



Rapppport de stage Assistant Ingénieur

S8 - Industrie 4.0

THEME:

**Application des techniques
de Deep et Machine
Learning dans la détection
et la classification de
Textile**

**Application of deep and machine learning
techniques in textile detection and classification**

Réalisé Par :

KAKPEU Hermann Désiré, Elève
Ingénieur de L'Ecole Centrale.

Tuteur de Stage :

Samir TETOUANI, Responsable
Filière IMS.

Synthèse

The Industry 4.0 Research Laboratory, located within the IMS department of ESITH, is equipped with several 4.0 devices allowing students, companies and visitors to familiarise themselves with new technologies and new MEMS sensors of materials whose installation, handling and manipulation are still not well known. The problem is actually to understand how these new sensors work and to use them in a concrete application within the context of industry 4.0 and ESITH, which is a textile school. It was therefore a question of studying the Nirone sensor in order to provide a simple, comprehensible and relevant manual that would allow any user to quickly get to grips with it. Then a Machine Learning project was carried out from data collection to the deployment of the ML and DL models in a built deskTop application for textile detection and classification.

Le Laboratoire de Recherche sur l'Industrie 4.0 se situant au Sein du département IMS de l'ESITH, est équipé de plusieurs dispositifs 4.0 permettant aux étudiants, à des entreprises et visiteurs de se familiariser aux nouvelles technologies et de nouveaux capteurs MEMS de matériaux dont l'installation, la prise en main et la manipulation restent encore peu connues. Le problème se situe donc au niveau de la compréhension du fonctionnement du capteur et la possibilité de l'utiliser dans une application concrète dans le contexte de l'ESITH qui est une école de textile. Il a donc été question d'étudier le capteur Nironne pour fournir un manuel simple, comprehensible et pertinent permettant une prise en main rapide pour un utilisateur quelconque. Ensuite un projet de Machine Learning a été réalisé depuis la collecte des données jusqu'au déploiement des modèles ML et DL dans une application deskTop Concue pour la classification de textile.

Sommaire

Synthèse	2
Sommaire	3
I. Introduction	4
II. Approches et Méthodes utilisées	6
1. Production et redaction de manuel d'installation et d'utilisation	6
2. Projet d'application - Textilus (l'analyseur de tissu)	7
Image 1 : le capteur Nirone de Spectral Engine et ses composants	8
III. Technologies choisies et les raisons de choix.	10
Image 2: moi au stage, sur mon poste de travail avec le capteur Nirone S2.0	10
Image 3: fichier text renvoyé par l'application sensorControl	12
Image 4: fonction python servant à passer de text à pandas	12
Image 6: échantillon coton Image 7: échantillon laine	13
Image 8: échantillon polyamide	13
Image 9: échantillon polyster	13
IV. Présentation détaillée de l'approche de résolution.	15
1. Elaboration du manuel d'installation et de manipulation	15
2. Textilus, l'application ML et DL	16
V. Résultats Obtenus.	19
1. Manuel d'installation du capteur	19
2. les modeles ML et DL	20
3. l'Application Textilus - Kivy (multiplateforme)	23
VI. Prolongement Possible du Travail	26
VII. Apports du stage et la perspective professionnelles	28
VIII. Schemas, graphiques et illustrations	29

I. Introduction

Ces recentes années ont été fortement marquées par la montée en puissance des données. La data représente aujourd'hui, comme on le dit souvent, le nouveau pétrole. En effet, Depuis nos smartPhones, les réseaux sociaux jusqu'aux grandes industries en passant par les maisons, les vehicules et d'autres appareils connectés, il est notoire que la collecte des données est plus facile mais surtout que ses domaines d'application sont diversifiés et ouvrent le champs à de nouvelles opportunités à travers tous les secteurs d'activité. Mais la Vérité est que la majeure partie de cette collecte de donnée, surtout dans le monde industriel, n'est possible que grace à une variété de capteurs de plus en plus sophistiqués et intelligents qui sont mis en place de nos jours. Il existe donc une diversité de capteurs avec des systèmes de fonctionnement spécifiques à la mesure cible. Les plus étonnant d'entre eux sont les capteurs de matériaux dotés de technologie MEMS(Micro Electronic and Mechanical System) utilisés pour détecter la capacité d'un matériau à absorber une certaine intensité lumineuse.

Le Laboratoire de recherche sur l'industrie 4.0 de LESITH (Ecole Supérieure de l'Idustrie du Textile et de l'Habillement) vient d'être équipé d'un de ces capteurs MEMS de matériaux appelé Nirone S2.0 et cela, par souci d'être une école à la pointe des nouvelles technologies offrant ainsi à ses étudiants l'opportunité de se familiariser avec de tels dispositifs pour devenir plus efficaces dans la suite de leur parcours

dans le monde professionnel mais aussi pour permettre à d'autres parties prenantes telles que des entreprises externes de bénéficier de certaines compétences relatives à cela. Le capteur Nirone étant nouveau et sa documentation se trouvant rare ou parfois complexe à déchiffrer pour un utilisateur quelconque, le problème majeur se situe dans la compréhension du capteur et son fonctionnement avec tous les périphériques, mais aussi, dans le contexte du textile et comme seconde part d'un autre projet de Robotique dans le «*Pick to Light*¹» du même labo, il est donc question de mettre en place des modèles Machine ou Deep Learning permettant au Robot de détecter automatiquement des tissus et de savoir de quel tissu il s'agit pour ensuite effectuer un tri puis une classification sur une ligne de production automatisée. Par souci de temps et de disponibilité des Robots, ces modèles ont dû être déployés dans une application juste pour tester leur performance.

¹ pick to light: pour parler du processus industriel selon lequel un robot ou dispositif automatisé détecte des objets à partir de la lumière renvoyée.

II. Approches et Méthodes utilisées

Le capteur Nirone S est un capteur de matériaux fonctionnant dans la gamme du proche infrarouge (NIR) de 1100 à 2450 nm et offre une sensibilité et une spécificité accrue pour la détection des matériaux. Il dispose d'une seule connexion électrique et son optique frontale réglable lui permet de s'adapter à une large gamme d'applications . Le Laboratoire de recherche sur l'industrie 4.0 de l'ESITH entend donc l'utiliser pour des applications dans le textile et dans d'autres domaines si besoin est. Mais le premier obstacle à cet objectif est que la documentation de ce capteur est rare et parfois pas à la portée de tous, tandis que l'école désire que l'outil serve d'instrument d'études pour ses étudiants qui en feront des applications diverses. La première étape de ma mission a donc été de construire un manuel personnalisé d'installation et d'utilisation du capteur Nirone S. Manuel facile d'accès pour n'importe quel futur utilisateur. Ensuite de l'utiliser dans une application concrète permettant aux futurs utilisateurs de découvrir un exemple de ce qu'on peut réaliser avec Nirone Sensor.

Quelles sont donc les Méthodes et les approches qui ont conduit mon travail et m'ont permis d'atteindre le mieux possible ces deux objectifs ?

1. Production et rédaction de manuel d'installation et d'utilisation

Produire un manuel d'installation simplifié, succinct et à la portée de tous sur un dispositif électronique et physique dont on ne comprends pas le principe de fonctionnement est sans aucun doute quasi-impossible. C'est pourquoi ma première approche a été pour ce volet, d'effectuer un état de l'art pour rassembler le maximum possible d'informations disponible sur Le capteur Nirone, sa composition et la base de fonctionnement de la grande famille des capteurs ou dispositifs fonctionnant sur le principe d'absorption de rayons lumineux par la

matière. Ensuite j'ai procédé par élimination des informations que j'ai jugées inutiles ou trop encombrantes tout en gardant aussi les sujets qui revenaient sans cesse tel que la spectroscopie en vue de les étudier plus en détail. Enfin, mon tuteur, M. Samir TETOUANI m'a mis en contact avec Mr Mehdi EL BOUCHTI , enseignant superviseur de Laboratoire à l'Esith, qui en savait beaucoup sur le sujet. J'ai appris auprès de lui les bases fondamentales de la spectroscopie infrarouge. Il m'a également remis des échantillons de résultats obtenus à partir de la grande machine à spectroscope de son laboratoire qui m'ont été très utiles dans la compréhension du fonctionnement du capteur Nirone et de la spectroscopie en générale. Le site officiel de Spectral Engine fournit une documentation riche bien que peu développée à certains égards sur le sujet tel que la nécessité de désactiver les signatures obligatoire des pilotes sur le PC. Celà a aussi été une grande source d'information dans la rédaction du dit manuel.

