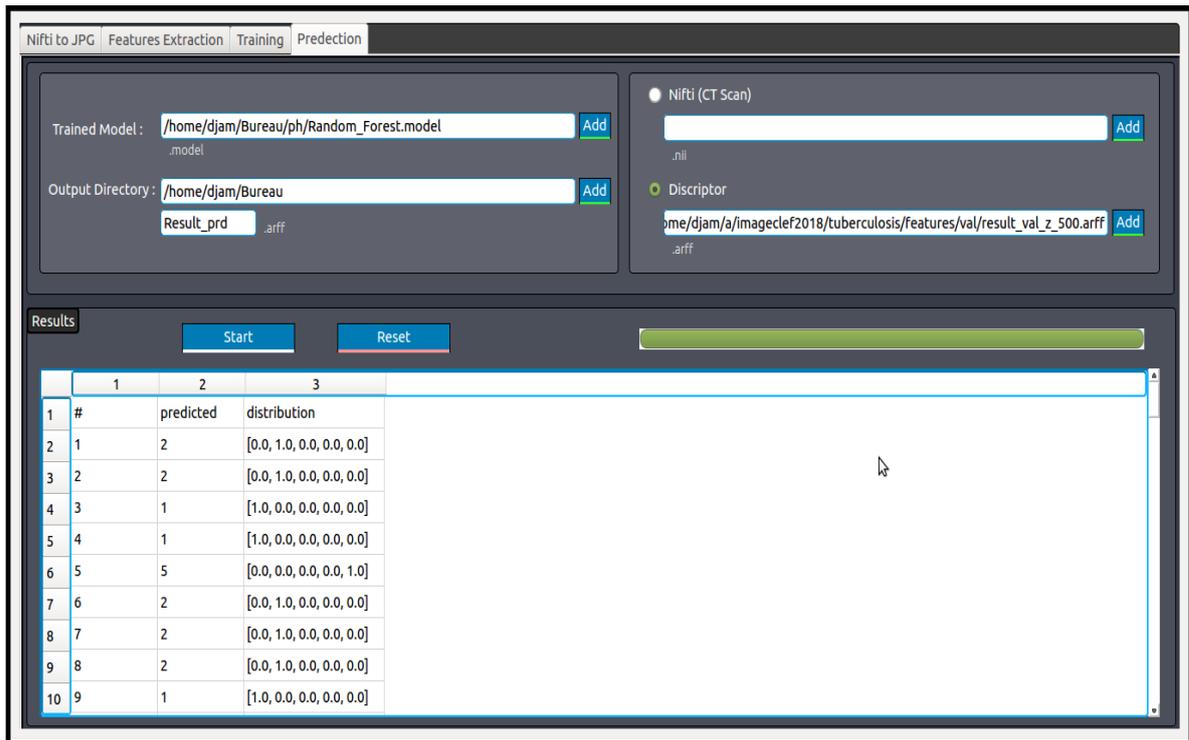


Figure 41 - Interface graphique pour la prédiction.

L'utilisateur peut utiliser les descripteurs sémantiques pour faire l'apprentissage (figure 40). Après il peut utiliser le modèle de classification qui a été généré dans la phase Apprentissage pour faire la prédiction (figure 41).

#### 4. Conclusion partielle

A travers ce chapitre nous avons pu présenter les différentes étapes de conception de notre système, en montrant les différents diagrammes UML utilisés. Nous avons présenté l'environnement de travail, et détaillé ensuite les différents langages et outils utilisés pour le développement de notre système. Nous avons terminé par une présentation des interfaces graphiques principales de fonctionnement du système. Dans ce qui suit nous allons présenter une conclusion générale.

## Conclusion générale

Nous avons présenté dans ce rapport la description d'un système de classification de types de tuberculose pulmonaire.

Dans un premier temps, nous avons mené une étude bibliographique sur la tuberculose pour avoir une idée claire sur cette maladie et les différences entre ses cinq types. Ensuite, nous avons survolé divers concepts en lien avec notre problématique, concernant l'imagerie médicale et l'apprentissage automatique. Nous avons présenté également les travaux qui ont été effectués dans le cadre du défi international *ImageCLEF* 2018 concernant la classification des types de tuberculose pulmonaire, et qui ont obtenu des résultats prometteurs.

Nous avons proposé comme contribution une approche qui consiste à extraire un seul descripteur sémantique pour chaque "image CT" / patient au lieu de considérer toutes les tranches (slices) comme des échantillons distincts. Nous avons participé pour tester notre approche au défi international *ImageCLEF* 2018 dans ses deux sous tâches: TBT et SVR. Nous avons atteint un Kappa de 0,0629 dans la tâche de TBT et notre meilleur résultat dans la tâche SVR a été classé 12ème sur 36 soumissions.

Nous notons que ces résultats ont été obtenus malgré les problèmes rencontrés dans la phase d'apprentissage de notre apprenant en profondeur qui n'a pu être achevée. Cependant, les résultats obtenus montrent que cette approche pourrait être beaucoup plus efficace et donner des résultats plus intéressants si elle est appliquée correctement.

En tant que perspectives, nous prévoyons d'adopter des stratégies d'enrichissement et de sélection des échantillons d'apprentissage. En effet, l'une des caractéristiques de la problématique abordée est la nature des ensembles de données fournis, qui sont de petite taille et bruités en raison de la présence de nombreux slices qui ne contiennent pas d'informations utiles. Nos stratégies de hiérarchisation et de sous-échantillonnage adoptées dans nos expérimentations le confirment. De plus, nous avons remarqué lors du sous-échantillonnage de nos données que la suppression ou l'ajout de certains échantillons avait un impact sur les résultats. D'autre part, le filtrage efficace des slices pour ne conserver que ceux qui sont vraiment informatifs est une idée clé qui pourrait améliorer encore les performances du système. De plus, nous avons remarqué dans nos expérimentations qu'il existe une différence de précision pour chaque classe étudiée. En effet, certaines classes sont plus difficiles à identifier que d'autres. C'est aussi une piste intéressante à étudier.

