université PARIS-SACLAY
IUT D'ORSAY

Département Informatique

Diplôme préparé : Informatique

N° Jury 3

Développement d'un logiciel d'aide au traitement de données expérimentales en utilisant l'intelligence artificielle

Mathieu LACROIX TP4A2

Tuteur enseignant

Xavier LACOUR

Responsable en entreprise

Gaston EXIL

Année universitaire 2021-2022

Période de stage 19/04/2022 – 01/07/2022

Rapport remis le 17/06/2022

Table des matières

Remerciements

Je tiens à remercier mon maître de stage, M. Gaston EXIL, responsable du Service Informatique du laboratoire Léon Brillouin, qui m'a accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie.

Je tiens également à remercier M. Éric ELIOT, Directeur, et M. Grégory CHABOUSSANT, Directeur Adjoint de m'avoir donné l'opportunité d'intégrer le laboratoire.

Je tiens aussi à remercier Aurore VERDIER pour toute l'aide qu'elle m'a apporté afin que je puisse réaliser ce stage dans les meilleures conditions.

Enfin, je remercie les collaborateurs du pôle administratif et financier du laboratoire pour leur disponibilité et les conseils qu'ils ont pu me donner au cours de ces deux mois.

Introduction

Du 19 avril 2022 au 1er juillet 2022, j'ai effectué mon stage de fin d'études au Commissariat à l'Energie Atomique et aux Energies Alternatives sur le site de Saclay (Essonne – France) au sein du laboratoire Léon Brillouin.

Au cours de ce stage au sein du Service Informatique, je me suis intéressé au développement de logiciels d'aide au traitement de données expérimentales en utilisant l'intelligence artificielle.

Ce stage a été l'occasion pour moi de mettre en application mes compétences de développeur acquises durant mes deux années d'études. Au-delà de l'enrichissement de mes connaissances dans le développement informatique, cette expérience m'a permis de découvrir les missions des développeurs dans le monde professionnel.

Problématique et objectifs de stage

Ce stage a été pour moi une opportunité enrichissante de percevoir comment fonctionne une entreprise dans le secteur de la recherche.

Pour élaborer ce mémoire, j'ai puisé dans les nombreux enseignements issus des missions qui m'ont été confiées. De plus, les différents échanges que j'ai pu mener avec les collaborateurs du laboratoire et plus particulièrement de son service informatique sont autant d'enrichissements qui ont permis de rendre ce rapport cohérent.

Présentation de l’entreprise

LE CEA

Le Commissariat à l’Énergie Atomique et aux Énergies Alternatives est un organisme public de recherche à caractère scientifique, technique et industriel. Il est classé comme Établissement Public à caractère Industriel et Commercial (EPIC).

Acteur majeur de la recherche, du développement et de l'innovation, le CEA intervient dans quatre domaines :

- la défense et la sécurité,
- les énergies basses carbones (nucléaires et renouvelables),
- la recherche technologique pour l'industrie,
- la recherche fondamentale (sciences de la matière et sciences de la vie).

Fort de son expertise reconnue, le CEA participe, depuis près de 80 ans, à la mise en place de projets avec de nombreux partenaires et conduit, pour le compte de l’Etat, des programmes de recherche visant à accroître la connaissance scientifique et à contribuer à l’innovation dans de nombreux domaines.

En 2010, le CEA, historiquement le Commissariat à l’Énergie Atomique, a vu sa mission évoluer et a changé de nom pour devenir le Commissariat à l’Énergie Atomique et aux Énergies Alternatives.

Le CEA dénombre 20 181 salariés dont 1 233 doctorants et 176 post-doctorants. Son budget annuel atteint 5 milliards d’euro.

Ses travaux lui ont permis de déposer de très nombreux brevets. Ainsi selon le classement annuel des déposants de brevets de l’Institut National de la Propriété Industrielle (INPI), en 2021, le CEA est :

- le 1er organisme de recherche français avec 528 dépôts de brevets,
- le 2ème plus important dépositaire de brevets derrière le groupe Safran et devant les groupes Valeo et Saint-Gobain.

Le CEA, c’est également :

- le seul organisme de recherche français figurant dans le classement des « 100 premiers innovateurs mondiaux » publié par Clarivate Analytics et ce depuis 10 années,
- plus de 5 000 publications annuelles,
- le 1er organisme de recherche public au classement mondial des dépositaires selon le traité de coopération en matière de brevets (PCT) de l’Organisation Mondiale de la Protection Intellectuelle (OMPI).

LE CNRS

Le Centre National de la Recherche Scientifique (CNRS) est une institution de recherche publique française créée par décret en Octobre 1939 et dont la mission principale est de « coordonner l'activité des laboratoires en vue de tirer un rendement plus élevé de la recherche scientifique ».

Le rôle du CNRS est de faire progresser la connaissance et d'être utile à la société en utilisant la recherche. Sa mission est divisée en plusieurs axes :

- la recherche scientifique,
- la valorisation des résultats de recherche,
- le partage des connaissances,
- la formation par la recherche,
- la contribution à la politique scientifique.

Pour son activité scientifique, le CNRS s'appuie sur 10 instituts nationaux spécialisés dans :

- les sciences humaines et sociales,
- la biologie,
- la chimie,
- l'écologie et l'environnement,
- les sciences de l'information,
- les sciences de l'ingénierie et des systèmes,
- les mathématiques,
- la physique nucléaire et des particules,
- les sciences de l'univers.

Le CNRS est reconnu aussi bien au niveau national qu'international pour la qualité de ses travaux scientifiques tant dans l'univers de la recherche et du développement que pour le grand public.

LE LABORATOIRE LEON BRILLOUIN

Le Laboratoire Léon Brillouin (LLB) est une unité mixte de recherche, qui a pour tutelles le Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA) et le Centre National de la Recherche Scientifique (CNRS). Son objectif est de mener des recherches sur la structure et la dynamique de la matière condensée en utilisant les faisceaux de neutrons fournis par le réacteur Orphée. Il est situé dans le centre de recherche CEA/Saclay

L'utilisation des faisceaux de neutrons est une technique récente indispensable pour les recherches au niveau microscopique dans des domaines très variés tels que la physique, la chimie, la biologie, la science des matériaux, le magnétisme… Au niveau industriel, cette technique permet de réaliser des études non-destructives.

Le LLB dispose de son propre programme de recherche scientifique et collabore avec des scientifiques de nombreux laboratoires de recherche fondamentale, de recherche appliquée et de l'industrie. Le LLB produit chaque année près de 180 publications dans des revues scientifiques et plus de 150 communications dans des conférences et réunions.

Les missions du laboratoire sont :

- Conduire une recherche de haut niveau sur ses propres programmes en s'appuyant sur ses installations sur le site de Saclay, de Grenoble ou d'autre sites
- Soutenir la formation à la diffusion neutronique et préparer une nouvelle génération d'utilisateurs aux méthodes les plus récentes de diffusion des neutrons
- Promouvoir l'utilisation de la diffusion et de la spectroscopie neutronique pour la recherche fondamentale et l'industrie et mettre en place des collaborations et/ou des partenariats avec des sociétés industrielles et commerciales dans le cadre de l'accès aux faisceaux de neutrons du LLB
- Assurer après l'arrêt du réacteur Orphée, le démontage, le recyclage et l'organisation de l'optimisation de la réutilisation au mieux des intérêts de la

communauté scientifique française de tout ou partie des instruments sur d'autres sources de neutrons en France et à l'étranger le cas échéant

- Mener une prospective pour une stratégie à long terme visant à garantir l'excellence française en diffusion des neutrons et proposer dans ce cadre, le développement d'une source compacte de neutrons destinée à redonner à la France au moins une source nationale
- Assurer la coordination opérationnelle de la contribution française pour la construction et la mise en service d'instruments sur la nouvelle source européenne de neutrons ESS (European Spallation Source, Lund, Suède) dans le cadre des accords entre la France et l'ERIC ESS

ESS :

L'**ESS** (pour **European Spallation Source** en anglais, soit en français source européenne de spallation) est le nom d'une future installation de recherche scientifique sur la matière utilisant des techniques de dispersion des neutrons. La construction à Lund, en Suède, a commencé le 30 juin 2014, avec des prévisions de mise en service en 2025 et une installation entièrement opérationnelle en 2028. Le Comité directeur de l'ESS rassemble 16 pays partenaires et devrait avoir un cout final de plus de 2 milliards d'euros.