2. Projet d'application - Textilus (l'analyseur de tissu)

Le capteur nirone S est accompagné d'une partie matériel, un Kit composé d'un câble USB, d'une cible blanche servant au calibrage des références sombre et blanche pouvant exister dans l'environnement de travail , et d'une partie logicielle c'est à dire d'une application Desktop (*SensorControl*) permettant de visualiser en temps réel l'intensité lumineuse brute réfléchie, l'absorption, et la transmittance, ainsi que la collecte des données sous forme de fichiers .csv .

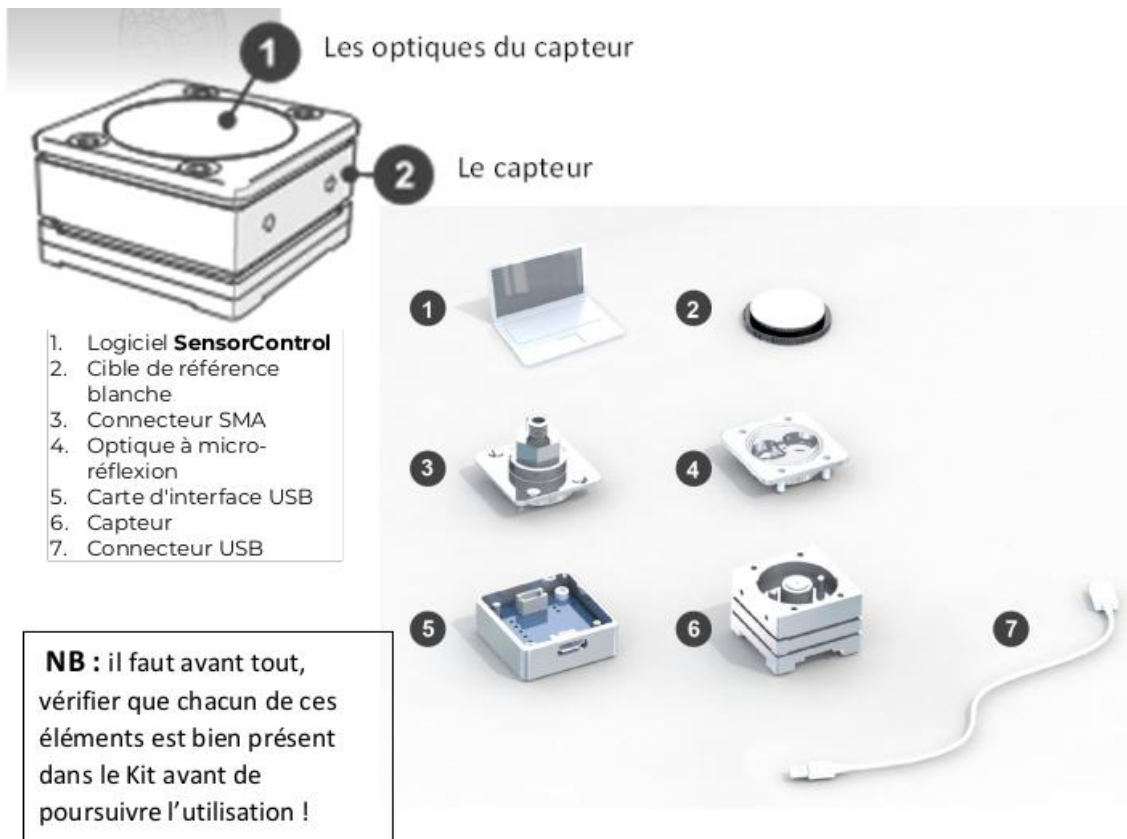
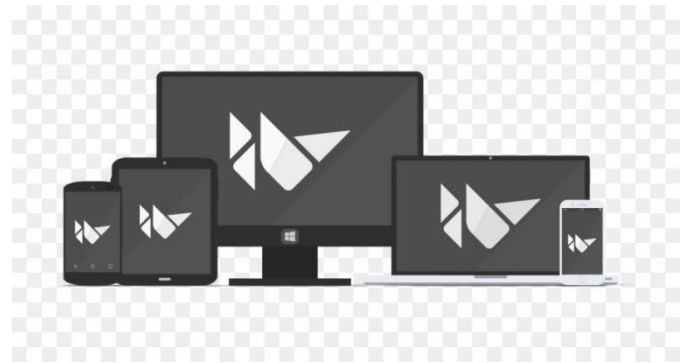


Image 1 : le capteur Nirone de Spectral Engine et ses composants

Graces aux recherches effectuées pour la constitution du manuel, j'ai découvert que spectral Engine proposait des solutions nous permettant de mettre en place notre propres logiciel Customisé conçu avec python ou tout autre langage de notre choix. Mon tuteur a donc jugé bon qu'on fasse comme projet d'application un logiciel personnalisé avec nos propres analyses adaptées à nos besoins et une intégration de modèles Deep et Machine Learning dans le cadre de la détection et la classification du textile qui sera plus tard associé à un projet de *pick to light* sur le quel travaillait un autre collègue élève ingénieur de l'Ecole Centrale. Il fallait donc choisir la technologie pour l'application bureau, le design, les différentes fonctionnalités et appreter les données, les nettoyer pour entrainer plusieurs modèles pour procéder ensuite à une série d'éliminations jusqu'à en garder ceux qui présentaient des précisions au delà de 80%. J'ai décidé par affinité de travailler avec python puisqu'il est mieux adapté pour le machine learning et pour ajouter une nouvelle compétence à mon panier et pour donner plus de dynamysme au projet, j'ai décidé de choisir Kivy au lieu

du classique Tkinter pour le developpement de l'appliacation bien que j'aie deja un certificat en developpement avec Tkinter.



III. Technologies choisies et les raisons de choix.

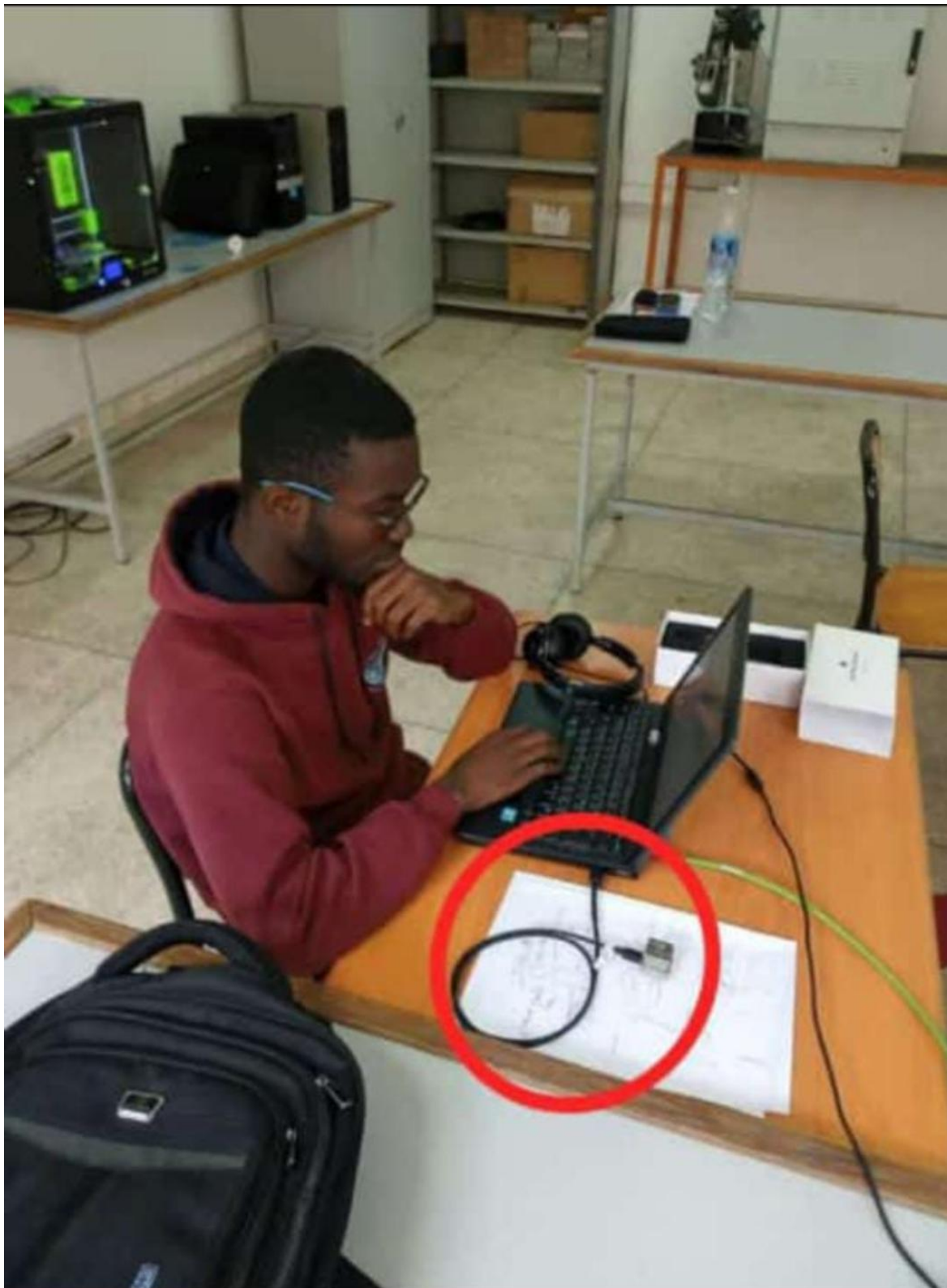


Image 2: moi au stage, sur mon poste de travail avec le capteur Nirone S2.0

Lorsque la décision de mettre en place une application a été finalement prise, puisque j'avais la liberté concernant les choix

