

الجمهورية الشعبية الديمقراطية الجزائرية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة العليا للإعلام الآلي - 08 ماي 1945 - بسيدي بلعباس
Ecole Supérieure en Informatique
-08 Mai 1945- Sidi Bel Abbes



MEMOIRE

En Vue de l'obtention du diplôme d'**ingénieur d'état**

Filière : **Informatique**

Spécialité : **Ingénierie des Systèmes Informatiques (ISI)**

Thème

Entraîneur personnel avec reconnaissance des aliments pour l'estimation des calories

Présenté par :

- Mr Abadlia Ramy

Soutenu le : **00/07/2021**

Devant le jury composé de :

- | | |
|------------------|-------------|
| - M/Mme/Mlle XXX | Président |
| - M/Mme/Mlle XXX | Encadreur |
| - M/Mme/Mlle XXX | Examinateur |
| - M/Mme/Mlle XXX | Examinateur |

Entraîneur personnel avec reconnaissance des aliments pour l'estimation des calories

Abadlia Ramy

École supérieure en informatique (Sidi Bel Abbès)

Résumé

Ces dernières années et en raison de l'émergence d'une pensée saine, beaucoup de gens se soucient de leurs habitudes alimentaires et d'exercice mais comment faire pour que les gens surtout les cols blancs, à s'impliquer activement dans ces pratiques fatigantes est un gros problème. Notre vie quotidienne est plus pratique que jamais grâce aux grands progrès de la science et de la technologie. Cependant, le fitness et l'habitude alimentaires sont toujours ignorés dans notre vie quotidienne. Pour les aider, de nombreuses applications mobiles et plate forme web pour ont été lancées jusqu'à présent. Cependant, la plupart de ces applications présentent certains problèmes, notamment au niveau de leur facilité d'utilisation. Dans cet projet, je propose une plate-forme web qui offre un entraîneur personnel qui suivre les progrès de l'athlète et de lui donner un régime et aussi permet au athlète l'enregistrement des repas quotidiens en calculant les nutritions alimentaire à l'aide d'un model Deep Learning (DL) et on offrant un boutique online les athlète peut acheter les Compléments alimentaires facilement .

Remerciements

Tout d'abord, je tiens à remercier Dieu de m'avoir donné le courage et la patience de réaliser ce travail.

Je tiens à exprimer ma gratitude à mon superviseur, M. Belkacem Khaldi le remercier pour son aide, son orientation et ses précieux conseils.

Je tiens également à remercier mon organisme d'accueil, Artec-int, pour son accueil et son soutien et de m'avoir permis d'apprendre et de m'épanouir au sein de l'une des grandes organisations d'Annaba.

je remercier d'une manière plus particulière et exprimer mon appréciation spéciale à M. Ahmed Bendjelloul, PDG d'Artec-int et mon maître de stage, pour sa confiance et toutes les connaissances qu'il a partagées avec moi pendant et après le stage, je tiens également à Mr. Nabil, chef de projet, et à toutes les personnes avec qui j'ai eu le plaisir de travailler et d'interagir et pour m'avoir fait sentir la bienvenue dès le premier jour et avoir contribué au succès de mon stage.

Enfin, je tiens à exprimer ma profonde gratitude aux membres du jury qui ont accepté de juger ce modeste travail et qui s'apprêtent à partager avec moi leurs expériences et leurs conseils.

Abadlia Ramy

Contents

I	Introduction	1
0.1	Motivation et énoncé du problème	4
0.2	Questions de recherche	5
0.3	Objectives	6
0.4	Limitations :	6
0.5	Plan de la thèse	7
II	Contexte	9
1	Modèle d'apprentissage profond	13
1.1	Classification des images	13
1.2	L'apprentissage profond	14
1.2.1	Les différences entre l'apprentissage profond et l'apprentissage automatique	14
1.2.2	Applications de l'apprentissage profond dans l'alimentation	15
1.3	Réseaux neuronaux artificiels	16
1.4	Réseaux neuronaux convolutifs	17
1.5	Apprentissage par transfert	18
1.5.1	Utilisation d'un modèle pré-entraîné	19
1.5.2	Avantages de l'apprentissage par transfert	20
1.5.3	Les défis de l'apprentissage par transfert	20
1.6	EfficientNet	21
1.6.1	EfficientNet Architecture	21
1.6.2	Performance d'EfficientNet	22
1.6.3	EfficientNet-B0	23
2	Plate-forme Web	25
2.1	Introduction	25
2.2	Website	25
2.3	HTTP	26
2.4	Front-end	26
2.4.1	Langages de programmation front-end	26
2.5	Back-end	27
2.5.1	Langages de programmation back-end	27
2.6	Serveurs de base de données	28