La future installation est composée d'un accélérateur linéaire dans lequel des protons sont accélérés et projetés sur une cible en tungstène. Ces deux sous-ensembles constituent la source de neutrons thermiques et froids.

L'ESS sera dix fois plus puissante que les installations américaines et japonaises actuelles et fournira à ses utilisateurs des expériences cent fois meilleures que les sources à neutrons actuelles.

Mission

Mise en situation du projet

Comme précédemment indiqué le service informatique du LLB et les chercheurs collaborent dans le cadre de leurs missions de recherche.

La mission qui m'a été confiée, a pour objectif d'apporter une aide aux chercheurs par le développement de logiciels de traitement de données.

Ces logiciels doivent permettre :

- de représenter graphiquement des données d'acquisition sur une application web afin de faciliter l'interprétation des résultats (1ère partie du rapport),
- d'indiquer la structure d'un échantillon à partir de la représentation graphique des données d'acquisition en s'appuyant sur la reconnaissance d'images par apprentissage automatique (2ème partie du rapport).

En effet, les chercheurs utilisent les neutrons afin d'analyser la composition atomique d'un échantillon et d'en déterminer ses propriétés. L'environnement de test de l'échantillon permet de modifier les paramètres de température, pression, de champs magnétiques et de positionnement de l'échantillon.

Les faisceaux de neutrons traversent l'échantillon et sont réceptionnés sur un détecteur qui collecte les données précédemment citées. Elles sont ensuite stockées dans un fichier nommé « fichier .32 » sur un modèle spécifiquement développé par le CEA.

Ce fichier contient toutes les informations relatives aux caractéristiques de l'échantillon, les paramètres de la manipulation et les résultats de l'expérimentation.

Pour mener à bien la mission confiée, il m'a fallu tout d'abord analyser la structure des données contenues dans le fichier .32, puis concevoir une base de données pour sauvegarder les fichiers, et développer avant de mettre en œuvre des modalités de classification.

Pour ce faire, nous nous sommes orientés vers un domaine de l'intelligence artificiel : l'apprentissage automatique ou machine Learning en anglais. Nous avons retenu plus particulièrement, la reconnaissance d’images sur les graphiques obtenus. Elle permet :

- de reconnaître les paramètres et l’échantillon utilisé lors de l’expérimentation,
- de trier les fichiers et de récupérer les plus significatifs.

Ainsi les chercheurs disposent d’un outil pour extraire les fichiers sur lesquels sont stockées les données utiles et peuvent gagner en temps de traitement considérant le volume conséquent de données collectées.

Phase conceptuelle du projet

Tout projet de développement débute par une étape de cadrage qui permet d'anticiper autant que de possible comment va se dérouler le projet en intégrant les objectifs attendus, les ressources nécessaires et les contraintes diverses.

Fonctionnalités

Le cahier des charges relatif au projet prévoyait les fonctionnalités suivantes :

- Stockage sur une base de données,
- Visualisation des données issues des acquisitions neutroniques,
- Création d'une application web interactive,
- Extraction des données depuis les fichiers .32,
- Test des modèles d'apprentissage automatique.

Contraintes

Le cahier des charges relatif au projet prévoyait également les obligations suivantes :

- Utilisation de Framework Streamlit pour l'interface utilisateur,
- Utilisation de Framework Plotly pour la représentation graphique,
- Développement d'une interface simple d'utilisation pour les chercheurs,
- Utilisation du langage de programmation Python.
- Utilisation de la gestion de versions décentralisées du code avec Git.

Déroulement du projet

Sous la supervision de M. EXIL, j'ai mené le projet en autonomie, en suivant les consignes et en appliquant une méthodologie dite agile.

J'ai divisé le projet en sous–projets au sein desquels j'ai défini 3 phases :

- une phase de recherche,
- une phase de développement
- une phase de test avec réajustements si nécessaires.

L'issue de chaque phase était considérée comme un jalon permettant de valider les travaux réalisés et d'engager les suivants.

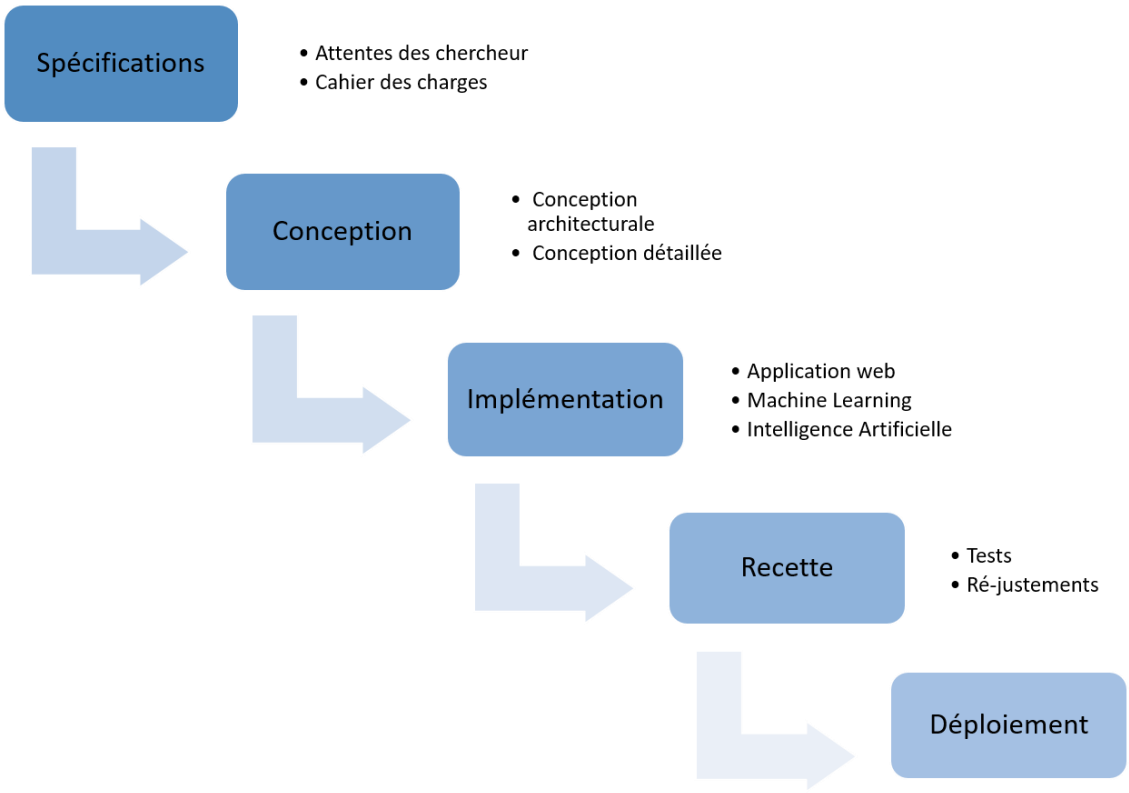


Figure 1 : Modèle de développement

PARTIE 1 - Conception de l’application web

Structure des fichiers .32

Un fichier .32 est le livrable obtenu à la suite du bombardement d’un échantillon par un faisceau neutronique. C’est l’ensemble des données collectées (caractéristiques de l’échantillon, paramètres d’acquisition et résultats de la manipulation).