techniques, ma première idée a tout de suite été de faire une application Web avec le framework django de python puisque je suis un peu familier avec lui. J'ai donc commencé une grande partie de l'application avec Django. mais la prise en main de l'application devenait de plus en plus complexe alors que le temps imparti pour la mission était relativement court. Il fallait gérer les bases de données et utiliser des bibliothèques javascript pour la visualisation des données, des bibliothèques telles que D3.js et Chart.js. D3.js est bien adapté puisqu'il permet de construire les visuels et les graphes du début à la fin et ceci à la convenance du développeur. Mais c'est un travail fastidieux qui demande une bonne maîtrise de JavaScript et aussi beaucoup de temps, ce qui me manquaient. Chart.js quant à elle, est une bibliothèque simple et facile à prendre en main mais puisqu'elle épargne le développeur des longs et complexes codes JavaScript pour ne se concentrer que sur la data à visualiser ainsi que les périphériques définissant les graphes. Cependant, il lui était difficile de supporter le poids important de données collectées par le capteur qui en effet, contenait des valeurs décimales très proches les unes des autres. Tout cela m'a donc conduit à penser à autre chose que Django. J'ai d'abord pensé à Bokeh, une bibliothèque python permettant la visualisation de données dans le navigateur. J'ai voulu l'associer à django mais le souci de temps persistait encore. Finalement j'ai opté pour une application desktop construite entièrement de façon customisée et adaptée aux besoins des parties prenantes. J'ai choisi de travailler avec python et pour la préparation des modèles et aussi pour la programmation de la dite application avec Kivy plus spécifiquement parce qu'il supporte toutes les plateformes sans complications et aussi et surtout parce qu'il offre de grandes opportunités d'amélioration. En effet, le Kit de Développement de Nirone S offre la possibilité de développer ses propres applications avec le capteur directement grâce à python, C++ ou Arduino. J'ai donc pensé qu'on pourrait à la longue se passer de Sensor Control pour la collecte des données et pour l'analyse et le déploiement de nos propres modèles machines et Deep learning et tout cela pourra même être automatisé. En ce qui concerne les

éditeurs de codes, l'application Textilus a été développée avec Pycharm et les traditionnels NoteBook Google Colab ont servis à développer entrainer et tester les modèles Machine et deep learning et aussi pour effectuer le preprocessing des données.

En effet, les données renvoyées par le capteur à travers l'application SensorControl de spectral Engine, étaient sous format .txt et contenaient des impuretés inadaptées à l'entraînement. Il a donc fallu écrire du code spécifique pour transformer le tout en dataframes pandas, supprimer les lignes et les colonnes indésirables, transformer les valeurs textes en numériques et nettoyer les données.

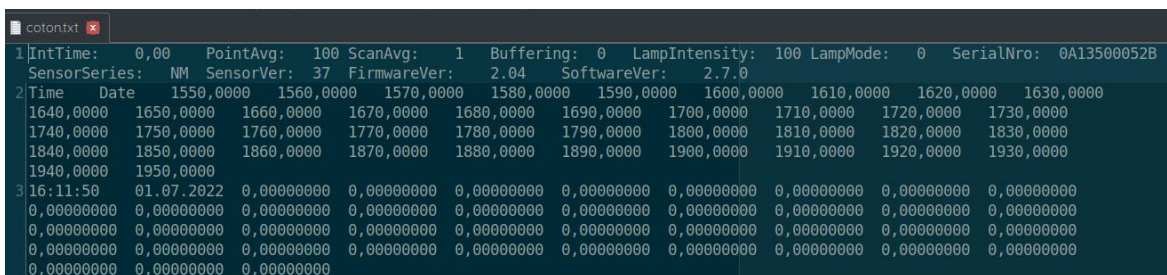


Image 3: fichier text renvoyé par l'application sensorControl

Image 4: fonction python servant à passer de text à pandas

```
def text_to_pandas(path, etiquette):
    """
    Cette fonction prend en entrée le chemin du fichier
    texte contenant les donnee recueillies par le capteur nirone à
    travers SensorControl et les retourne sous forme de DataFrames
    Pandas en ajoutant une colonne specifiant le type de tissu.
    """
```

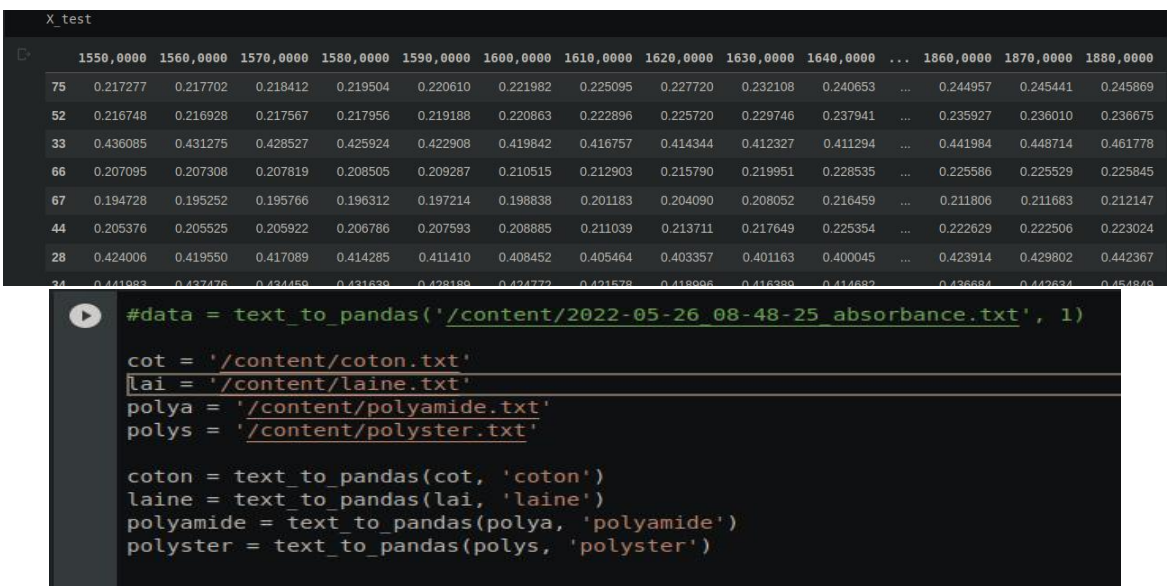


Image 5: pandas dataframe

Et Comme signalé précédemment, mon tuteur Monsieur TETOUANI m'a mis en contact avec Monsieur Mehdi EL BOUCHTI, enseignant superviseur de laboratoire à l'ESITH, qui a mis à ma disposition des échantillons de résultats de la spectroscopie du coton, de la laine, du polyster et du polyamide avec des échantillons de tissus pour que je sois à mesure d'effectuer mes propres prélèvements pour l'entraînement de mes modèles ML et DL. Il m'a aussi orienté sur le fait que la compréhension de la spectroscopie était essentiel pour mener à bien ce projet. Je me suis donc lancé dans beaucoup de recherches et inscrit à un cours intitulé «Introduction to molécular Spectroscopy» pour mieux comprendre la spectroscopie.