## Bibliographie

- [1] LO, S.-CB, LOU, S.-LA, LIN, Jyh-Shyan, *et al.* Artificial convolution neural network techniques and applications for lung nodule detection. *IEEE Transactions on Medical Imaging*, 1995, vol. 14, no 4, p. 711-718.
- [2] SHEN, Dinggang, WU, Guorong, et SUK, Heung-Il. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 2017, vol. 19, p. 221-248.
- [3] CID, Yashin Dicente, KALINOVSKY, Alexander, LIAUCHUK, Vitali, *et al.* Overview of the ImageCLEF 2017 Tuberculosis Task—Predicting Tuberculosis Type and Drug Resistances.
- [4] LAKHANI, Paras et SUNDARAM, Baskaran. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 2017, vol. 284, no 2, p. 574-582.
- [10] KRIZHEVSKY, Alex, SUTSKEVER, Ilya, et HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In : *Advances in neural information processing systems*. 2012. p. 1097-1105.
- [11] LECUN, Yann, BOTTOU, Léon, BENGIO, Yoshua, *et al.* Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, vol. 86, no 11, p. 2278-2324.
- [12] LAWRENCE, Steve, GILES, C. Lee, TSOI, Ah Chung, *et al.* Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 1997, vol. 8, no 1, p. 98-113.
- [13] WEI, Jun, CHAN, Heang-Ping, ZHOU, Chuan, *et al.* Computer-aided detection of breast masses: Four-view strategy for screening mammography. *Medical physics*, 2011, vol. 38, no 4, p. 1867-1876.
- [14] LAROBINA, Michele et MURINO, Loredana. Medical image file formats. *Journal of digital imaging*, 2014, vol. 27, no 2, p. 200-206.
- [15] RAZZAK, Muhammad Imran, NAZ, Saeeda, et ZAIB, Ahmad. Deep Learning for Medical Image Processing: Overview, Challenges and the Future. In : *Classification in BioApps*. Springer, Cham, 2018. p. 323-350.

- [16] GIBSON, Eli, LI, Wenqi, SUDRE, Carole, *et al.* NiftyNet: a deep-learning platform for medical imaging. *arXiv preprint arXiv:1709.03485*, 2017.
- [17] SUN, Jiamei, CHONG, Penny, TAN, Yi Xiang Marcus, *et al.* ImageCLEF 2017: ImageCLEF tuberculosis task-the SGEast submission. In : *CLEF2017 Working Notes. CEUR Workshop Proceedings, Dublin, Ireland, CEUR-WS. org* < <http://ceur-ws.org> > (September 11-14 2017).
- [18] STEFAN, L. D., DICENTE CID, Y., JIMENEZ-DEL-TORO, O., *et al.* Finding and classifying tuberculosis types for a targeted treatment: MedGIFT-UPB participation in ImageCLEF tuberculosis (2017). In : *CLEF2017 Working Notes. CEUR Workshop Proceedings, Dublin, Ireland, CEUR-WS. org* < <http://ceurws.org> > (September 11-14 2017).
- [19] WORLD HEALTH ORGANIZATION. *World health statistics 2015*. World Health Organization, 2015.
- [27] H. B. Barlow. Unsupervised Learning. *Neural Computation*, vol. 1, no. 3, pages 295-311, 1989.
- [28] O. Chapelle, B. Schölkopf and A. Zien. *Semi-supervised learnin. Adaptive computation and machine learning*. MIT Press, 2006
- [29] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001
- [30] B. E. Boser, I. Guyon, and V. Vapnik. A Training Algorithm for Optimal Margin Classifiers. *Proc. of the 5th ACM Conf. on Computational Learning Theory (COLT'92)*, pp. 144-152, Pittsburgh, PA, USA, July, 1992.
- [31] Hamadi, A., Mulhem, P., Quénot, G.: Extended conceptual feedback for semantic multimedia indexing. *Multimedia Tools Appl.*74(4), 1225–1248 (2015)
- [32] Dicente Cid, Y., Liauchuk, V., Kovalev, V., , Müller, H.: Overview of ImageCLEF tuberculosis 2018 detecting multi-drug resistance, classifying tuberculosis type and assessing severity score. In: *CLEF2018 Working Notes. CEUR Workshop Pro-ceedings*,

## Webographie

- [5] <http://respiratory.vsebolezni.com/tuberkulez/infiltrativnyj-tuberkulez-legkix-cto-eto-takoe.html>. - Consulté le 19/06/2018
- [6] <http://medicalformat.com/1194-focal-pulmonary-tuberculosis-causes-symptoms-diagnosis-and-treatment.html>. - Consulté le 19/06/2018
- [7] <https://fr.slideshare.net/princerockz99/pulmonary-tuberculoma>. - Consulté le 19/06/2018
- [8] <http://www.msmanuals.com/home/infections/tuberculosis-and-leprosy/miliary-tuberculosis-tb>. - Consulté le 19/06/2018
- [9] <http://respiratory.vsebolezni.com/tuberkulez/fibroznno-kavernoznyj-tuberkulez-legkix.html>. - Consulté le 19/06/2018
- [20] <http://www.physiotherapiepourtous.com/symptomes-de-la-tuberculose/>. - Consulté le 19/06/2018
- [22] <https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>. - Consulté le 19/06/2018
- [23] <http://www.shortscience.org/paper?bibtexKey=journals/corr/SzegedyLJSRAEVR14>. - Consulté le 19/06/2018
- [24] [https://medium.com/@siddharthdas\\_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5](https://medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5). - Consulté le 19/06/2018
- [25] [En ligne]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. - Consulté le 19/06/2018
- [26] [https://blogs.msdn.microsoft.com/big\\_data\\_france/2014/06/05/lapprentissage-automatique-machine-learning-comment-a-marche](https://blogs.msdn.microsoft.com/big_data_france/2014/06/05/lapprentissage-automatique-machine-learning-comment-a-marche) - Consulté le 19/06/2018