La première étape a consisté à prendre connaissance et à comprendre la structure d’un tel fichier.

Un fichier .32 est composé de 3 blocs :

- bloc 1 : taille fixe de 256 octets, contient le détail des paramètres d’acquisition programmés et réels, codés à l’aide de caractères ASCII,
- bloc 2 : les valeurs d'intensité acquises par un détecteur de résolution 128 x 128. Celles-ci sont représentées dans un tableau d’une seule dimension, en binaire sous le format d'entiers non signés codés sur 4 octets,
- bloc 3 : le détail de paramètres d'acquisition complémentaires codés à l’aide de caractères ASCII.

Après avoir compris la structure du fichier .32, l’enjeu a été de trouver la solution pour extraire et stocker les données afin de pouvoir les traiter.

Stockage des fichiers

Afin de rendre plus aisée la manipulation des fichiers et l'affichage des données sur l'application web, les données utiles doivent être stockées sur une base de données mise en place grâce à l'application DB Browser sous SQLite.

Celle-ci intègre 2 tables liées :

- La table « Fichier » pour stocker la valeur de chaque caractéristique des fichiers (ID, nom, Checksum, …) et les données collectées lors de la manipulation,
- La table « Checksum » qui va servir à stocker le Checksum de chaque fichier.

Etant donné la limite du nombre de caractères par valeur imposée par SQLlite, les valeurs d'acquisition enregistrées dans le fichier .32 sont converties en hexadécimal afin d'être intégrées à la base de données.

Les valeurs associées aux paramètres étant plus réduites en nombre de caractères, il n'est pas nécessaire de convertir celles-ci.

Figure 2 : schéma de la base de données

Interface de l'application Web

L'application web doit répondre à 2 caractéristiques importantes :

- permettre une utilisation simple pour les chercheurs,
- fournir des résultats précis leur permettant de faire les analyses prévues.

Pour ce faire, l'interface a été conçue avec un maximum d'ergonomie en rassemblant l'ensemble des fonctionnalités indispensables pour les chercheurs.

Le langage Python et l'API Streamlit ont été utilisés pour :

- la conception de l'application web,
- l'association avec des widgets tels que celui utilisé pour charger les fichiers .32,
- l'association avec d'autres bibliothèques Python (Matplotly, Pandas…) permettant par exemple d'afficher les données sous forme de graphiques ou de tableaux.

Pour utiliser l'application, les chercheurs doivent en premier lieu charger un fichier .32. Pour cela, l'application utilise le widget st.file_uploader via l'API Streamlit.

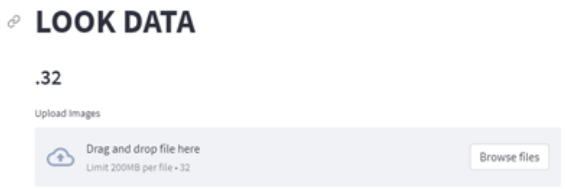


Figure 3 : widget pour charger un fichier .32 dans l'application

Le fichier chargé est ensuite enregistré dans la base de données dédiée. J'ai ensuite développé des requêtes SQL sous python pour extraire des données telles que par exemple :

- le nom du fichier
- le chemin du fichier
- le type du fichier
- les paramètres configurés pour la manipulation

- le tableau des valeurs de l'intensité
- le bloc descriptif des données d'acquisition.

Ci-dessous, deux exemples de requête développés pour :

- extraire le nom du fichier en fonction d'un checksum donné :

```
"SELECT Nom_Fichier From Fichier Where CheckSum_Fichier = '%s'"%id
```

- insérer les données du fichier .32 dans la base de données :

```
"INSERT INTO Fichier VALUES(?)", (item)
```

« Item » correspond aux typologies de données à insérer.

Les résultats des requêtes sont différemment traités en fonction de leur type, à savoir :

- les paramètres de configuration pour réaliser les manipulations,
- les résultats des manipulations.

Concernant les paramètres de configuration, ceux-ci sont affichés dans un tableau à 2 colonnes :

- le nom du paramètre,
- la valeur du paramètre.

Concernant les valeurs des intensités acquises, celles-ci sont converties d'un format hexadécimal (Cf. § Stockage des fichiers) en valeur entière numérique, puis affichées dans un tableau à 2 colonnes :

- l'index du point d'acquisition,
- la valeur mesurée au point d'acquisition.

L'index de chaque point d'acquisition correspond au couple de ses coordonnées cartésiennes « abscisse ; ordonnée ».

L'abscisse et l'ordonnée sont comprises entre 0 et 127. Il existe donc 16 384 couples de coordonnées qui sont convertis en nombre unique.

tableau des paramètres

	nom	valeur
0	basename	XY056
1	title	MnGe-
2	sample	MnGe-D20
3	counting	00030
4	monitor	0050866
5	date	22/03/201
6	time	15:41:3
7	max	0005
8	selectorSpeed	2202
9	lambda	0

tableau des intensitées

	intensité
2791	8
2792	8
2793	9
2794	3
2795	8
2796	12
2797	5
2798	3
2799	5

Figure 5 : tableaux des données extraites du fichier .32

Afin de faciliter la lecture des résultats, les valeurs d'intensité acquises sont représentées sous différentes formes graphiques créés grâce à la bibliothèque Plotly.

La représentation graphique ci-dessous est un graphique en 2D avec une vue par le dessus.

Pour réaliser cette représentation graphique, le tableau d'intensité acquise a été remodelé. Pour cela, chaque nombre unique relatif à un couple de données est re-converti en couple de coordonnées X et Y distinctes.

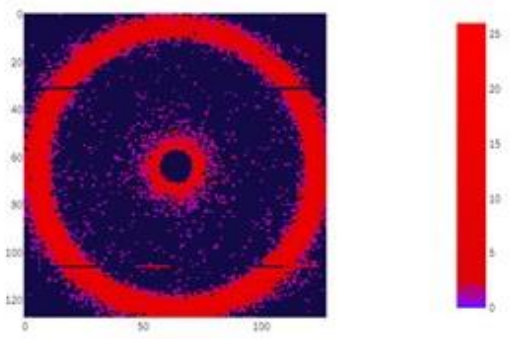


Figure 6 : Graphique en 2D

La représentation graphique ci-dessous est une vue en 3D du graphique précédent auquel l'intensité mesurée a été ajoutée en tant que 3ème axe (Z).

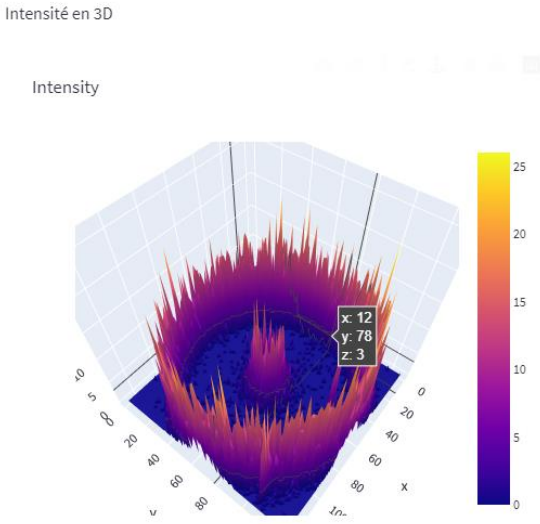


Figure 7 : Graphique en 3D

La représentation graphique ci-dessous est une vue de côté en 2D, du graphique précédent coupé à la verticale.