Image 6: échantillon coton



Image 7: échantillon laine

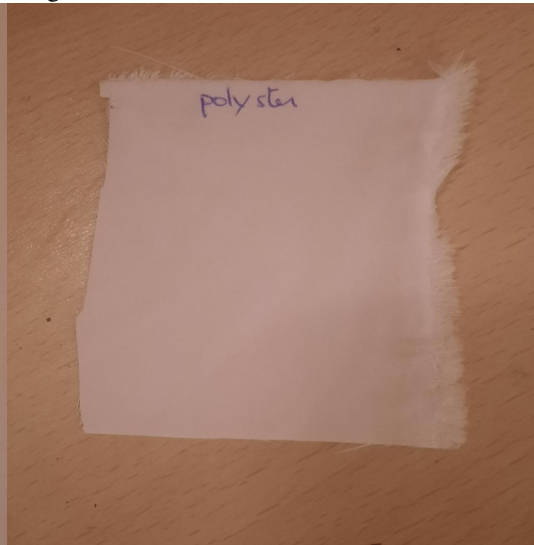


Image 8: échantillon polyamide

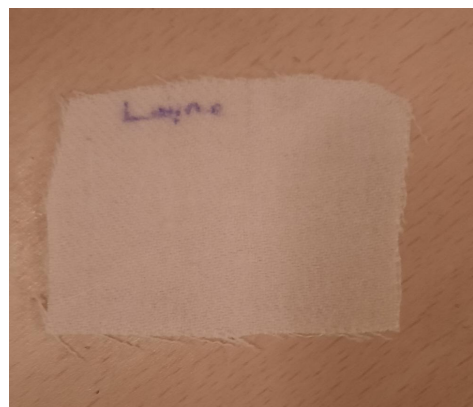
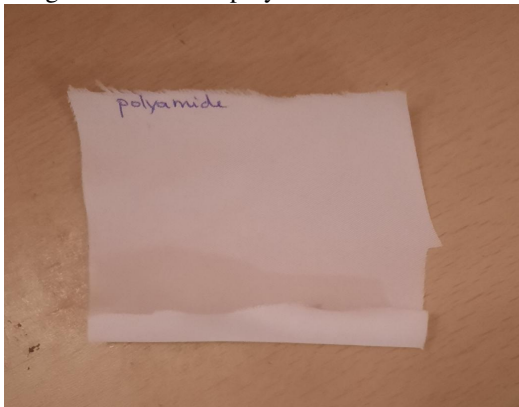
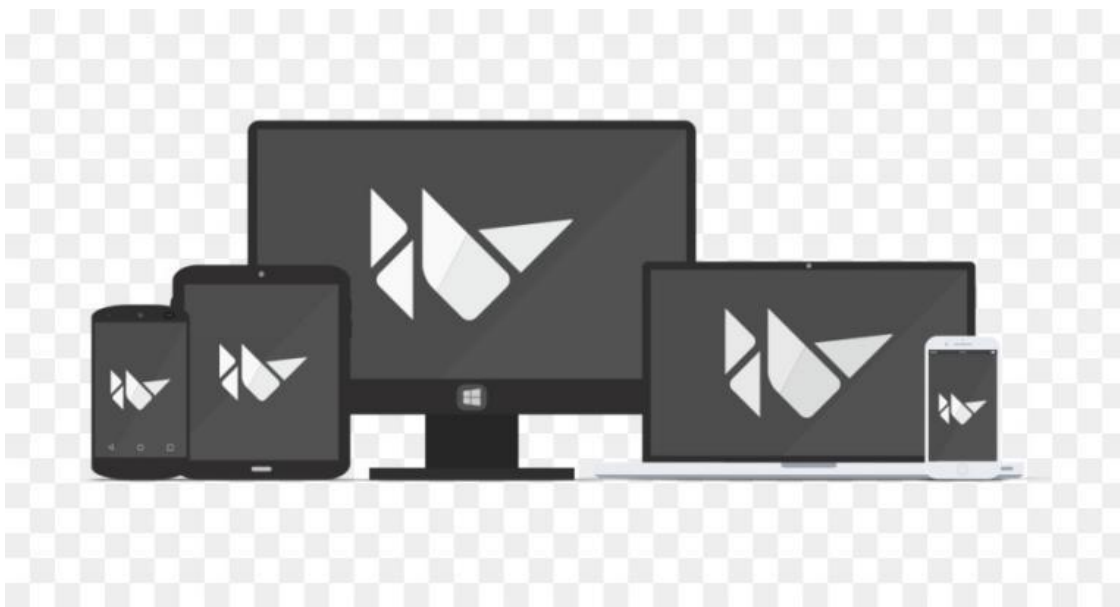


Image 9: echantillon polyster

C'est avec ces échantillons de tissus que j'ai effectué la collecte de données par le capteur Nirone S. Pour chacun des quatre tissus utilisés pour notre teste, il a été pris près de 20 mesures différentes afin de varier les valeurs et mieux entrainer les modèles.

En Outre, la bibliothèque python Kivy a été choisie pour le développement de l'application DeskTop (Textilus) parce qu'elle est flexible, facile à prendre en main donc bien adapté vu la contrainte de temps. les applications développée avec Kivy sont aussi multi-plateforme. Ainsi, l'idée fondamentale derrière Kivy est de permettre aux developpeur de créer une application une fois et de l'utiliser sur tous les appareils, redant le code réutilisable et deployable, permettant une conception d'interaction rapide et facile et un prototypage rapide.



Kivy Fournit également une large gamme de widgets construits avec un support multi-touch, une bibliothèque graphique utilisant uniquement OpenGL ES 2 et un langage kv intermédiaire un peu semblable à du CSS, utilisé pour concevoir facilement des widgets personnalisés.

L'application Textilus a bien été conçue même si elle n'est pas complètement terminée pour des raisons que je présenterai dans la suite du rapport. Textilus permet dans un premier temps de visualiser les données recueillies, les visualiser non seulement sous forme de table mais aussi sous forme graphique de différents types pour mieux appréhender l'étude.

IV. Présentation détaillée de l'approche de résolution.

1. Elaboration du manuel d'installation et de manipulation

Le manuel qu'il fallait produire devait avoir des caractéristiques bien spécifiques. Il devait être premièrement cohérent avec le guide d'utilisation produit par Spectral Engine et ensuite répondre aux exigences d'installation et du capteur sur les différents systèmes. Il devait aussi être simple et contenir des instructions faciles à suivre pour un utilisateur aux connaissances basiques en informatique. J'ai donc rassemblé toutes les informations complexes sur le capteur, sur la spectroscopie et sur les pilotes qui



servent à contrôler la connexion entre le câble USB avec le logiciel de contrôle du capteur.

Ensuite j'ai rédigé le rapport avec le suivi de mon maître de stage par rapport à ses exigences.

Finalement, le manuel a été

rédigé sur un design template de word avec un total de 11 pages seulement. et un design spécifique simple et esthétique. voici ci-

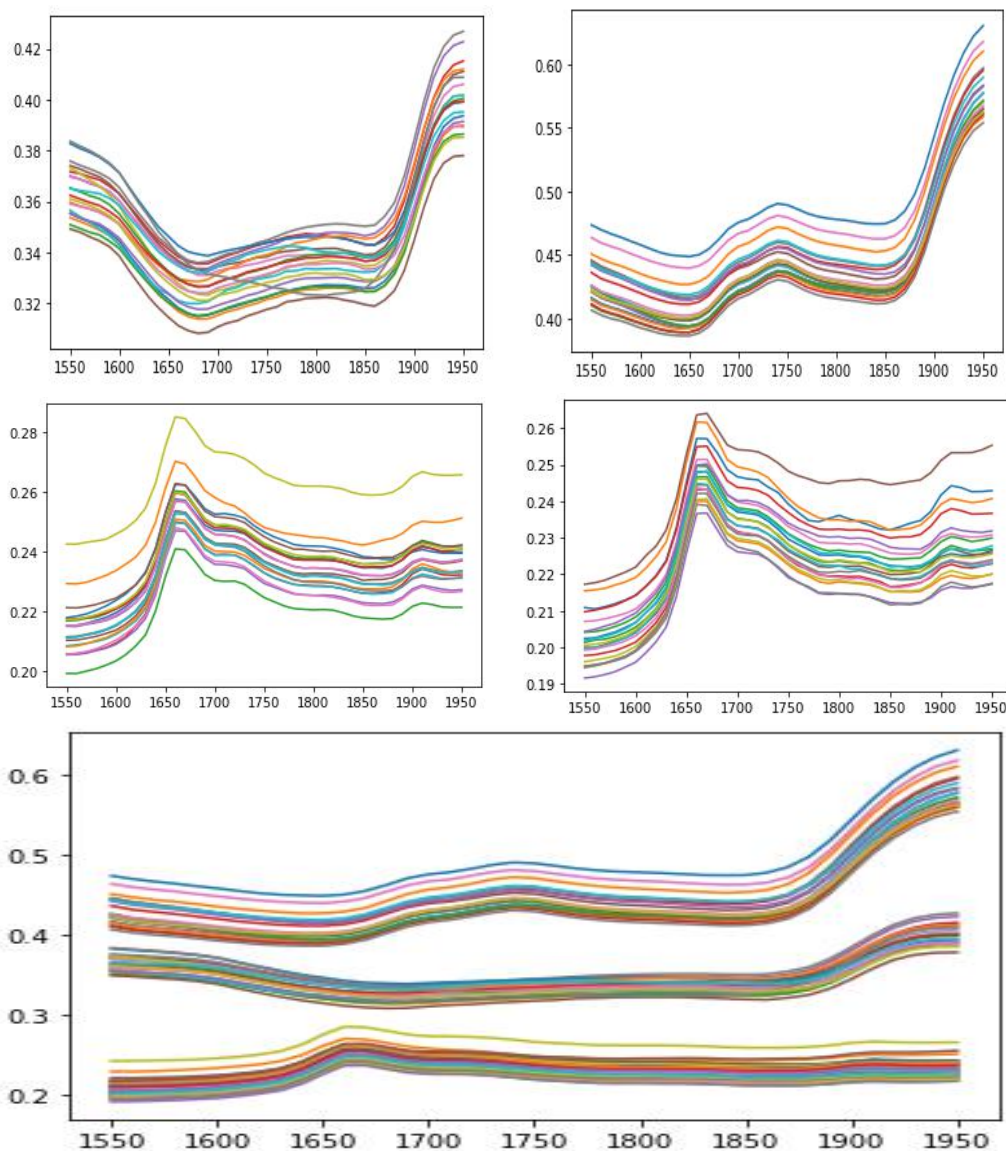
dessous un aperçu de deux pages du manuel. le lien pour manuel entier se trouve ici.