# Annexes

**Annexe 1** : ImageCLEF 2018: Semantic descriptors for Tuberculosis CT Image Classification, 2018

## ImageCLEF 2018: Semantic descriptors for Tuberculosis CT Image Classification

Abdelkader HAMADI<sup>[0000-0001-9990-332X]</sup> and Djamel Eddine YAGOUB

University of Abdelhamid Ibn Badis Mostaganem  
Faculty of Exact Sciences and Computer Science  
Mathematics and Computer Science Department  
Mostaganem, Algeria

abdelkader.hamadi@univ-mosta.dz  
djamel.ed.y@gmail.com

**Abstract.** In this article, we present our methodologies used in our participation at the two sub-tasks of the ImageCLEF 2018 Tuberculosis Task (TBT and SVR task). We proposed to extract a single semantic descriptor of 3D CT image to describe each patient rather than using all his slices as separate samples. In TBT task, the resulting descriptors are then exploited in a second learning stage to identify the type of tuberculosis among five given classes. In SVR task, the same experimental design is used to predict the degree of severity of the disease. We reached a Kappa coefficient value of about 0.0629 in TBT sub-task, and our best run on SVR was ranked 12<sup>th</sup> out of 36 submission and 5<sup>th</sup> out of 7 participant teams. We believe that our approach could give better results if applied properly.

**Keywords:** ImageCLEF · Tuberculosis Task · Deep Learning · CTImage · Tuberculosis CT Image Classification · Tuberculosis Severity Scoring.

### 1 Introduction

Tuberculosis is an infectious disease caused by a bacterium called *Bacillus microbacterium tuberculosis*. With a high mortality rate in the world, this disease remained one of the top ten causes of death in the world in 2015. Diagnosing this sickness quickly and accurately is a vital goal that would limit its invasion and damage. One of the major problems of this disease is that traditional tests produce inaccurate or too long results. For these reasons, researchers have been interested in this disease diagnosis, particularly in the context of the international challenge ImageCLEF 2017 [3] where two tasks have been reserved for it. The first aims to detect multi-drug resistant (MDR) status of patients. The goal of the second task is to identify the type of tuberculosis. A third task has been introduced in ImageCLEF 2018 [5] which consists to predict the degree of severity of the patient's case. In all the three tasks, the predictions are based

on 3D CT scans images. Algorithms involving deep learning have been tested to diagnose the presence or the absence of tuberculosis. The results obtained were interesting. However, they must be improved for better control and effective diagnosis, helping doctors to make the decisions and to choose the necessary treatments at the right time.

We can summarize the objectives of the Tuberculosis task through the following points:

- Helping medical doctors in the diagnosis of drug-resistant TB and TB type identification through image processing techniques;
- Introducing work towards inexpensive and quick methods for early detection of the MDR status and TB types in patients;
- Predicting quickly the type of TB and its severity degree to help doctors to make quick decisions and give the effective treatments.

We present in the following our work that has been made in the context of our participation to the two sub-tasks of ImageCLEF 2018 Tuberculosis Task: Tuberculosis Types classification (TBT) and Tuberculosis Severity Scoring (SVR).

The remainder of this article is organized as follows. Section 2 describes the two tasks to which we had participated. In section 3, we present our contribution by detailing the system deployed to complete our submissions. Section 4 details our experimental protocols followed to generate our predictions. We detail and analyze in the same section the results obtained. We conclude in the last section by presenting our perspectives and future works.

## 2 Participation to imageCLEF 2018

### 2.1 Tasks description

In this paper, we focus on our participation in the TBT and the SVR sub-tasks that we describe in the following sections.

In both tasks the data is provided as 3D CT scans. For some patients several 3D CT scans are given while for some others only one is provided. All the CT images are stored in NIFTI file format with .nii.gz extension file (g-zipped .nii files). For each of the 3-dimensions of the CT image, we find a number of slices varying from about 50 to 400. Each slice has a size of about  $512 \times 512$  pixels.

A training collection is provided at the beginning of the task with its ground-truth (labels of samples). Participants prepare and train their systems on this dataset. A test collection is provided at a later date. Participants interrogate their system and return their predictions to the organizers' committee. An evaluation is performed by the latter to compare the performance of the systems.

**TBT task** consists of the automatic categorization of TB cases in 5 target classes based on CT scans of patients. The five types considered are:

1. Infiltrative
2. Focal,
3. Tuberculoma
4. Miliary
5. Fibro-cavernous

The results will be evaluated using unweighted Cohens Kappa.

**SVR task** aims to predict the degree of severity of TB cases. Given a TB patient, the main goal is to predict its severity score based on his 3D CT scan. The degree of severity is modeled according to 5 discrete values : from 1 (“critical/very bad”) to 5 (“very good”). The score value is simplified so that values 1, 2 and 3 correspond to “high severity” class, and values 4 and 5 correspond to “low severity”.

The classification problem are evaluated using ROC-curves (AUC) produced from the probabilities provided by the participants. For the regression problem, the root mean square error (RMSE) is used.