2.7 Web Design	29
2.8 Responsive web design	29
2.9 API	29
2.10 MVC (Model, View, Template)	30
2.11 PWA	30
2.12 SPA	30
2.13 MPA	31
2.14 ORM	31
III Conception de la solution	33
1 Conception du modèle Deep Learning	35
1.1 Définition du problème	36
1.2 Data	37
1.3 Évaluation	38
1.4 Features	39
1.5 Modélisation	40
1.5.1 Le choix d'un modèle	40
1.5.2 Ajuster et améliorer un modèle	40
1.5.3 Comparaison des modèles	41
1.6 Expérimentation	41
2 Conception du Plate-forme Web	43
2.1 Diagramme de cas d'utilisation	43
2.2 Diagramme de classe	45
2.3 Modèle Relationnel	46
2.4 Architecture du plate-forme	47
IV Mise en œuvre de la solution	49
1 Technologies utilisées	51
1.1 Deep Learning Model	51
1.1.1 Environnement	51
1.1.2 Les bibliothèques utilisées	52
1.1.3 Les Outils	54
1.1.4 TensorBoard	55
1.2 Plate-forme Web	55
1.2.1 React Js	55
1.2.2 Django	57
1.2.3 Base de donnée	59
1.2.4 Langues	61
1.2.5 Outils	62
1.2.6 bibliothèques	63
1.2.7 Déploiement	65

2 Méthodes et résultats	67
2.1 Mise en œuvre du deep learning modal	67
2.1.1 Importation de dataset	67
2.1.2 Prétraitement des données	67
2.1.3 Création du modelling callbacks	70
2.1.4 Mise en place mixed precision training	70
2.1.5 Construire le model	71
2.1.6 Résultat	73
2.2 Mise en œuvre de la plate-forme Web	74
2.2.1 Structure du projet	74
2.2.2 Mise en œuvre des parties intéressantes des cas d'utilisation et Ecrans principaux de l'application	77
V Gestion de projet	87
1 Travail Collaboratif	89
1.1 Introduction	89
1.2 Outils collaboratifs	89
1.2.1 Google Drive	89
1.2.2 GIT	89
1.2.3 Google Colab	90
1.3 Suivi du projet	90
1.3.1 Supervision à l'ESI-Sba	90
1.3.2 Supervision à Artec Int	90
2 Méthode de gestion	93
2.1 Scrum Method	94
2.2 Diagramme de Gantt	95
3 Conclusion et travaux futurs	97
3.1 Conclusion	97
3.2 Appréciation personnelle	98
3.3 Travaux futurs	98
References	101

List of Figures

1.1	Reconnaissance des aliments	16
1.2	Tableau de la matrice RVB	17
1.3	Étapes du réseau neuronal convolutif	18
1.4	Image de transfer learning	19
1.5	Mise à l'échelle du modèle	21
1.6	EfficientNet Architecture	22
1.7	Comparaison entre la taille du modèle et sa précision.	22
1.8	Architecture d'EfficientNet-B0	23
1.1	Les étapes d'un projet machine learning	35
1.2	Food-101 Dataset	37
1.3	Confusion matrix	39
1.4	Conversion d'images en tenseurs	39
1.5	Conversion d'images en tenseurs	42
2.1	Use Case Diagram	44
2.2	Diagramme de classe	45
2.3	Architecture du plate-forme	47
1.1	La communication entre Django et React js	59
2.1	Importation de dataset avec TFDS	67
2.2	Étapes du prétraitement des données	69
2.3	Prefetching vs not using prefetching	69
2.4	Architecture de modèle	71
2.5	Création de modèle	72
2.6	Ajustement du modèle	73
2.7	Évaluation du modèle	73
2.8	Train VS Test data accuracy	74
2.9	Visualisation de loss function	74
2.10	Contenu du dossier backend	75
2.11	Contenu du dossier frontend	76
2.12	Food AI avec cas réel	78
2.13	Food AI in Django	79
2.14	Login screen design	80
2.15	Login screen in React screen	80
2.16	Authentication in React Action	81