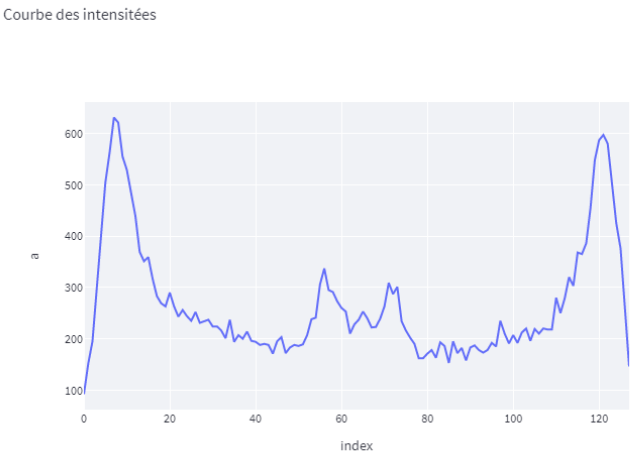


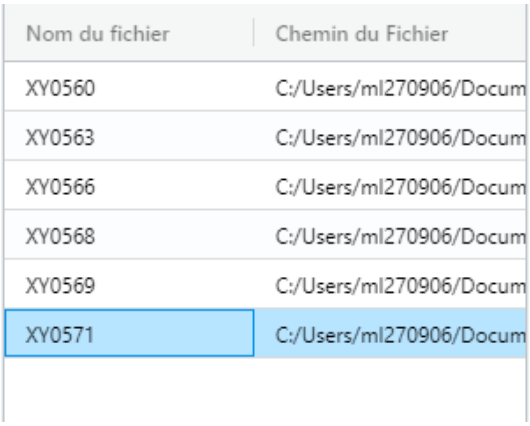
Figure 8 : graphique en 2D (coupe verticale)

En complément, une fonctionnalité supplémentaire a implémenté afin de pouvoir charger des dossiers de fichiers .32. Cette fonctionnalité permet d'alléger l'étape de chargement, notamment si les chercheurs disposent de nombreux fichiers.

Dans ce cas, face à un nombre important de fichiers et afin de permettre un affichage sans surcharger la page, AG Grid (Agnostic Grid) a été utilisée en tant qu'outil de visualisation de base de données permettant d'afficher les fichiers .32 comme une bibliothèque.

Ainsi en cliquant sur le nom d'un fichier dans la bibliothèque, les représentations graphiques et les tableaux sont actualisés à partir des données portées par le dit-fichier.

Les modalités de fonctionnement précédemment décrites et applicables à un fichier unique restent applicables.



Nom du fichier	Chemin du Fichier
XY0560	C:/Users/ml270906/Docum
XY0563	C:/Users/ml270906/Docum
XY0566	C:/Users/ml270906/Docum
XY0568	C:/Users/ml270906/Docum
XY0569	C:/Users/ml270906/Docum
XY0571	C:/Users/ml270906/Docum

Figure 9 : liste des fichiers présents dans la base de données

Conclusion

La mise en place de cette application web de représentation graphique des données d'expérimentation m'a permis de découvrir et d'utiliser un langage de programmation (Python) qui était nouveau pour moi et d'expérimenter de nouveaux outils (librairies associées et API).

Grâce à cette application, les chercheurs vont dorénavant disposer de l'ensemble des outils nécessaires pour mener à bien leurs analyses. Bien qu'étant fonctionnelle à l'issue de mon stage après avoir réussi les tests sur des données de recette, l'application devra être mise sous tension en situation réelle. L'application a encore beaucoup de potentiel et devrait être amenée à évoluer en fonction des attentes des chercheurs.

PARTIE 2 - Conception du logiciel de traitement des données

Cette seconde partie du rapport va être consacrée à l'Intelligence Artificiel utilisée dans le cadre de ce projet.

La quantité de données collectées durant une seule acquisition peut être de l'ordre du millier. Face au volume de plus en plus important de données collectées grâce à l'évolution des technologies neutroniques, il arrive que certaines d'entre elles soient aberrantes pour différents facteurs. Il en résulte des représentations graphiques inexploitables.

Aujourd'hui, le tri des images d'acquisition est fait manuellement par les chercheurs. Cela représente un travail long, fastidieux et répétitif.

L'objectif de cette seconde partie de mon stage a été d'utiliser l'Intelligence Artificielle afin :

- d'alléger le tri manuel des images d'acquisition grâce à un algorithme de recherche et de classification,
- de concevoir un modèle afin de reconnaitre la structure d'un échantillon et de la rapprocher d'une base de données de structures grâce à de la reconnaissance d'images.

Afin de comprendre le fonctionnement de la reconnaissance d'images, un cas simple a été développé permettant de reconnaitre un type d'oiseau parmi 4 espèces retenues.

Récupération des données

La 1ère étape a nécessité la constitution d'un jeu de données à partir de photos d'oiseau des 4 espèces retenues.

L'utilisation de la reconnaissance d'images par apprentissage automatique nécessite d'être vigilant ; et notamment pour que le modèle soit fiable, il est nécessaire de :

- disposer d'un jeu d'images conséquent,
- s'assurer de l'absence de doublon d'images,
- tendre vers l'équilibre dans la répartition des images des différentes espèces.

La collecte des images peut nécessiter :

- l'acquisition de jeux de données d'images déjà constitués,
- la constitution de jeux de données soit par l'acquisition d'images unitaires soit par la prise de photos.

L'acquisition de jeux de données d'images déjà constitués est une méthode plus rapide mais qui peut être onéreuse. D'autre part, il n'existe pas forcément de jeux de données d'images sur tous les sujets.

La constitution de jeux de données d'images par acquisition unitaire ou prise de clichés est une méthode moins onéreuse mais qui peut nécessiter davantage de temps. Cependant cette méthode sécurise davantage le contenu du jeu de données.

Chaque donnée (en l'occurrence ici des images d'oiseaux) est caractérisée par 2 attributs : son nom et sa classe.

Dans notre cas, les 4 classes seront :

- chouette effraie,
- pygargue à tête blanche,
- épervier,
- aigle royal.

En projetant ces travaux au niveau du projet principal, les 4 classes concerneront des structures neutroniques telles que verre de glass, sphère, cylindre, ellipsoïde

Mise en forme des images

La 2ème étape a consisté en la mise en forme de l'ensemble des images afin de les rendre exploitables. Pour cela, il a fallu toutes les redimensionner selon le format choisi, dans notre cas 128 x 128 pixels.