### 3 Our contribution

We proposed to extract semantic descriptors from 3D CT scans. We noticed that participants of the ImageCLEF TBT 2017 task used each extracted slice as a separate sample. Thus, hundreds of slices are considered as separate learning samples while these slices represent the same patient. This introduces a lot of noise. In addition, each slice will be assigned the label of the patient (its type) even those whose content does not present any information to identify the type of TB case. This introduces more noise. The majority of the participants [10] of ImageCLEF 2017 highlighted this problem and its impact on the results.

To overcome this problem, we believe that the simplest solution is to produce a single descriptor for each patient. This constitutes the key idea of our contribution.

Our proposed system goes through three main stages:

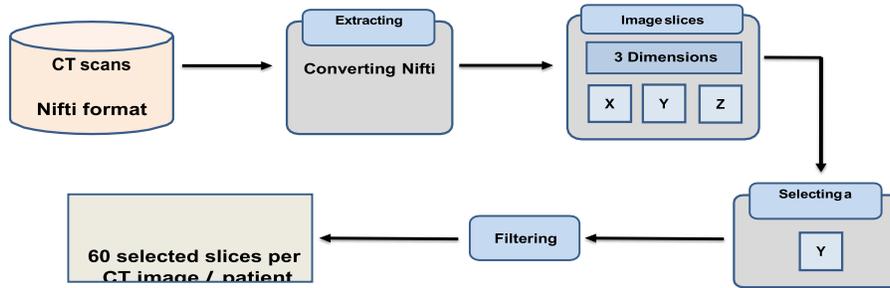
1. Input data pre-processing
2. Features extraction
3. Learning a classification model

We will detail each step in the following.

### 3.1 Input data pre-processing

We remind that in both tasks, 3D CT scans are provided in compressed Nifti format. Firstly, we decompress the files and extract the slices. At the end, we have three sets of slices corresponding to the three dimensions of the 3D image. For each dimension and for each Nifti image we obtain a number of slices ranging from 40 to 400 jpeg images.

The visual content of the images extracted from the different dimensions is not similar. Indeed, the images of each dimension are taken with from a different angle of view. We noticed from our experiments slices of the -Y- dimension give better results compared to the two others (X and Z). However, the following steps can be applied to slices of any of the three dimensions.



**Fig. 1.** Pre-processing of input data.

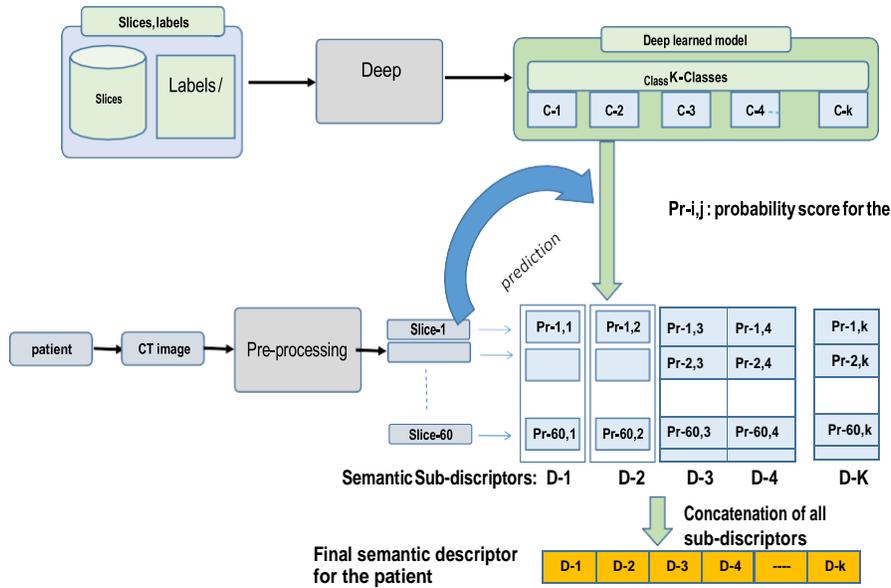
On the other hand, not all slices necessarily contain relevant information that can be useful to identify types of TB. This why, it is essential to filter slices by keeping only those that can be informative and may contain relevant information. Moreover, since we want to extract a single descriptor per patient, it is essential to keep the same number of slices for each patient. We found that there is usually a maximum of 60 slices visually informative. Since the slices are ordered, the 60 most informative are usually at the center of the list. We propose then to keep the 60 middle slices. This is not optimal but we opted for this choice for a fully automatic approach. This choice can be improved by performing a manual filtering with the intervention of a human expert, preferably with medical skills on TB disease. Figure 1 summarizes the process.

### 3.2 Features extraction

After slices extraction and filtering, we propose to extract a single descriptor per patient. The transfer learning presents in this context an interesting track that can be exploited. The results of SGEast [10] and even other teams in the same task of ImageCLEF 2017 prove its efficiency of this approach [4,10]. Indeed,

SGEast opted for the transfer learning where they exploited the output of a Resnet-50 [8] deep learner layer. However, this idea presents a problem of the resulting descriptor size. Indeed, for example, SGEast considered a descriptor per slice and not per patient. However, since we want to have a single descriptor, it is important that the information extracted from each slice must not be very large. Therefore, we propose to describe each slice by semantic information. This idea is inspired by the work presented in [7].

So, we choose to exploit the probabilities predicted by a deep learner trained on the set of slices. If  $K$  is the number of classes considered, this information typically corresponds to the  $K$  predicted probability values for the  $K$  classes (five probabilities of the five types for the TBT task, or the five severity degrees for the SVR task). We obtain then for each slice  $K$  values corresponding to the number of the considered classes.