2.17 Authentication in django	81
2.18 Shop screen design	82
2.19 Shop screen in React	82
2.20 Shop screen in Django	83
2.21 Plan-repas	83
2.22 Ajouter un repas	84
2.23 Table de tous les utilisateurs	84
2.1 Le processus de méthode Agile	94
2.2 Le processus de scrum	94
2.3 Diagramme de Gant	95
2.4 Les étapes de planification	96

List of Tables

1.1	Différences entre Machine learning et Deep learning	15
2.1	Développeur Front-end vs Back-end	28
1.1	Différences entre React.js et Plain js	57
1.2	Comparaison entre SQL et NOSQL	61
2.1	Résultat du temps par epoch en utilisant MPT	70

Abréviations

AI	Artificial Intelligence.
ANN	Artificial Neural Network.
MP	Mixed Precision.
ConvNet	Convolutional Neural Network.
DNN	Deep Neural Network.
GPU	Graphical Processing Unit.
SNN	Spike Neural Networks.
MCP	McCulloch–Pitts.
FC	fully connected,
JSON	JavaScript Object Notation.
MKL	Multiple Kernel Learning.
ML	Machine Learning.
R-CNN	Regional Convolutional Neural Network.
ReLU	Rectified Linear Unit.
ResNet	Residual Neural Network.
2D	Two-Dimensional.
AR	Augmented Reality .
RVB	Rouge vert blue.
NAS	Neural Architecture Search.
API	Application programmable Interface.
WWW	World Wide Web.
HTTP	Hypertext Transfer Protocol.
HTML	Hypertext Markup Language.
CSS	Cascading Style Sheets.
JS	JavaScript.
UI	User Interface.
UX	User Experience.
RWD	Responsive web design.
MVT	Model, View, Template.
PWA	Progressive Web App.
URL	Uniform Resource Locator.
SPA	Single Page Application.
MPA	Multiple Page Application.
AJAX	Asynchronous JavaScript And XML.
ORM	Object–Relational Mapping.
ROC	Receiver Operating Characteristic.
UML	Unified Modeling Language.
MR	Modèle Relationnel.
DRF	Django Rest Framework.
SASS	Syntactically Awesome Style Sheets.
JWT	Json Web Token.

Part I

Introduction

Encourager les gens à faire plus d'exercice est la clé pour maintenir ou le rétablissement de la santé personnelle mais malheureusement difficile à réaliser dans la pratique. L'un des obstacles à l'exercice physique est que les profanes ne sont souvent pas suffisamment informés sur les exercices physiques efficaces et sûrs. Le maintien d'un régime d'exercice à long terme exige des niveaux élevés de motivation et de temps, ce qui entre souvent en conflit avec le mode de vie actif des gens. Il est bien établi que l'accès à un entraîneur personnel a un impact significatif sur l'adhésion à un programme d'exercice physique et sur la qualité de l'exercice entrepris . Les entraîneurs personnels surveillent en permanence les exercices et à la fois fournissent des conseils individualisés et motivent l'athlète. Ils jouent également un rôle important dans la réadaptation, par exemple dans les programmes d'exercices pour la récupération musculaire après une chirurgie, où le besoin de conseils concernant l'efficacité et la sécurité est encore plus important. Malheureusement, la disponibilité des entraîneurs spécialisés est limitée . Il est également difficile de respecter le régime alimentaire requis en termes de nombre de calories, de glucides et de protéines, car parfois l'athlète ne connaît pas la valeur énergétique contenue dans les aliments suggérés par l'entraîneur. C'est pourquoi un système permettant d'enregistrer et de mesurer le nombre de calories consommées au cours d'un repas est d'une grande utilité. La prédiction précise des calories alimentaires est donc tout aussi importante dans de tels cas. Au cours des trois dernières années, les capacités de classification et de détection des objets se sont considérablement améliorées grâce aux progrès de l'apprentissage profond et des réseaux convolutifs. Exploiter cette technologie pour classer et détecter avec précision les objets alimentaires est significativement essentiel pour le Suivi correct du régime. Dans ce projet, je vais créer une plateform Web spécial pour un entraîneur personnel professionnel afin qu'il puisse proposer des programmes aux athlètes en fonction de leurs objectifs et qu'ils puissent suivre leurs progrès via la plate-forme et connaître la valeur nutritionnelle de leur alimentation, puis la comparer avec la valeur recommandée par l'entraîneur en utilisant un algorithme de reconnaissance d'images des aliments basé sur l'apprentissage profond pour améliorer la précision de l'évaluation diététique.