2. Textilus, l'application ML et DL

Mon travail dans le développement de l'application Textilus s'est structuré en deux grandes étapes. à savoir la construction des modèles ML et deep Learning et le développement de l'application proprement dit

Du premier côté, j'ai pu apprendre par l'expérience que j'ai eu au cours de cette mission en plus de ce que nous disaient les profs de machine learning que l'étape la plus importante dans la préparation de bons modèles, c'est effectivement l'étape de préparation des données communément appelée le preprocessing. Ainsi après avoir, à partir des différents échantillons, pris des mesures et collecté suffisamment de données, je suis tout de suite passé au preprocessing c'est à dire au nettoyage et à la préparation des données. Pour prélever les données pour chaque type de tissu, il fallait suivre les instructions du manuel, connecter le capteur à l'ordinateur et ensuite faire des réglages d'éclairage, choisir la plage de longueur d'onde souhaitée comprise entre 1550 nm et 1950 nm pour le premier capteur et de 1300 à 1600 nm environ pour le second capteur Nirx. Après cela il me fallait mesurer deux références fondamentales: La référence sombre qui fait référence au signal de base généré par l'électronique du capteur. Ce signal varie par exemple en fonction par exemple de la température du capteur ou de certains paramètres de mesure de la configuration. Pour obtenir de meilleurs résultats, nous souhaitons supprimer cet effet de capteur et mesurer uniquement la lumière réelle réfléchie par la cible. C'est d'ailleurs pour cela qu'il faut à chaque fois skoper ce signal avant chaque mesure. la deuxième référence à prendre en compte est la référence blanche. Elle se réfère au spectre de la lampe utilisée pour l'éclairage. Lorsque le signal de référence blanche est connu, le logiciel SensorControl est capable de calculer les unités d'absorbance à partir du signal brut. Pour cela, il nous faut utiliser la cible de référence blanche contenue dans le Kit du capteur. Une fois la data collectée, place au nettoyage et au preprocessing. Les données renvoyées par SensorControl sont sous format txt. des fonctions ont donc été développées pour automatiser le processus de conversion de txt en dataframe pandas avec des valeurs numériques utilisables dans l'entraînement de modèle. et une classe rassemblant toutes ces fonctions a aussi permis d'intégrer le tout dans l'application. les données contenaient des informations qui n'étaient utiles dans le cadre

de notre projet. Il s'agissait par exemple, de la date de prélèvement, l'intensité lumineuse, la moyenne des points et du scanne... Ces informations ainsi que les valeurs nulles aux deux premières lignes de chaque dataset ont été correctement supprimées. Avec Matplotlib j'ai effectué des sequences de visualisations pour voir aussi l'allure des courbes les vraisemblances et se donner une idées des modèles à choisir. En Outre, j'ai conclut qu'il était évident de procéder à un apprentissage supervisé. J'ai donc intégré dans la fonction de transformation de texte à pandas un attribut permet d'étiqueter les prélèvement de chaque tissu de sorte à avoir à la fin un grand dataset labelisé. Le pas de mesure était de 10 parce que en 10 nm la variation est négligeable. Ainsi de 1550nm à 1950nm on a un dataset de 41 colonnes puis 42 lorsqu'on ajoute la colonne des labels (target). En tout on a dont 42 colonnes et 80 lignes avec des valeur décimale à six (6) chiffres après la virgules.



Les spectres que nous observons ci dessus sont respectivement ceux du coton, de la laine, du polyamide, du polyster et le dernier présente tous les spectres sur une meme figure. On apperçoit clairement grace à la visualisation que de simples algorithmes ne suffiraient pas à établir la classification et faire des prédictions. En effet les spectres du polyamide et du polyster semblent etre complètement confondus. d'ou la nécessité d'introduire des modèles machine et deep learning adaptés à la classification. les modèles ML que j'ai utilisé sont SGDclassifier et K-NeighborsClassifier. J'ai fait varier les normalisations et j'ai comparer les métriques pour ne garder que les modèles avec les meilleures précisions. Pour le modèle DL, je n'ai utilisé que des denses Layers puisque le modèle était assez simple pour les réseaux de neurones. J'ai fait appel à mes connaissance en machine et deep learning pour mesurer la pertinence des modeles et les précision, voir éviter l'overfitting.

De l'autre coté, en ce qui concerne la conception de l'application en elle meme, ca m'a pris beaucoup plus de temps parce qu'il me fallait apprendre Kivy et ensuite commencer l'implémentation. Heureusement, Kivy est une bibliothèque que j'ai trouvé facile à prendre en main. J'ai commencé en faisant une design graphique à la main de l'interface utilisateur de l'application. et grace aux layouts que propose Kivy j'ai réussi à reproduire mon croquis en ajoutant les boutons nécessaires, les vues, les labels et tous les autres composant de chaque page.

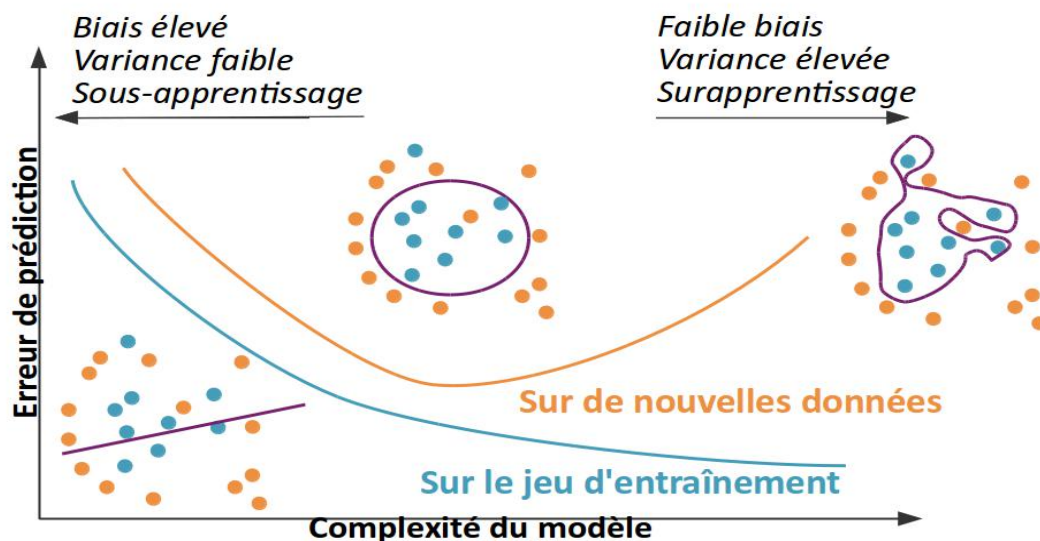
V. Résultats Obtenus.