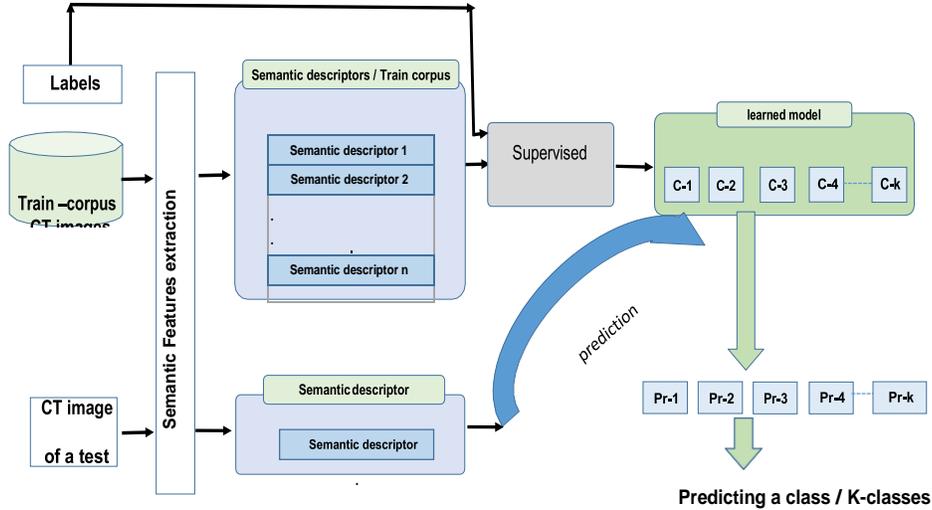


**Fig. 2.** Our semantic features extraction process.

Furthermore,  $K$  sub-descriptors are generated:  $D_1, D_2, D_3, D_4, \dots, D_k$ . Each sub-descriptor  $D_i$  contains the predicted probabilities for the class  $i$  for all the slices of the patient. A final semantic descriptor is constructed by concatenating the  $K$  sub-descriptors. Figure 2 details the process of the semantic feature extraction for one patient.

### 3.3 Learning a classification model

In this step, we propose to exploit the semantic descriptors of patients obtained in the previous step. Any approach of supervised classification can be applied as shown in figure 3.



**Fig. 3.** Learning a classification model based on the semantic descriptors.

We recommend for this step some ideas:

- To use a deep learner having as input the semantic descriptors of patients and their labels. As an alternative, we propose to use a bagging method that collaborates several learners and sub-samples the train collection. This would lead to better results as our experiments showed.
- To apply a samples selection, especially in the TBT task where several CT images were provided for some patients. We noticed in our experiments that using all the images for each patient introduces a lot of noise and would give less good results than using only one image per patient. An alternative consists of creating multiple sub-collections where each one contains a different single CT image per patient, and generating then a learner on each sub-collection to aggregate finally their results. This would probably lead to a much more robust model.

## 4 Experiments and results

We describe in the following sections our runs submitted to the TBT and SVR tasks.

We implemented the semantic descriptor approach described in section 3. We used for that the following tools:

- The Caffe framework [9] for deep learning;
- Weka [6] for testing several learning and classification algorithms;
- med2image [1] for the conversion of nifti medical images to the classic Jpeg format.

We chose to use slices of the -Y- dimension because our experiments showed that they are more suitable than those of the two others and got better results.

For descriptors extraction, our approach consists to learn a deep model to generate semantic information. Unfortunately, we had problems with our machines deployed for training our deep learner. Due to lack of time, we could not achieve the learning process. As an alternative to this step, we deployed the same model as the one proposed by the SGeast team [10] at the CLEF 2017 TBT Task. The model is accessible from the following link [2]. It is based on a Resnet-50 [8] and got the best results at the TBT task of 2017 edition. We have therefore exploited the outputs of the last layer (named *prob*) of the Resnet-50 corresponding to the probabilities of the 5 considered classes.

#### 4.1 TBT task

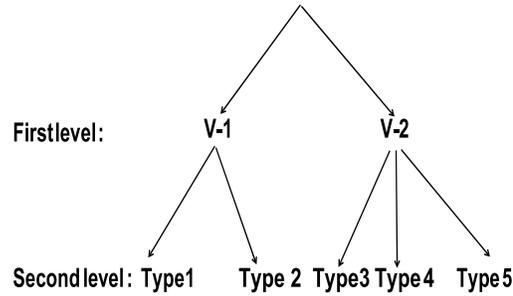
**Dataset:** The dataset used in TBT tasks includes chest CT scans of TB patients along with the TB type. Some patients include more than one scan. All scans belonging to the same patient present the same TB type. Table 1 summarizes the distribution of CT scans according to the five types of TB considered.

**Table 1.** Dataset given for Tuberculosis TBT task.

TB types	Train		Test	
	#Patients	#CTs	#Patients	#CTs
Type 1	228	376	89	176
Type 2	210	273	80	115
Type 3	100	154	60	86
Type 4	79	106	50	71
Type 5	60	99	38	57
Total	677	1008	317	505

**Experimental protocol:** We used the train collection provided by the organizers and we split it into two sub-collections: 80% for training and 20% as validation set. We have exploited in all our runs the semantic descriptors generated as previously described. We tested several learners in the classification step. We finally submitted three main runs. The other submissions are some variants or are generated through the fusion of some of these three runs:

- Run 1 (TBT mostaganemFSEI run1): random forest as supervised classifier. We tuned the two parameters referring to the number of iterations performed and the number of features selected randomly;
- Run 2 (TBT mostaganemFSEI run2): bagging of a set of random forest learners. We tuned the number of learners for the bagging and the same two parameters as Run1 for random forest;
- Run 4 (TBT mostaganemFSEI run4): A hierarchical classification. We organized the five 5 classes into a hierarchical structure as described in figure 4. We have created two new virtual classes  $V_1$  and  $V_2$ .  $V_1$  regroups the three classes Type 1, Type 2, and  $V_2$  contains the classes Type 3, Type 4 and Type 5. We have reorganized our collections in order to achieve a classification on two different levels. In the first stage, we classify the samples into two virtual classes  $V_1$  and  $V_2$ . In the second level of classification, we performed a classification of the samples regarding the set of classes of the predicted class in the previous stage. In two classification process we used a random forest learner by tuning its two parameters as described for Run1.