Augmentation du nombre de données

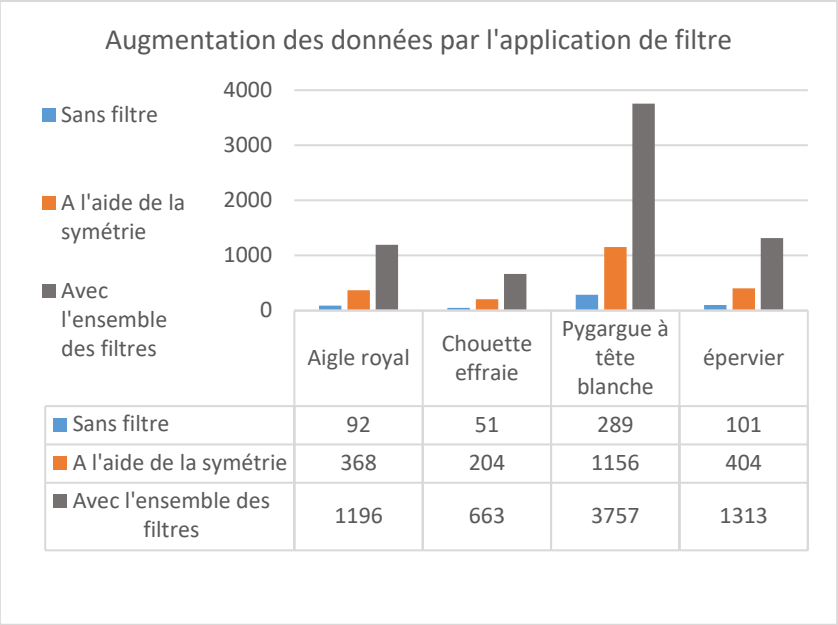
Comme précédemment indiqué, la richesse et la qualité du jeu de données d'images sont les critères majeurs pour constituer le modèle.

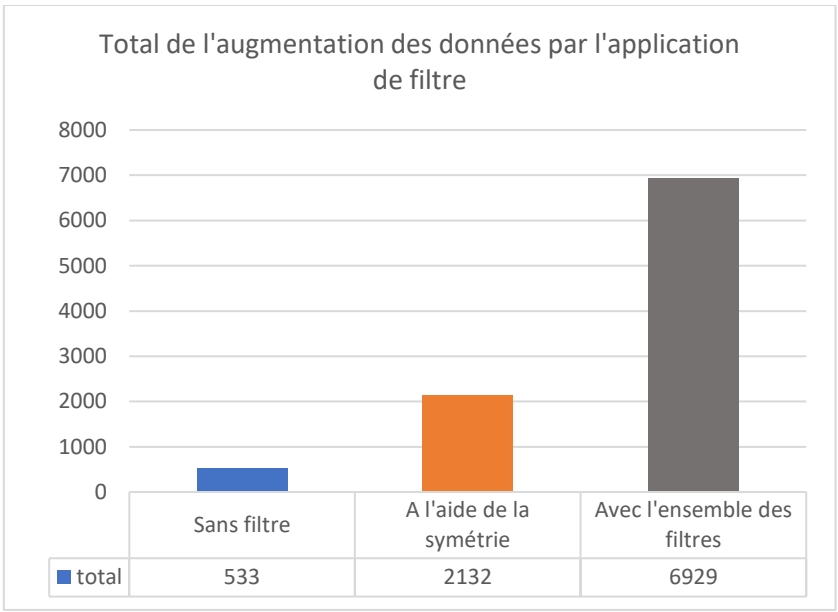
Afin d'augmenter « artificiellement » le jeu de données, nous avons appliqué différents filtres aux images.

Ceux-ci modifient légèrement l'image et créent ainsi des clones de celles-ci.

Parmi les principaux filtres, nous avons utilisé :

- une symétrie verticale,
- une symétrie horizontale,
- le floutage (« blur » en anglais),
- une rotation de 45°,
- un effet miroir combiné à rotation à 45°,
- un décalage de 10 pixels vers la droite,
- un décalage de 10 pixels vers la gauche,
- un effet miroir combiné à un décalage de 10 pixels vers la droite,
- un effet miroir combiné à un décalage de 10 pixels vers la gauche,
- un décalage de 20 pixels vers la droite,
- un décalage de 20 pixels vers la gauche,
- un effet miroir combiné à un décalage de 20 pixels vers la droite,
- un effet miroir combiné à un décalage de 20 pixels vers la gauche.





L'utilisation des filtres permet de faire croitre le jeu de données d'images de :

- 400 % en utilisant uniquement les symétries. Cette méthode n'a pas été retenue car elle ne fournissait pas assez d'images. D'autre part, les modifications apportées n'étaient pas assez importantes pour que les images soient distinctes.
- 1 300 % en utilisant l'ensemble des filtres. C'est cette méthode que nous avons retenu car elle donnait les meilleurs résultats en termes de quantité et de différence d'images.

Conception du modèle

Pour concevoir notre modèle, nous avons utilisé les librairies TensorFlow et Keras, qui sont 2 outils utilisés dans le Machine Learning.

Nous avons commencé par définir le pourcentage d'images utilisées pour entrainer le modèle. Ici nous avons retenu 80% du jeu de données d'images.

Les images restantes ont été utilisées pour valider le modèle.

Il est important de ne pas utiliser une même image pour l'entrainement et la validation du modèle au risque de fausser les résultats.

Nous avons ensuite indiqué les paramètres du jeu de données d'entraînement, à savoir :

- le pourcentage de 80% retenu précédemment,
- le label « training » pour spécifier que c'est le jeu d'entrainement,
- le mélange des images pour rendre l'entrainement aléatoire et ne pas sélectionner les mêmes images à chaque fois,
- la taille des images, pour notre cas 128x128 pixels.

Les paramètres pour le jeu de données de validation étaient identiques sauf le label qui a été défini sur « validation » et non plus « training ».

Après avoir paramétré les 2 jeux de données, nous avons paramétré le modèle en utilisant les librairies précédemment citées.

Les paramètres retenus ont été les suivants :

- le type de modèle, ici « sequential », applicable pour l'utilisation de réseau de neurone par couche,
- l'utilisation de la méthode « rescaling » afin de convertir la valeur des pixels comprise à l'origine entre 0 et 255 (selon l'échelle RVB) dans une nouvelle plage plus concise entre 0 et 1. On appelle cette opération la normalisation, cela permet d'aider la machine pour les calculs.
- l'utilisation de la méthode « Conv2D » qui va appliquer des filtres aux images. Cette opération permet de faire ressortir les caractéristiques fortes des images,

comme le contour par exemple et de les « nettoyer » afin de faciliter la reconnaissance automatique,

- l'utilisation de la méthode « MaxPooling2D », qui va réduire la taille de l'image à 64 x 64 pixels et ainsi permettre de réduire davantage les détails.

Entraînement et validation du modèle

A ce stade l'ensemble des paramètres ont été configurés.

Le modèle doit dorénavant être compilé afin de transformer le code source écrit en Python en code objet pour le rendre interprétable par la machine. Cette opération est réalisée à partir du code ci-dessous et l'optimisation utilisée au sein du service informatique :

```
model.compile(optimizer='adam')
```

Une fois le modèle compilé, son entraînement peut être lancé.

Pour entraîner le modèle nous avons déterminé le nombre de répétitions « epochs » que celui-ci devait mettre en œuvre, sachant que plus le nombre de répétitions est élevé, plus la précision du modèle sera grande. Le nombre de répétition a été défini à 10 car cela permet un compromis acceptable entre le temps passé à entraîner le modèle et la qualité des résultats en terme de fiabilité.

Les résultats de l'entrainement ont été affichés grâce à la librairie Plotly

Figure 10 : résultat de l'entraînement et de la validation du modèle

Sur le graphique de gauche sont représentés l'évolution de la précision de l'entrainement et de la validation du modèle (en ordonnée) en fonction du nombre de répétitions de 0 à 10 (en abscisse).

Sur le graphique de droite sont représentées l'évolution du nombre d'échecs lors de l'entrainement et la validation du modèle (en ordonnée) en fonction du nombre de répétitions de 0 à 10 (en abscisse).

On peut conclure que plus le nombre de répétitions est important, plus la précision du modèle est grande et moins le modèle génère d'échecs.