1. Manuel d'installation du capteur

Le manuel a bien été réalisé avec l'approbation de mon tuteur de stage. voici le [lien vers le manuel](#) entier. Il est flexible et facile à utiliser les instructions sont précises et concises avec des images de cas pratiques pour faciliter encore plus la compréhension. Un espace est réservé dans l'application textilus pour reconduire également vers le manuel. Après avoir exposé le manuel à mon maître de stage il a été satisfait parce qu'il permet convenablement à un utilisateur quelconque de juste suivre les étapes et parvenir aux résultats souhaités. On l'a plusieurs fois essayé et les étapes fonctionnent lorsqu'on les suit à la lettre. C'est nouveau parce que la documentation sur l'utilisation de nirone sur internet est rare et en plus si elle existe elle est complexe. Ce manuel permet de gagner du temps pour ne se concentrer que sur l'essentiel. Il sera d'une aide précieuse pour les nombreux étudiants qui comme moi, travailleront sur d'autres projets ou des projets dans la continuité de celui-ci avec Nirone S. C'est en essayant d'installer l'application par moi même que j'ai découvert avec peine qu'il fallait désactiver, pour les PC qui utilisent windows comme système d'exploitation, l'application de signature obligatoire des pilotes. Sans cela, rien ne fonctionne même quand on a suivi à la lettre le gros manuel de Spectrale Engine. C'est donc un plus que ce manuel apporte car j'y explique clairement cette nécessité et comment les désactiver.

2. les modeles ML et DL

Partant sur le principe que en machine Learning le bon modèle c'est le modèle qui généralise le mieux, j'ai cherché à construire et entrainer les modèles de sorte à éviter soit un sur apprentissage soit un sous apprentissage. et pour cela je joue sur le compromis biais variance.



Voici ci-dessous les scores obtenus pour les les modèles ML et [ici le lien complet](#) vers le notebook utilisé por developper les modèles. Le projet ML et DL sur lequel je travaille est la pièce manquante d'un autre projet sur lequel a travaillé un autre stagiaire. Le but générale étant d'associer ces modèles à au robot industriel du laboratoire dans le cadre du pick to light. Le robot, considéré comme dans une usine de textile devra etre capable de sa deplacer seul dans un environnement complexes en permanent changement pour prendre des tissu d'un endroit à un autre après avoir detecté le type de Tissu il procède à un tri sur une ligne de production.

```

print('')
print(model3.score(X_train, code1))
print(model32.score(X_train, code1))
print(model33.score(X_train, code1))

0.765625
0.734375
0.765625
*****

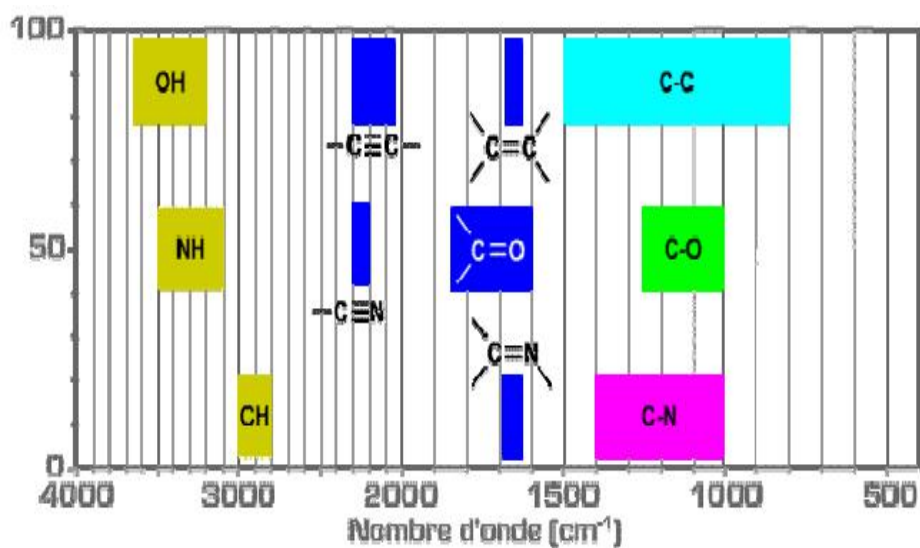
0.8125
0.8125
0.859375
*****

0.9915980778672748
0.9915980778672744
0.9915980778672744

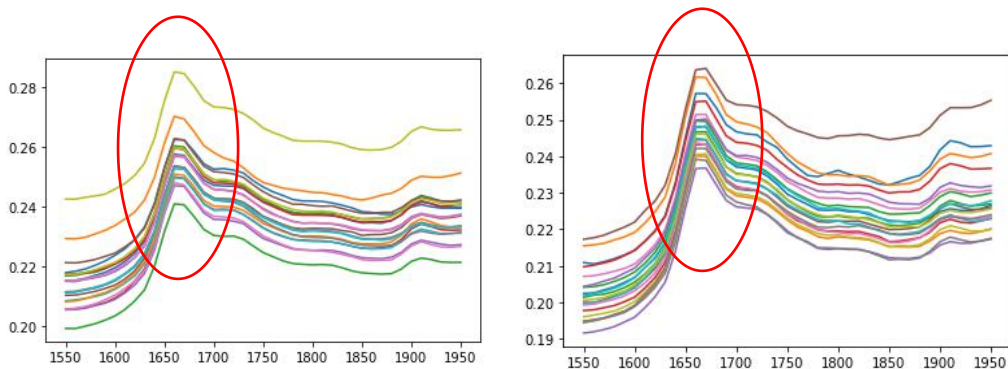
```

une limite des résultats est que l'apprentissage du modèle se fait très bien lorsqu'il s'agit d'effectuer la classification entre coton laine mais en ce qui concerne le polyester et le polyamide l'erreur est difficile à corriger. En réalité, d'après la spectroscopie, les plus importantes informations sur les molécules et la réflexion des rayons lumineux lorsqu'ils entrent en contact avec la matière se trouvent au niveau de certaines valeurs précises de longueur d'onde. Cela est dû au fait que les radiations électromagnétiques vibrent à une certaine fréquence. Les molécules de la matière aussi vibrent. ainsi lorsque certaines longueurs d'ondes rencontrent dans la matière des molécules qui vibrent à leur fréquence alors il y a interférence et donc absorption de la lumière. Les informations importantes qui distinguent une matière d'une autre se trouvent donc sur les pics dans les spectres d'absorption de ces matières. Tandis que lorsqu'on observe les spectres du polyester et du polyamide, on voit qu'ils ont sensiblement les mêmes allures et les mêmes pics. La classification est donc difficile à s'effectuer par de simples modèles ML. Néanmoins le modèle KNeighborsClassifier donne des résultats un peu satisfaisants pour un début. La grande limite qui cause cela se trouve dans les performances du capteur Nirx S. la version disponible au Laboratoire ne pouvait envoyer des rayons lumineux avec des longueurs d'ondes comprises uniquement entre 1550 et 1950 nm alors que la majorité de l'information utile se trouve au-delà de cette plage comme vous pouvez le constater sur cette image ci-dessous

Liaison	Nombre d'onde (cm ⁻¹)
Liaison OH libre	Entre 3500 et 3700 cm ⁻¹
Liaison OH liée (liaison hydrogène)	Entre 3100 et 3500 cm ⁻¹
Liaison N-H	Entre 3050 et 3500 cm ⁻¹
Liaison C=O	Entre 1700 et 1800 cm ⁻¹
Liaison C-H	Entre 2900 et 3100 cm ⁻¹
Liaison C=O des esters	Entre 1700 et 1750 cm ⁻¹
	Entre 1660 et 1740 cm ⁻¹
Liaison C=O des amides	Entre 1630 et 1710 cm ⁻¹
Liaison C-H de CHO	Entre 2650 et 2800 cm ⁻¹
Liaison OH des acides carboxyliques	Entre 2500 et 3300 cm ⁻¹
Liaison C-O des acides carboxyliques	Entre 1200 et 1320 cm ⁻¹
Liaison C-O des esters	Entre 1210 et 1260 cm ⁻¹
Liaison NH des amides	Entre 3050 et 3500 cm ⁻¹
Liaison NH des amides substituées	Entre 3050 et 3400 cm ⁻¹



Vous pourrez voir dans la section réservée aux graphiques des images de d'autres spectres de ces même échantillons relevés avec une machine beaucoup plus puissante dans le laboratoire de monsieur EL BOUCHTI.