**Fig. 4.** Hierarchical re-organization of TBT types.

**Results:** Table 2 shows the results obtained by our runs on validation collection.

**Table 2.** Results on validation set for TBT task.

Runs	Kappa	Accuracy
Run 1 (TBT mostaganemFSEI run1)	0.21	0.38
Run 2 (TBT mostaganemFSEI run2)	0.25	0.41
Run 4 (TBT mostaganemFSEI run4)	0.26	0.52

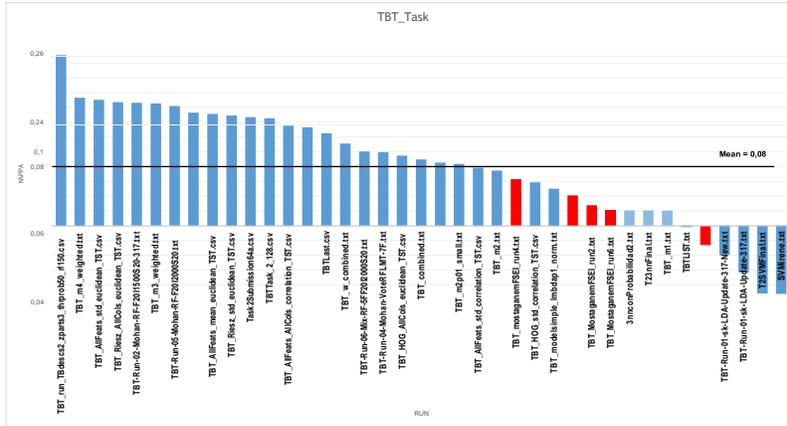
Table 3 shows the results obtained by our runs on the evaluation performed by the ImageCLEFcommittee.

**Table 3.** Results on test set for TBT task.

Runs	Kappa	Rank	Accuracy	Rank
Run 1 (TBT_mostaganemFSEI_run1)	0.0412	28	0.2650	29
Run 2 (TBT_mostaganemFSEI_run2)	0.0275	29	0.2555	32
Run 4 (TBT_mostaganemFSEI_run4)	0.0629	25	0.2744	27

As shown on validation results, Run 4 has been our best submission and got also the best results on test collection compared to run 1 and run 2.

Figures 5 and 6 describes the results and ranking of all submissions on TBT task in terms of kappa coefficient and accuracy, respectively.



**Fig. 5.** Results and ranking in terms of Kappa coefficient on test data for TBT Task.

Although the results achieved by our submissions are not very efficient compared to those of well ranked runs, we can notice several runs belong to the same teams that had good results. On the other hand, we recall that our semantic descriptors were extracted using a model that was not very well trained. In fact, we met problems with our machines during the training of our deep learner. Indeed, although SGEast’s deployed model got the best results at ImageCLEF 2017 Tuberculosis TBT task, we did not have the ability to perform exactly the same pre-processing performed by this team as described in [10]. We believe that our semantic descriptors could give better results if they are extracted from a more adapted and well-developed deeper model.

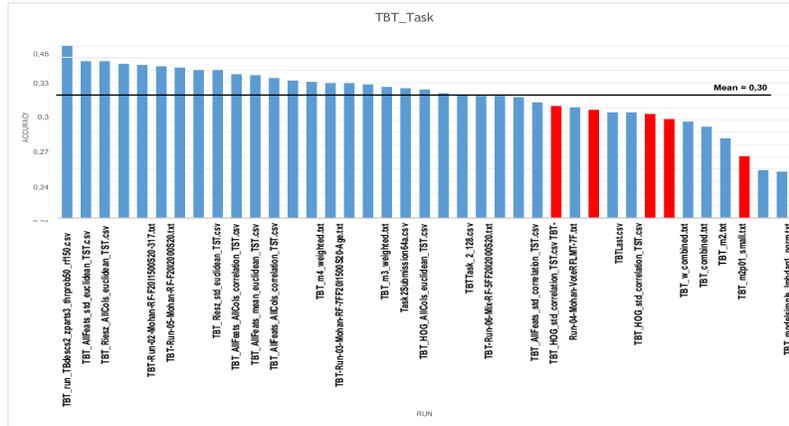


Fig. 6. Results and ranking in terms of accuracy on test data for TBT Task.

### 4.2 SVR task

**Dataset:** The dataset for SVR task includes chest CT scans of TB patients along with the corresponding severity score (1 to 5). Scores from 1 to 3 correspond to the “High” severity whereas the two scores 4 and 5 refer to the “Low” degree of severity. Table 4 summarizes the distribution of CT scans according to two severity classes.

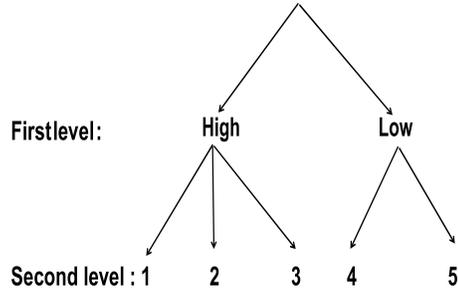
Table 4. Dataset given for Tuberculosis SVR task.

	Train	Test
Low severity	90	62
High severity	80	47
Total	170	109

**Experimental protocol:** We generated in a first step the semantic descriptors following the approach described in the section 3. For the prediction of TB severity scores, we treated the problem as a classification problem. We used for this two approaches :

1. Multi-class classification problem: we considered the five scores as separate classes. We then tested several classifiers. We selected two that have been most effective compared to those tested: Random forest, bagging of a set of random forest learners.
2. Hierarchical classification: We organized our data in order to carry out a hierarchical classification. We considered the hierarchy described in figure 7. Then, a two-level hierarchical classification is carried out. In the first level

the samples are classified into “High” or “Low” classes. In the second level, the samples are reclassified into the descending classes of the one predicted in the first level.