Afin d'augmenter la précision et de réduire le nombre d'échecs du modèle, il est possible de relancer l'entrainement et la validation d'un modèle déjà entrainé.

Elaboration du jeu de test

A ce stade, l'entrainement du modèle est finalisé, nous avons décidé d'éprouver sa précision à partir de 12 images d'oiseaux appartenant aux 4 classes retenues mais ne faisant pas partie du jeu de données.

Pour ce test, le modèle devait être en capacité de fournir la classe de l'oiseau testé ainsi que le pourcentage de fiabilité de la reconnaissance.

Ci-dessous un extrait des résultats obtenus :

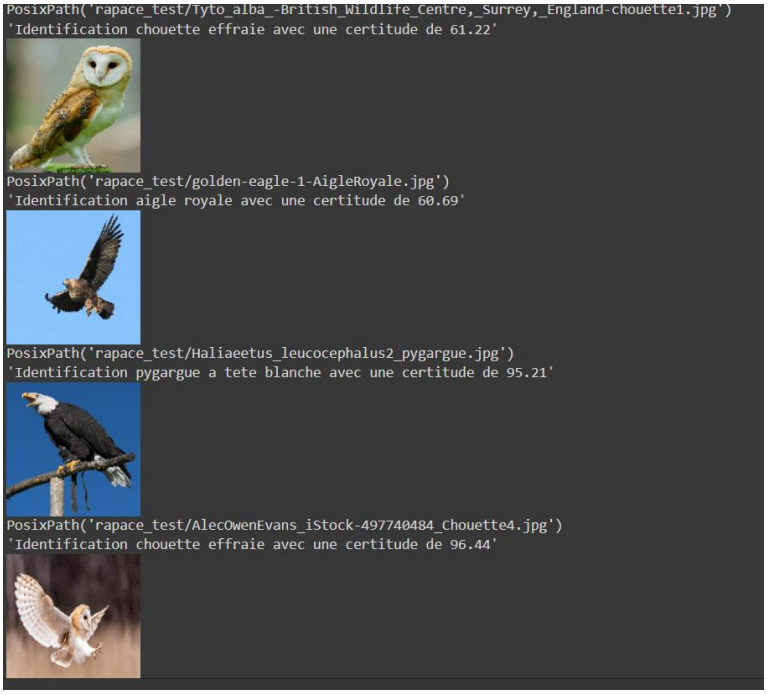


Figure 11 : résultats des tests de notre model

Le test a été concluant puisque :

- pour 75% des images testées (9/12), le modèle a été capable d'identifier leur classe,
- la fiabilité des classes correctement identifiées atteint 93,2% répartis comme suit :
    - pygargue à tête blanche : 100% de fiabilité
    - chouette effraie : 99,5 % de fiabilité
    - épervier : 92,5% de fiabilité
    - aigle : 77,5 % de fiabilité.

Ces résultats sont à considérer au regard de la taille du jeu de données d'images et de la répartition des images parmi les 4 classes.

Le faible taux de fiabilité concernant l'aigle royal est dû au fait que celui-ci présente une « forme » très proche de celle du pygargue à tête blanche. Cette ressemblance impacte la reconnaissance par le modèle.

Enfin, si le test est réalisé une nouvelle fois à partir des mêmes 12 images d'oiseaux, les résultats seront potentiellement différents, notamment du fait des fiabilités.

Conclusion

La reconnaissance d'images par apprentissage automatique est une technologie performante qui peut fortement aider les chercheurs. Cependant, elle nécessite un important travail de préparation, notamment pour constituer le jeu de données d'images, entrainer et valider le modèle.

D'autre part, le test sur un environnement simplifié a démontré les limites de la méthode, notamment en cas d'images de forme assez proche.

Cependant, cette méthode est viable et pourra être utilisée par les chercheurs pour reconnaitre les représentations graphiques et identifier plus aisément la composition chimique d'un échantillon.

Conclusion générale

Les objectifs qui m'avaient été fixés au début de mon stage ont été atteints. Les fonctionnalités attendues, les demandes et contraintes exprimées au travers du cahier des charges ont été respectées.

L'ensemble des fonctionnalités ont été testées, voire optimisées et améliorées afin d'obtenir un livrable simple d'utilisation et performant.

D'une part, l'application Web permettant d'afficher graphiquement les résultats des expérimentations neutroniques pourra être complétée par de nouvelles fonctionnalités en fonction des attentes des chercheurs.

D'autre part, l'outil de classification par reconnaissance par apprentissage automatique est fonctionnel. Le modèle pourra être également affiné en complétant le jeu de données de graphiques.

Enfin, celui-ci nécessite l'installation d'un exécutable. Son hébergement sur un serveur ou un cloud permettrait son accès plus facile via un lien hypertexte.

Spécification des logiciels et outils

Pour ce projet, j'ai travaillé à partir des outils utilisés par le Service Informatique du LLB, dont le principal est le langage Python qui était nouveau pour moi.

La première semaine de stage a donc été une semaine d'apprentissage au langage pour être opérationnel le plus rapidement possible. J'ai commencé par suivre des tutoriels et lire différente documentation sur Python et toutes les autres librairies. Dans un second temps je me suis entrainé en créant quelques outils utiles pour la suite du projet.

De plus j'ai également appris à travailler avec « Git » qui est fréquemment utilisé dans le monde de l'entreprise.

Microsoft Visual Studio Code *(https://code.visualstudio.com/)*

Microsoft Visual Studio Code est un éditeur de code développé par Microsoft pour Windows. C'est un environnement de développement intégré. Je l'ai utilisé tout au long du projet afin de développer tous les outils informatiques en Python. Il est pratique car on peut installer de nombreux modules extérieurs pour faciliter le développement, l'analyse et la correction de bug.

Git (*https://github.com/)*

Git est un outil de développement pour gérer les changements apportés au code source au fil du temps. Les logiciels de contrôle de version gardent une trace de chaque changement apporté au code dans un type spécial de base de données. Grâce à ces outils, on peut travailler en groupe de manière efficace et optimisée. Il permet aussi de garder un projet complet sans perte de fichier.

Db Brower (*https://sqlitebrowser.org/)*

DB Browser est un logiciel libre de gestion des bases de donnée sans avoir besoin de serveur. Avec une interface graphique facile d'utilisation il permet de faire des actions plus facilement.

Streamlit API (*https://streamlit.io/)*

Streamlit est une API open source en langage Python. Il nous aide à créer des applications web pour la science, l'analyse des données et l'apprentissage automatique. Il est compatible avec les principales bibliothèques Python, telles que scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib, etc... La mise en cache des données simplifie et accélère les calculs.

Google Colab (*https://colab.research.google.com/*)

Google Colab ou Colaboratory est un service cloud, offert par Google, basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud.

Difficultés rencontrées

Durant ce projet, je n'ai pas rencontré de problème majeur.

L'apprentissage du codage en Python n'a pas été une difficulté car les principes codages sont similaires aux autres langages informatiques.

Lors de la 1ère partie de mon stage, la conception de l'application a été l'étape la plus challengeante, car bien que l'utilisation d'une API facilite le travail, ses limites ne permettent pas forcément de faire tout ce qui est souhaité.

Lors de la 2ème partie de mon stage, la découverte de l'Intelligence Artificielle et de ses utilisations possibles a été très enrichissante mais a nécessité un important travail de recherche. Pour autant, cela ne m'a pas posé de difficulté étant donné la quantité infinie de ressources en ligne disponibles sur ce sujet en plein essor.