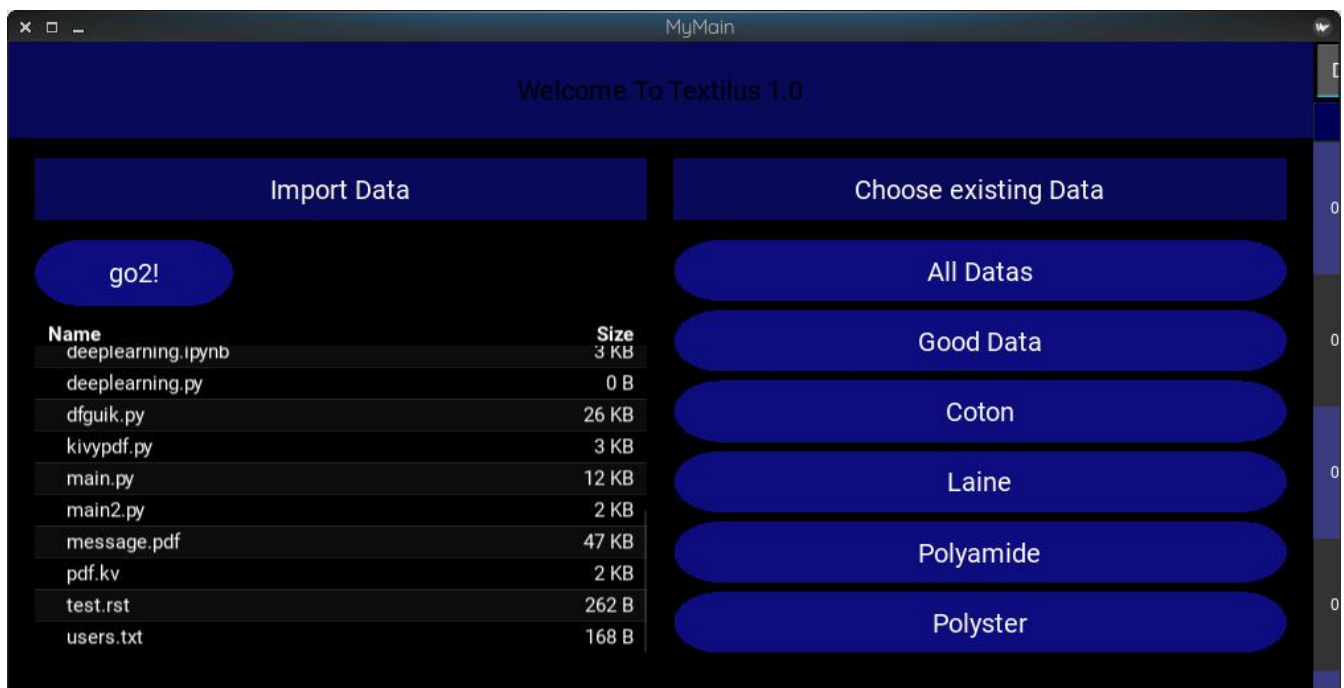


BOUCHTI. Comme on le constate, le spectre du polyamide et du polyester se ressemblent sur cette plage donc impossible de reelement

les distinguer alors que les capteur dont nous disposons est incapable d'aller au delà de ca.

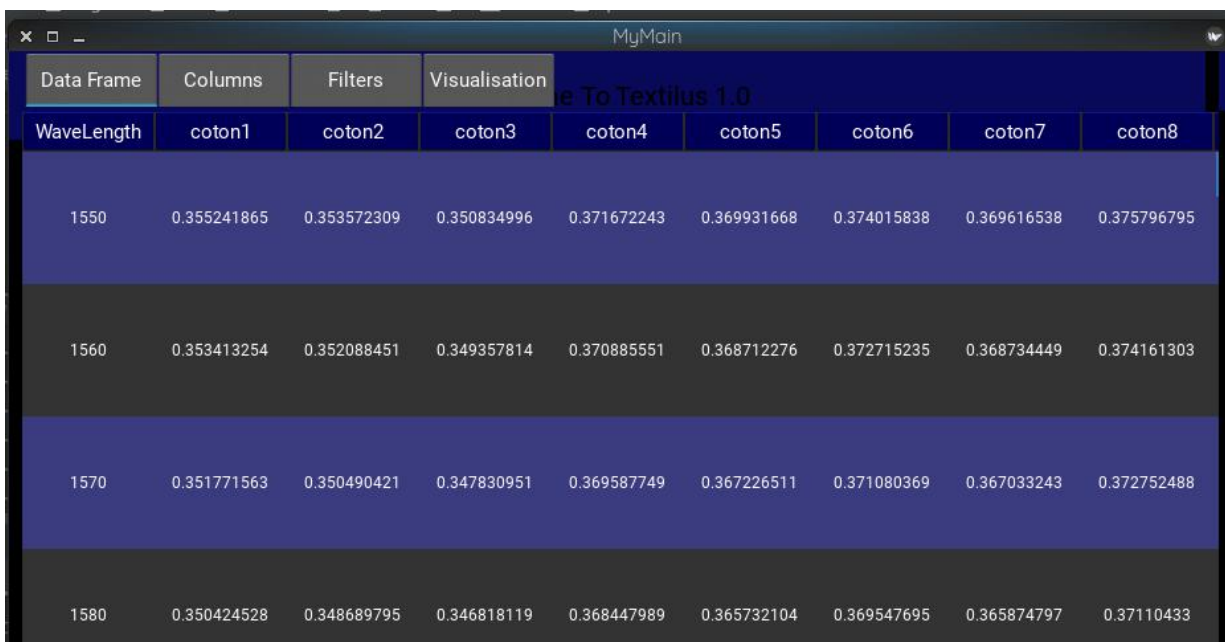
3. [l'Application Textilus - Kivy \(multiplateforme\)](#)

La version 1.0 de l'application Textilus([une vidéo démo](#)) a été entièrement réalisée avec python, plus précisément grace à la bibliothèque Kivy. elle comprends un système d'accès avec mot de passe donc l'intégration d'une gestion de base de donnée. Elle dispose de trois grandes pages dont la première est dédiée au choix d'un data su lequel travailler. Il est possible d'importer les données depuis son ordinateur vers l'application. L'application a été concue de facon spécifique aux données renvoyées par SensorControle de spectrale Engine, elle s'attend donc à recevoir des dataset sous format .txt ou .csv pour ensuite effectuer toutes les opérations possible sur les données et les insérer dans son système. Il est aussi possible de choisir des données deja existant dans l'application, ce sont les échantillons utilisés pour entrainer les modèles ML et DL.

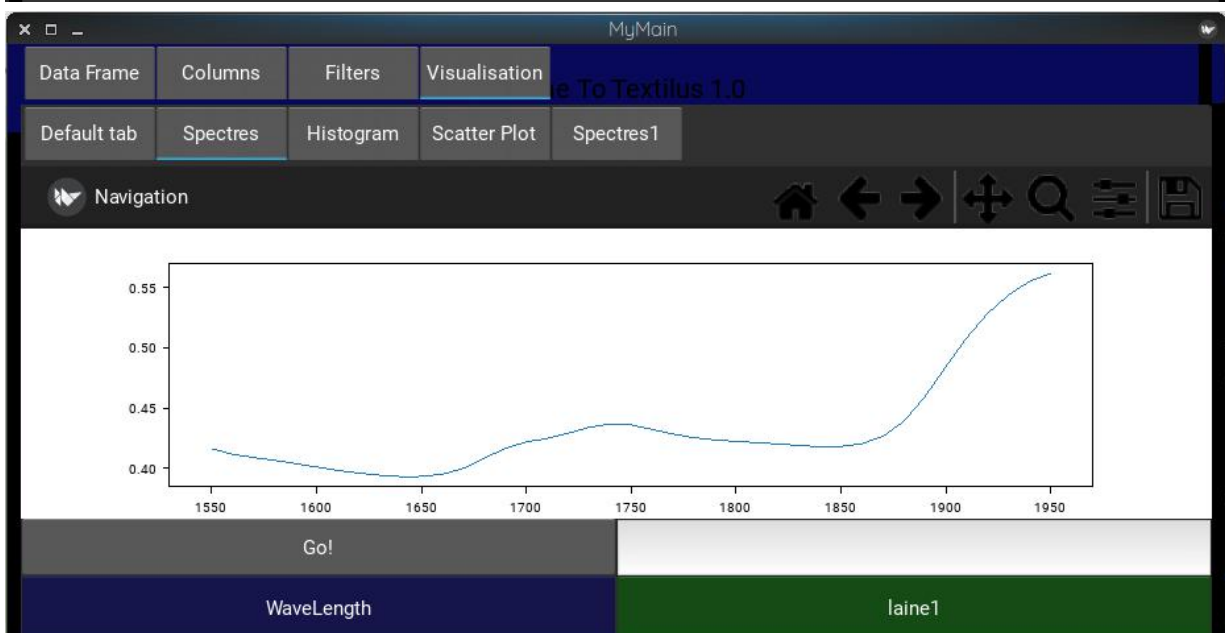


La deuxième page principale affiche le dataset selectionné et tous les services que l'application offre sur les données. C'est principalement de la visualisation de données. Il est possible de voir le spectre de chaque enregistrement , de visualiser aussi l'histogramme et le scatterplot.

C'est une application multi-plateforme, c'est à dire qu'elle n'aura aucun problème à fonctionner sur PC, sur android, MACOS, Linux, Windows...