**Fig. 7.** The hierarchy of classes considered for SVR Task.

We submitted five runs:

1. Run 1 (SVR mostaganemFSEI run1): Multi-class model using Random forest as classifier. We tuned the two parameters : the number of iterations performed and the number of features randomly chosen;
2. Run 2 (SVR mostaganemFSEI run2) : Multi-class model using a bagging of a set of random forest learners with sub-sampling of the main train collection. We created two sub-collections by balancing the number of samples for the 5 classes. We then merged the results obtained by the two sub-collections;
3. Run 3 (SVR mostaganemFSEI run3): Hierarchical classification using a Bagging of a set of Random forest learners in each level of the hierarchical classification process.
4. Run 4 (SVR mostaganemFSEI \_run4): fusion of Run 1 and Run 2
5. Run 6 (SVR mostaganemFSEI \_run6): fusion of Run 3 and Run 1

**Results:** Table 5 shows the results obtained by our runs on validation collection.

**Table 5.** Results on validation set for SVR task in terms of Accuracy and Root Mean Square Error (RMSE).

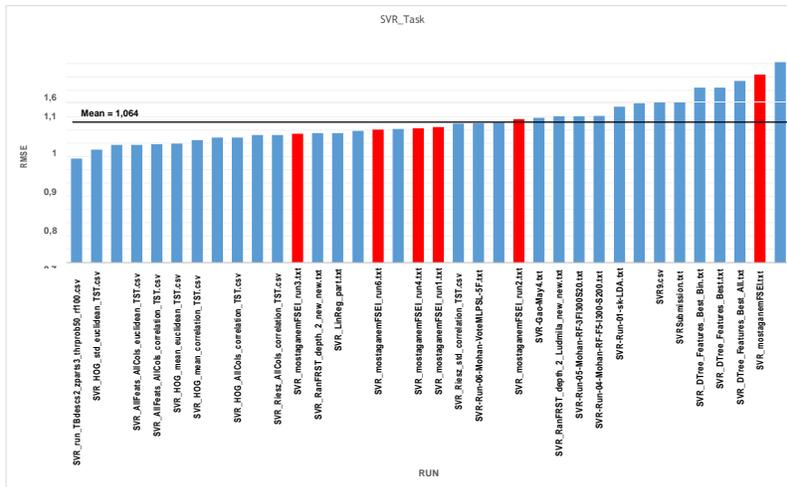
Runs	Accuracy	RMSE
Run 1 (SVR mostaganemFSEI run1)	0.41	0.37
Run 2 (SVR mostaganemFSEI run2)	0.36	0.45
Run 3 (SVR mostaganemFSEI run3)	0.56	0.3
Run 4 (SVR mostaganemFSEI run4)	0.42	0.36
Run 6 (SVR mostaganemFSEI run6)	0.48	0.34

Table 6 shows the results obtained by our runs on the evaluation performed by the ImageCLEF committee on test collection.

**Table 6.** Results on test set for SVR task.

Runs	RMSE	Rank	AUC	Rank
Run 1 (SVR_mostaganemFSEI_run1)	1.0227	19	0.5971	26
Run 2 (SVR_mostaganemFSEI_run2)	1.0837	22	0.6127	22
Run 3 (SVR_mostaganemFSEI_run3)	0.9721	12	0.5987	25
Run 4 (SVR_mostaganemFSEI_run4)	1.0137	18	0.6107	24
Run 6 (SVR_mostaganemFSEI_run6)	1.0046	16	0.6119	23

We can see that our run 3 got best results in terms of RMSE compared to our other runs on validation collection and even on test data. However, in terms of AUC, the run 2 seems to be more efficient.



**Fig. 8.** Results and ranking in terms of Root Mean Square Error on test collection.

Figures 8 and 9 describes the results and ranking of all submissions on SVR task in terms of RMSE and AUC values, respectively.

We can see that our best run is ranked 12<sup>th</sup> out of 36 submissions. However, the difference between the performances of the 12 best runs is not very significant. We recall that our best result is achieved by a hierarchical classification approach using a bagging of random forest learners at each level of the hierarchy. We believe that our approach could give better results using a well-trained deep model in the semantic features extraction step.

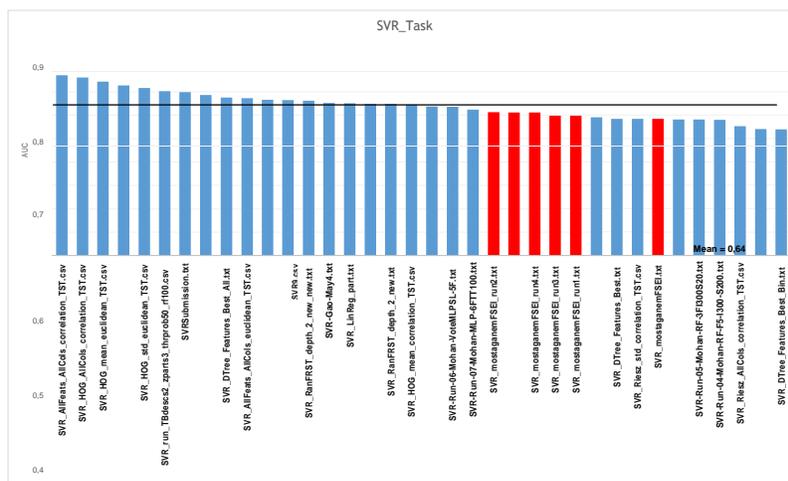


Fig. 9. Results and ranking in terms of Area Under ROC curve on test collection.