0.1 Motivation et énoncé du problème

L'importance de faire de l'exercice est une évidence depuis longtemps, mais comment faire pour que les gens surtout les cols blancs, à s'impliquer activement dans ces pratiques fatigantes est un gros problème. Notre vie quotidienne est plus pratique que jamais grâce aux grands progrès de la science et de la technologie. Cependant, la musculation ou le fitness sont toujours ignorés dans notre vie quotidienne. Les guides de fitness actuels comprennent des livres, des vidéos et des applications mobiles, etc... , dont la plupart ne fournissent pas de guide de remise en forme en temps réel, voire ne ou même ne permettent aucune interaction avec les utilisateurs. En général, ces produits présentent plusieurs inconvénients.C'est pour ça les gens ont besoin de trouver un entraîneur personnel pour fournir des plans et des instructions de leur propre forme physique.

Il est aussi difficile de respecter le régime alimentaire requis en termes de nombre de calories, de glucides et de protéines, car parfois l'athlète ne connaît pas la valeur énergétique contenue dans les aliments suggérés par l'entraîneur et si j'essaye d'utiliser l'apprentissage profond pour classer la nourriture dans l'image . L'un des principaux problèmes concernant la recherche des niveaux de nutrition dans une image est qu'une image est bidimensionnelle, et donc difficile de déterminer la taille de l'aliment dans l'image. Il sera donc important de trouver un moyen de mesurer la quantité d'aliments présents dans l'image. Il y aura également une différence dans la façon dont l'aliment a été préparé. Il sera plus facile de déterminer les niveaux nutritionnels d'une pomme que ceux d'une lasagne. Les plats complexes comme les lasagnes peuvent être difficiles à déterminer car beaucoup ingrédients sont cachés dans l'image, et il est donc presque impossible à déterminer. Cependant, une estimation peut être faite sur la base d'une recette connue pour ce type de plats, ce qui permet de s'en approcher. Pour les éléments alimentaires plus simples, il devrait être plus facile de déterminer les niveaux nutritionnels car l'image pourra de montrer tout ce qui est inclus dans l'image.

0.2 Questions de recherche

1. Quelles sont les technologies utilisées pour réaliser la plateforme web ?

Pour réaliser la plateforme j'utiliserais React Js pour le côté frontend avec Html,Css,Bootstrap ..., et pour le côté backend j'utiliserais Django et postgresql.

2. Comment puis-je déployer la plateforme avec le model de classification ?

Dans cette thèse, je vais déployer le model dans django et ensuite je mettrai le projet dans un conteneur docker pour contenir l'environnement virtuel qui contient tous les dépendances de django et après le déployer avec un serveur Heroku pour mettre le projet online .

3. Quels modèles de classification conviennent pour classer des images d'aliments ?

Dans cette thèse, je vais utiliser un réseau classique appelé EfficientNet afin d'obtenir des classifications d'images d'aliments .

0.3 Objectives

Mon plate-forme va veiller à maintenir un lien important entre l'entraîneur et l'athlète avec les objectifs suivants :

1. Permet également à l'entraîneur de suivre les progrès de l'athlète et de lui donner un régime.
2. Offre une boutique en ligne avec toutes les fonctionnalités qui permet à l'entraîneur de proposer des compléments nutritionnels aux sportifs.
3. Fournit également un Model Deep Learning(Food Recognition) pour analyser les images d'aliments pour sportifs et savoir ce qu'ils contiennent en nutriments(Carbs , Protéines, Calories) Pour voir s'il avait répondu à ses besoins quotidiens.

0.4 Limitations :

1. Le modèle nécessite une discrimination alimentaire très précise, une base de données nutritionnelle très diversifiée avec une analyse rapide ce qui est difficile lorsque le modèle n'est pas déployé sous sa propre serveur.
2. Lorsque l'on travaille avec l'apprentissage machine (ML), on a besoin d'une grande puissance de calcul. La formation de nouveaux modèles est très gourmande en ressources informatiques. Les grandes entreprises utilisent d'énormes centres de données pour ce type de tâches. Une solution consiste à utiliser des réseaux pré-entraînés, mais il sera toujours nécessaire de procéder à un certain entraînement j'utiliserais aussi colab pour l'entraînement du modèle avec GPU et la nouvelle technique d'entraînement à précision mixte (MP) .
3. En plus des limitations énumérées ci-dessus, le temps peut également être considéré comme une limitation. Le projet est prévu pour durer 4 mois, ce qui limite les tâches réalisables pendant cette période.