Bilan

Apports du stage

Ce stage aura été pour moi été une excellente expérience professionnelle au cours de laquelle j'ai pu appréhender le rôle et l'importance de l'informatique dans l'entreprise et mettre en application tout ce que j'ai appris lors deux années de DUT.

Sur l'aspect technique, découvrir et apprendre un nouveau langage, ainsi que plein de librairie, API et Framework ont été très enrichissant.

D'autre part, l'immersion dans le monde professionnel m'a également permis d'apprendre à coder en groupe. Cette expérience a été très instructive.

Enfin, travailler en autonomie m'a permis de développer mes capacités de résolution de problème, de recherche et d'adaptation à un nouvel environnement de travail, au code déjà écrit et aux actions réalisées en amont par un autre informaticien.

Perspectives futures

Ce stage a été pour moi la confirmation que je me suis orienté dans la voie que je veux suivre : le développement d'applications et le développement web.

Afin d'aborder des projets de plus en plus intéressants et complexes, je souhaite continuer à enrichir mes compétences et mes connaissances. La licence PRISM, licence Professionnelle Programmation Internet Et Systèmes Mobiles pour laquelle j'ai été retenu pour l'année prochaine devrait me donner entière satisfaction en cela.

Lexique

Machine Learning : Programme permettant à un ordinateur ou à une machine un apprentissage automatisé, de façon à pouvoir réaliser un certain nombre d'opérations très complexes.

API : Une « interface de programme d'application » (API) est un regroupement de routines, de protocoles et d'outils.

Framework : Un Framework contient des composants autonomes qui permettent de faciliter le développement d'un site web ou d'une application

SQL : est un langage informatique normalisé servant à exploiter des bases de données.

Dataset : Jeu de donnée en anglais, Un jeu de données est un ensemble de valeurs « organisées » ou « contextualisées », où chaque valeur est associée à une variable.

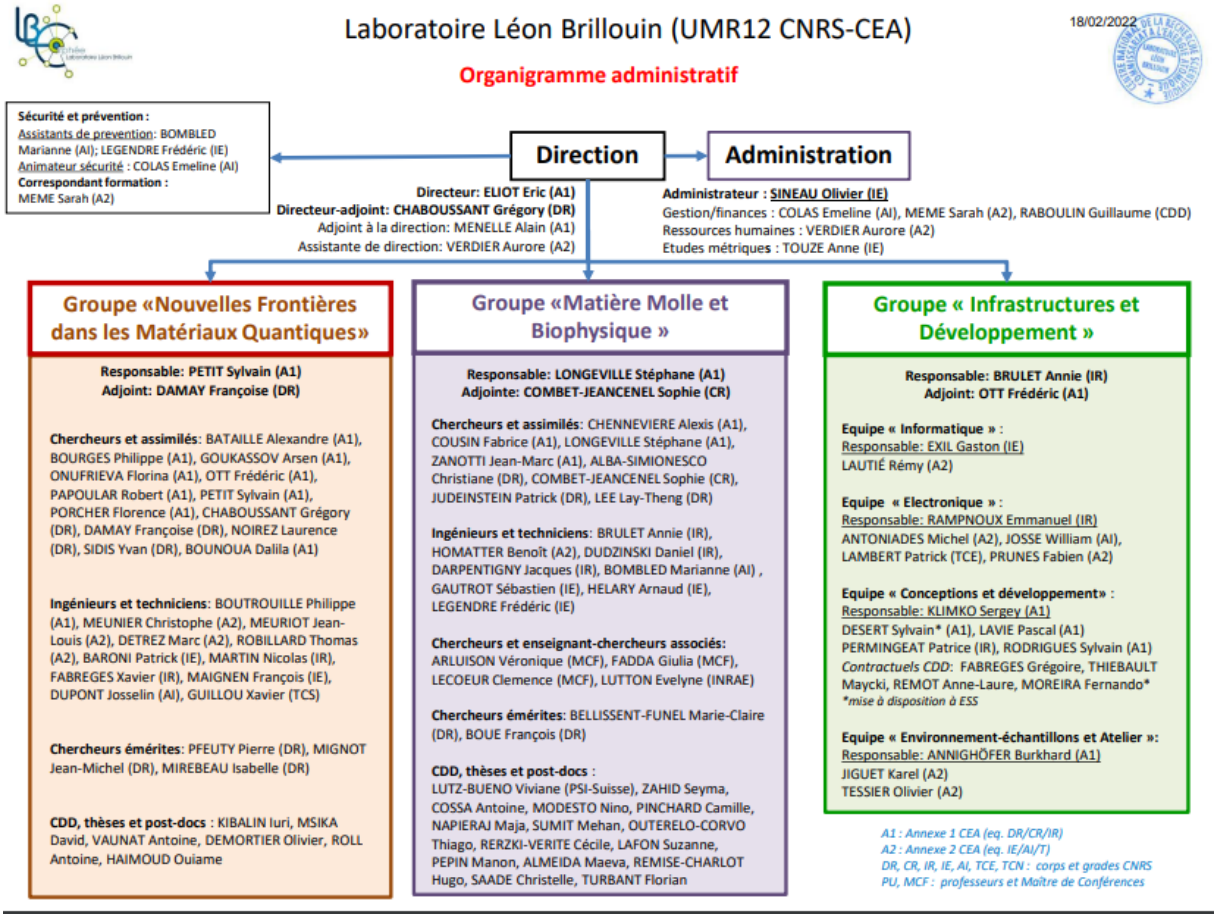
Model : est un fichier qui a été entraîné pour reconnaître certains types de modèles. Vous entraînez un modèle sur un ensemble de données, par un algorithme.

Checksum : est un dispositif qui permet de comparer des fichiers afin de s'assurer de l'absence de doublons.