## 5 Conclusion and future works

We have described in this article our contributions to the TBT and SVR tasks of ImageCLEF Tuberculosis 2018. We proposed an approach that consists in extracting a single semantic descriptor for each CT image / patient instead of considering all the slices as separate samples. Unfortunately, we could not achieve the training of our deep learner. However, the results obtained show that this approach could be much more efficient and give more interesting results if it is applied properly.

As perspectives, we plan to adopt enrichment strategies and learning samples selection. Indeed, one of the characteristics of the problematic addressed in the SVR and TBT tasks is the nature of the provided data collections, which are of a small size and are noisy because of the presence of many slices that do not contain useful information. Our bagging and sub-sampling strategies adopted in our experiments confirmed this. In addition, we noticed during the sub-sampling of our data that the deletion or addition of some samples had an impact on the results. On the other hand, filtering slices effectively to keep only those that are truly informative is a key idea that could further improve system performance as reported by several participating teams [10]. Furthermore, we noticed in our experiments that there is a difference in terms of precision achieved for each studied class. Indeed, some classes are more difficult to identify than others. This is also an interesting track to study.

## References

1. med2image: <https://github.com/fnndsc/med2image>. Last check: 30/05/2018.
2. Sgeast model for imageclef 2017 tuberculosis task : [https://github.com/maizesix92/imageclef2017\\_tbsgeast](https://github.com/maizesix92/imageclef2017_tbsgeast). Last check: 30/05/2018.

3. Cid, Y.D., Kalinovsky, A., Liauchuk, V., Kovalev, V., Müller, H.: Overview of the imageclef 2017 tuberculosis task - predicting tuberculosis type and drug resistances. In: Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017. (2017), [http://ceur-ws.org/Vol-1866/invited paper 1.pdf](http://ceur-ws.org/Vol-1866/invited%20paper%201.pdf)
4. Dicente Cid, Y., Kalinovsky, A., Liauchuk, V., Kovalev, V., Müller, H.: Overview of ImageCLEFtuberculosis 2017 - predicting tuberculosis type and drug resistances. In: CLEF2017 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <<http://ceur-ws.org>>, Dublin, Ireland (September 11-14 2017)
5. Dicente Cid, Y., Liauchuk, V., Kovalev, V., Müller, H.: Overview of ImageCLEFtuberculosis 2018 - detecting multi-drug resistance, classifying tuberculosis type, and assessing severity score. In: CLEF2018 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <<http://ceur-ws.org>>, Avignon, France (September 10-14 2018)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *SIGKDD Explorations* **11**(1), 10–18 (2009)
7. Hamadi, A., Mulhem, P., Quénot, G.: Extended conceptual feedback for semantic multimedia indexing. *Multimedia Tools Appl.* **74**(4), 1225–1248 (2015). <https://doi.org/10.1007/s11042-014-1937-y>, <https://doi.org/10.1007/s11042-014-1937-y>
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
9. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
10. Sun, J., Chong, P., Tan, Y.X.M., Binder, A.: Imageclef 2017: Imageclef tuberculosis task - the sgeast submission. In: Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017. (2017), [http://ceur-ws.org/Vol-1866/paper 130.pdf](http://ceur-ws.org/Vol-1866/paper%20130.pdf)

## Résumé

La tuberculose (TB) est une maladie infectieuse qui demeure une cause majeure de décès dans le monde. L'organisation mondiale de la santé (OMS) estime qu'il y a eu 10,4 millions de nouveaux cas de tuberculose dans le monde en 2015. La majorité des populations infectées proviennent de communautés pauvres et marginalisées avec une infrastructure de soins de santé médiocre. De nombreux patients tuberculeux guérissables meurent en raison d'un diagnostic tardif, en partie à cause des coûts excessifs des analyses médicales et du manque de radiographie et de radiologie. Par conséquent, le développement d'un système de diagnostic assisté par ordinateur pour le dépistage de la tuberculose s'avère utile pour un diagnostic précoce de cas tuberculeux, ce qui permet de prévenir les décès dus à cette maladie. D'autre part, l'apprentissage en profondeur s'impose de plus en plus ces dernières années dans le domaine de l'analyse et la classification des images, et confirme à travers ses bons résultats dans plusieurs problématiques sa suprématie dans cet axe de recherche. En effet, il a été vérifié que les réseaux de neurones convolutifs profonds (DCNN) sont des algorithmes performants pour résoudre diverses tâches dans les domaines de la vision par ordinateur et de l'analyse d'images. Cette performance se concrétise également dans l'analyse d'images médicales. Nous nous intéressons dans ce travail à la détection automatique des types de TB via une analyse d'images par tomodensitométrie, en exploitant les avantages des réseaux de neurones convolutifs. Une application réelle d'une solution à cette problématique pourrait être l'accélération du diagnostic de la maladie à partir d'une image de radiologie sans avoir recours à des analyses médicales coûteuses. Vu la qualité des images de radiologie, cette étude s'avère beaucoup plus difficile qu'une analyse d'images ordinaires. Dans ce travail, nous proposons une approche générant une description sémantique des images de scanner 3D des patients tuberculeux. Ces descripteurs sont ensuite utilisés dans une étape d'apprentissage pour apprendre un modèle de classification. Nos différentes expérimentations sont réalisées dans le contexte des deux sous-tâches de la tâche Tuberculosis de ImageCLEF 2018: Classification des types de tuberculose (TBT) et Prédiction du score de sévérité (gravité) du cas tuberculeux (SVR).

**Mots-clés :** Tuberculose, Apprentissage automatique, Apprentissage profond, DCNN, ImageClef.