0.5 Plan de la thèse

- Part 2 : Cette part fournit les informations de base nécessaires à la compréhension de la tâche à accomplir.
- Part 3 : Cette part présente l'architecture globale de la solution puis explique la méthodologie choisie pour la conception du modèle de prédiction et les technologies utilisées pour construire la plate forme web.
- Part 4 : Après avoir présenté mon méthodologie d'analyse et de conception de ma solution logicielle, dans cette part je passe maintenant à sa réalisation.
- Part 5 : Dans cette part, je vais relater différents aspects liés à la gestion de projet. Je vais détailler le suivi du projet, notamment les outils utilisés, les interactions avec les différents superviseurs.

Part II

Contexte

Cette partie commence par exposer la théorie de base et les algorithmes qui ont été utilisés pour répondre aux questions de recherche de cette thèse. Ensuite, cette partie également fournit les informations de base nécessaires à la compréhension de la tâche à accomplir du côté de la création d'une plateforme web et du côté de la création d'un modèle d'apprentissage automatique.

Chapter 1

Modèle d'apprentissage profond

Cette chapter contient toutes les connaissances nécessaires à la création d'un modèle d'apprentissage profond, notamment dans le domaine de la classification d'images.

1.1 Classification des images

La classification d'images permet à un ordinateur d'analyser une image et d'identifier la "classe" à laquelle elle appartient. (Ou la probabilité que l'image fasse partie d'une "classe"). Une classe est essentiellement une étiquette, par exemple "voiture", "animal", "bâtiment" etc.

Par exemple, vous saisissez l'image d'un mouton. La classification des images est le processus par lequel l'ordinateur analyse l'image et vous dit qu'il s'agit d'un mouton. (Ou la probabilité que ce soit un mouton).

Pour nous, la classification d'images n'est pas un gros problème. Mais c'est un exemple parfait du paradoxe de Moravec en ce qui concerne les machines. (C'est-à-dire que les choses que nous trouvons faciles sont difficiles pour l'IA).

Les premières classifications d'images reposaient sur les données brutes des pixels. Cela signifie que les ordinateurs décomposaient les images en pixels individuels. Le problème est que deux images d'une même chose peuvent être très différentes. Elles peuvent avoir des arrière-plans, des angles, des poses différents. C'était donc un véritable défi pour les ordinateurs de "voir" et de classer correctement les images.

1.2 L'apprentissage profond

L'apprentissage profond est un sous-ensemble de l'apprentissage automatique, qui consiste essentiellement en un réseau neuronal à trois couches ou plus. Ces réseaux neuronaux tentent de simuler le comportement du cerveau humain - même s'ils sont loin d'en avoir la capacité - en lui permettant "d'apprendre" à partir de grandes quantités de données. Si un réseau neuronal à une seule couche peut toujours faire des prédictions approximatives, des couches cachées supplémentaires peuvent aider à optimiser et à affiner la précision [1].

L'apprentissage profond est à la base de nombreuses applications et services d'intelligence artificielle (IA) qui améliorent l'automatisation, en exécutant des tâches analytiques et physiques sans intervention humaine. La technologie d'apprentissage profond est à l'origine de produits et de services de tous les jours (tels que les assistants numériques, les télécommandes de télévision à commande vocale et la détection des fraudes à la carte de crédit) ainsi que de technologies émergentes (telles que les voitures à conduite autonome).

1.2.1 Les différences entre l'apprentissage profond et l'apprentissage automatique

Le Machine learning et le Deep learning font partie de l'intelligence artificielle. Ces approches ont toutes deux pour résultat de donner aux ordinateurs la capacité de prendre des décisions intelligentes. Cependant, le Deep learning est une sous-catégorie du Machine learning, car il s'appuie sur un apprentissage sans surveillance.

Le Machine learning (apprentissage automatique) est la technologie la plus ancienne et la plus simple. Elle s'appuie sur un algorithme qui adapte lui-même le système à partir des retours faits par l'humain. La mise en place de cette technologie implique l'existence de données organisées. Le système est ensuite alimenté par des données structurées et catégorisées lui permettant de comprendre comment classer de nouvelles données similaires. En fonction de ce classement, le système exécute ensuite les actions programmées. Il sait par exemple identifier si une photo montre un chien ou un chat et classer le document dans le dossier correspondant.

Après une première phase d'utilisation, l'algorithme est optimisé à partir des feedbacks du développeur, qui informent le système des classifications erronées et lui indiquent les bonnes catégories.