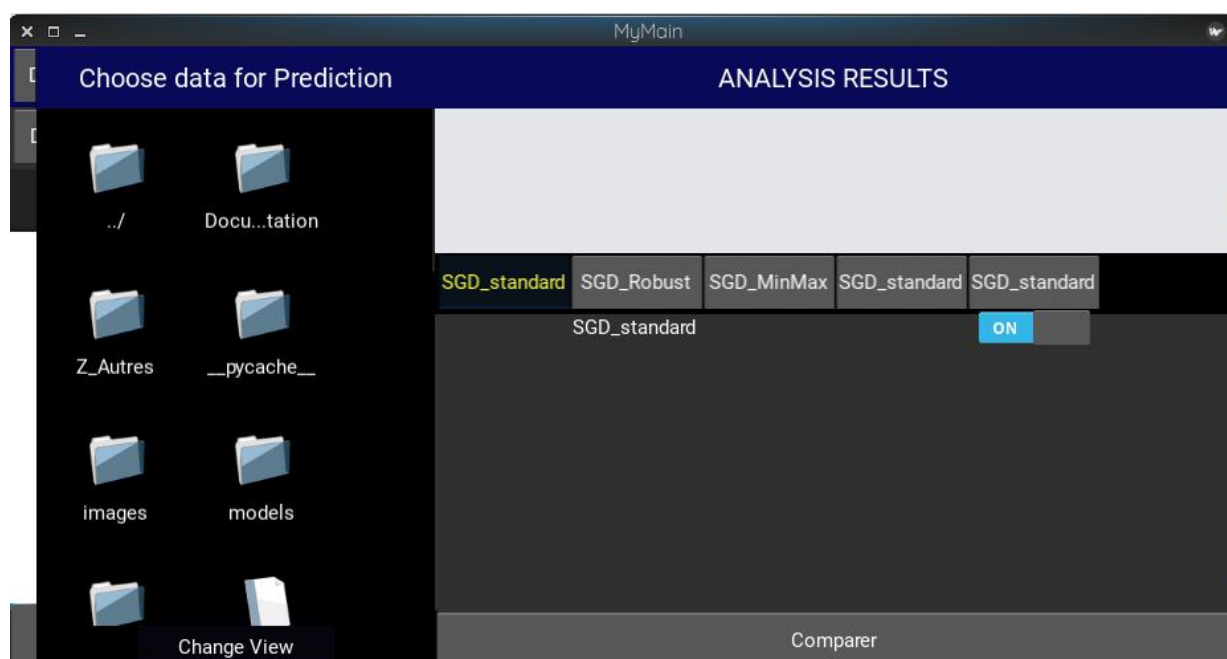


WaveLength	coton1	coton2	coton3	coton4	coton5	coton6	coton7	coton8
1550	0.355241865	0.353572309	0.350834996	0.371672243	0.369931668	0.374015838	0.369616538	0.375796795
1560	0.353413254	0.352088451	0.349357814	0.370885551	0.368712276	0.372715235	0.368734449	0.374161303
1570	0.351771563	0.350490421	0.347830951	0.369587749	0.367226511	0.371080369	0.367033243	0.372752488
1580	0.350424528	0.348689795	0.346818119	0.368447989	0.365732104	0.369547695	0.365874797	0.37110433



La troisième et dernière grande page de Textilus sert au déploiement du modèle ML et DL. C'est ici qu'on fait les prédictions. J'avoue que j'ai rencontré beaucoup de difficultés à construire un système facile permettant de d'insérer les données et obtenir le type de tissu. On pouvait réaliser une sorte de formulaire classique que l'utilisateur remplit pour obtenir le type de tissu mais le problème est que les entrées ici sont des vecteurs de 41 entrées. Il serait donc fastidieux de demander à quelqu'un de rentrer 41 valeurs pour avoir une seule prédiction. J'ai donc opté pour une sélection directe via un fichier

contenant toutes les données à classifier. On peut donc utiliser le capteur pour collecter des données, on les enregistre dans le dossier de textilus et une fois ces fichiers présents on les téléverse dans textilus qui à son tour effectue les classifications rapidement. C'est pratique c'est rapide et ça marche. On est parfaitement parvenu à deployer les modèles ML et DL dans une application concrète ce qui signifie que le premier but est atteint. Cette application avec est très prometteuse dans le sens où il est possible que nous automatisons tout ce processus directement via le capteur. Cependant je n'ai pas pu le faire pour deux raisons fondamentales. Premièrement parce que le seul moyen d'avoir accès à l'interface de développement personnalisée du dispositif de Spectrale Engine c'est d'avoir le Kit SDK qui lui est payant. et la deuxième est une contrainte de temps.



VI. Prolongement Possible du Travail

Je suis enthousiaste rien qu'à penser aux multiples ouvertures que ce travail apporte. Il y a en effet beaucoup de prolongement possible. D'abord, avec le SDK (Kit de développement logiciel) on transforme textiles en notre propre logiciel de contrôle des capteurs spectraux et des sources lumineuses. En quelques clics nous pourrions connecter les produits spectral Engines à nos propres systèmes de mesure et gagner du temps. C'est seulement ainsi qu'il nous sera possible de connecter le capteur directement à textiles pour faire les analyses ou même le connecter avec un robot de l'industrie textile. En effet le SDK contient une description complète du protocole de communication série. Cette description donne la liberté d'implémenter les commandes dans n'importe quel autre langage de programmation comme Python par exemple. Il serait simple d'intégrer le capteur même dans des systèmes embarqués.

Par ailleurs, le volet esthétique de l'application n'est pas parfait, ce travail pourra être confié à un UX designer pour le rendre plus beau et aussi pour disposer les composants de sorte à faciliter l'utilisation et à améliorer l'expérience client. Le choix des couleurs, des logos, des formes, des boutons et labels etc...

En outre, les versions de capteur Nirxone S de spectral Engines sont le 1.7, 2.0, 2.2 et 2.5. Avec le Nirxone S2.5 on aurait une plus large bande et qui dit plus large bande de longueur d'onde dit plus d'information importante sur les pics des spectres. Les modèles ML seraient ainsi beaucoup plus performants. Et on pourrait l'utiliser à d'autres fins.

Product	Wavelength	Material type
S1.7	1.35 – 1.65 μm	Plants, Animal feeds, Seeds
S2.0	1.55 – 1.95 μm	Textile, Plastic, Spices
S2.2	1.75 – 2.15 μm	Moisture, Soil
S2.5	2.00 – 2.45 μm	Soil

Il serait également possible d'améliorer le volet visualisation de Textilus en intégrant, grace à la classe Graph de kivy_garden des graphes en temps réel tels que dans SensorControle qui affichent directement dans textilus les mesures en temps réel. Aussi à travers les diverses modules scientifiques de python tels scipy on peut intégrer dans textilus d'autres analyses plus complexe car par l'étude la spectroscopie j'ai compris qu'on peut non seulement faire de la detection mais aussi et surtout il est possible grace à certaine transformation de fourrier de detecter les molécules qu'il y a dans les matériaux et cela parfois avec les quantités. L'application pourrait donc fournir de nombreuses informations intéressantes sur la qualité des tissu et rendre le tri beaucoup plus pertinent et profond que de simples classification.

VII. Apports du stage et la perspective professionnelles

L'une des plus grande leçon que j'ai appris par la pratique à travers ce stage et dont je compte me souvenir pour le reste de ma carrière professionnelle, c'est l'importance de la data et la l'étendue énorme de possibilité que ca apporte. Mon amour pour la science des données a augmenté et mon désir de poursuivre ma formation en DataScience est plus intense. Aussi j'ai développé des competences de recherches puisque mon stage était un stage recherche. Je trouve que nous sommes dans une ère ou l'information est très disponible. nous sommes capables de tout si nous le voulons. La preuve est que je suis passé de connaissance amateurs en developpement en developpeur d'application web en moins de trois. tout celà parce que l'information est à portée de main et qu'il très facile s'autoformer pour augmenter ses compétences à travers la recherche personnelle. J'ai pu travailler sur un projet de Machine et deep learning mêlé à de la programmation informatique du début à la fin. J'ai appris beaucoup de tous les cotés. J'ai pu comprendre ce qui est important, ce sur quoi il faut passer beaucoup de temps et là ou il ne faut sutout pas perdre son temps. par exemple j'ai compris l'importantce d'avoir des données fiables et propres avant de se lancer dans l'apprentissage des modèles. En plus j'ai compris aussi que dans le mode de la programmation il ne suffit pas d'avoir que du code qui fonctionne mais il est important de savoir bien rédiger son code et connaitre les bonnes habitudes d'un data Engeneer.

VIII. Schemas, graphiques et illustrations