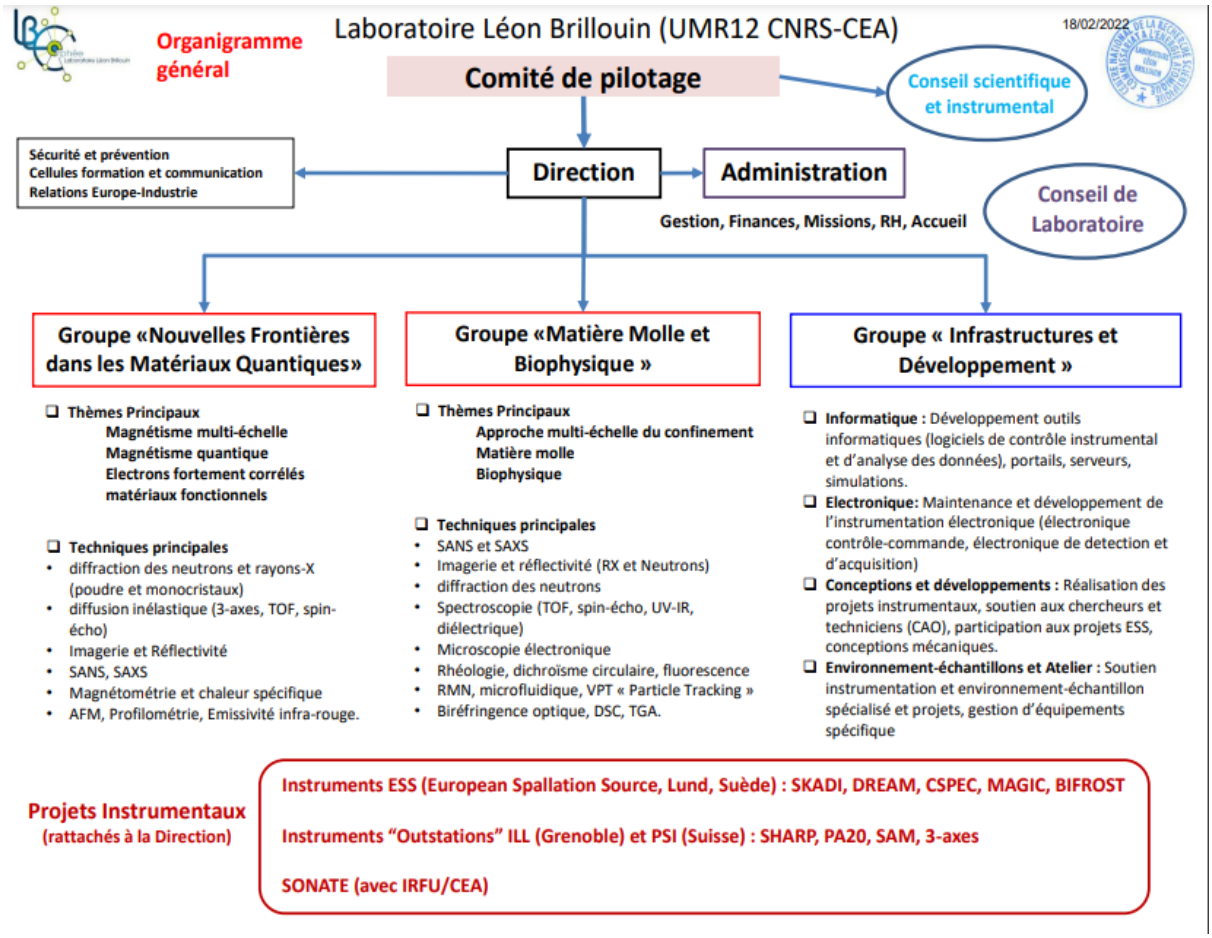
Widgets : est un outil de développement permettant de rajouter des fonctionnalités dans leur logiciel.

Table des illustrations

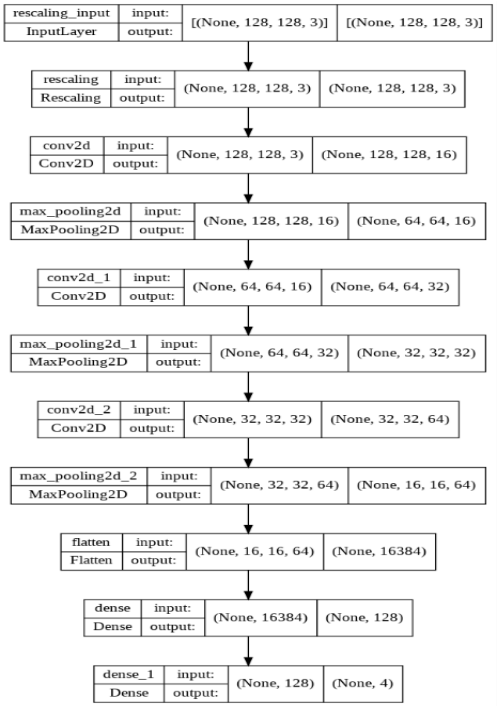
Annexes



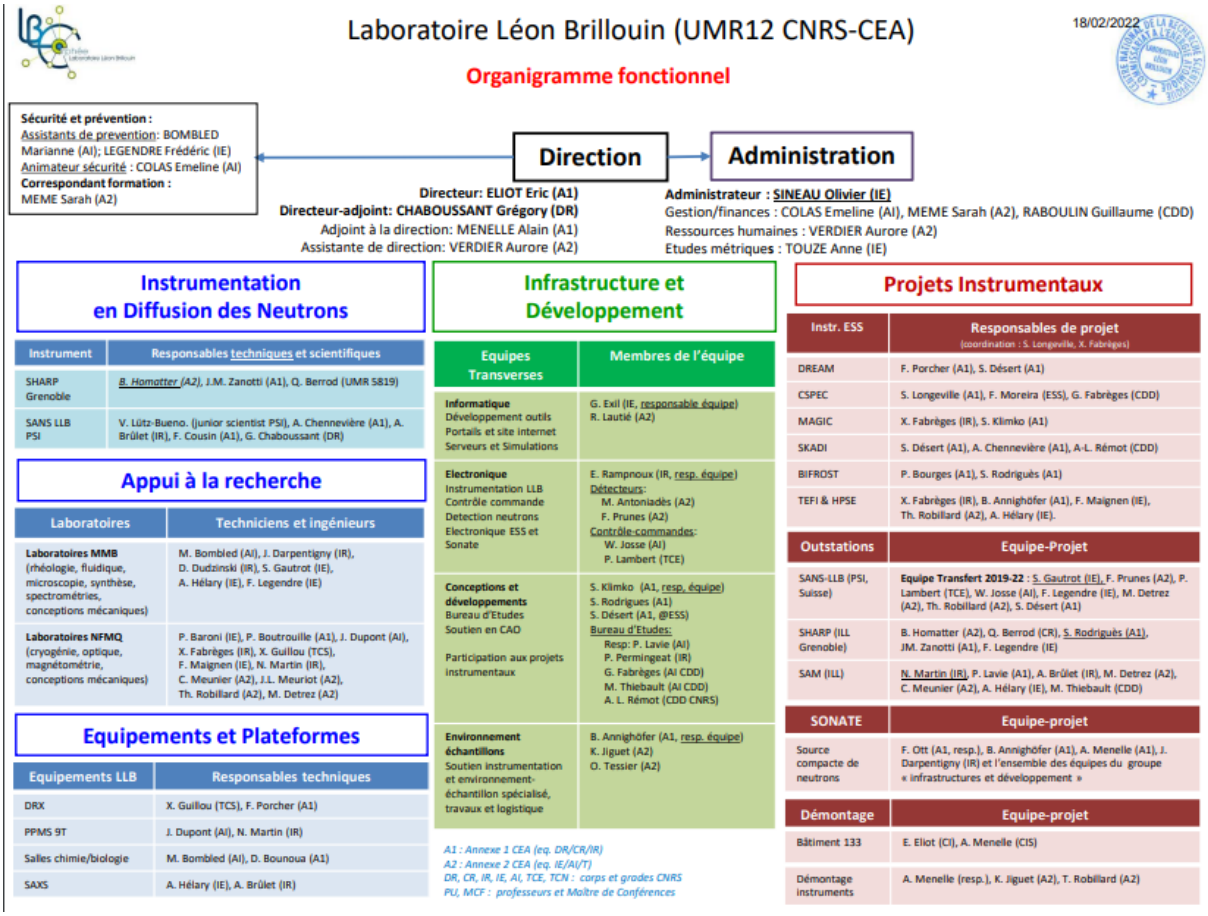
Annexe 1 : Organigramme Administratif du laboratoire



Annexe 2 : Organigramme Général du laboratoire



Annexe 3 : Architecture réseau de neurone



Annexe 4 : Organigramme Fonctionnel du laboratoire

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling (Rescaling)       (None, 128, 128, 3)       0

 conv2d (Conv2D)             (None, 128, 128, 16)      448

 max_pooling2d (MaxPooling2D  (None, 64, 64, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 64, 64, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 32, 32, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 16, 16, 64)       0
 2D)

 flatten (Flatten)           (None, 16384)             0

 dense (Dense)               (None, 128)               2097280

 dense_1 (Dense)             (None, 4)                 516

=================================================================
Total params: 2,121,380
Trainable params: 2,121,380
Non-trainable params: 0
```

Annexe 5 : Résultat de la compilation du modèle