Le Deep learning (apprentissage profond) n'a pas besoin de données structurées. Le système fonctionne à partir de plusieurs couches de réseaux neuronaux, qui combinent

différents algorithmes en s'inspirant du cerveau humain. Ainsi, le système est capable de travailler à partir de données non structurées.

Cette approche est particulièrement adaptée pour les tâches complexes, lorsque tous les aspects des objets à traiter ne peuvent pas être catégorisés en amont. Le système du deep learning identifie lui-même les caractéristiques discriminantes. Dans chaque couche, il recherche un nouveau critère spécifique de l'objet, qui sert de base pour décider de la classification retenue pour l'objet à la fin du processus.

avec le deep learning, le système identifie lui-même les caractéristiques discriminantes des données, sans avoir besoin d'une catégorisation préalable. Le système n'a pas besoin d'être entraîné par un développeur. Il évalue lui-même le besoin de modifier le classement ou de créer des catégories inédites en fonction des nouvelles données.

Tandis que le machine learning fonctionne à partir d'une base de données contrôlable, le Deep learning a besoin d'un volume de données bien plus considérable. Le système doit disposer de plus de 100 millions d'entrées pour donner des résultats fiables.

Par ailleurs, la technologie nécessaire pour le deep learning est plus sophistiquée. Elle exige plus de ressources IT et s'avère nettement plus coûteuse que le Machine learning : elle n'est donc pas intéressante, du moins à l'heure actuelle, pour une utilisation de masse par les entreprises.

	Machine Learning	Deep Learning
Organisation des données	Données structurées	Données non structurées
Base de données	Contrôlable	> 1 million de données
Entraînement	Entraînement par l'humain nécessaire	Système d'apprentissage autonome
Algorithme	Algorithme modifiable	Réseau neuronal d'algorithmes
Champ d'application	Actions simples de routine	Tâches complexes

Table 1.1: Différences entre Machine learning et Deep learning

1.2.2 Applications de l'apprentissage profond dans l'alimentation

Les régimes et les habitudes alimentaires peuvent affecter la santé des êtres humains. Les diabétiques et les personnes allergiques, etc... , doivent surveiller et contrôler strictement leur comportement alimentaire. La reconnaissance et la classification des aliments est une tâche importante pour aider les êtres humains à enregistrer les régimes alimentaires quotidiens. Les images des aliments sont l'une des informations les plus importantes pour refléter les caractéristiques des aliments. De plus, la détection d'images est

un outil d'acquisition d'informations relativement facile et peu coûteux pour l'analyse de l'apparence des aliments. Pour les produits naturels comme les aliments et les aliments transformés, les grandes variations de forme, de volume, de texture, de couleur et de composition des aliments font de la reconnaissance des aliments une tâche difficile. Les différents arrière-plans et la disposition des aliments introduisent également des variations pour la reconnaissance et la classification des aliments. Actuellement, en raison de l'utilisation courante du CNN, l'analyse d'image est le modèle le plus utilisé pour la reconnaissance et la classification des aliments.

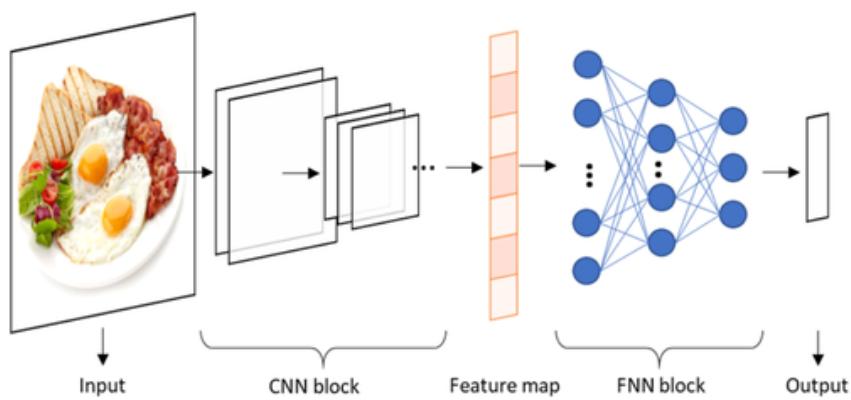


Figure 1.1: Reconnaissance des aliments
[2]

1.3 Réseaux neuronaux artificiels

Les réseaux neuronaux, également appelés réseaux neuronaux artificiels (ANN) ou réseaux neuronaux simulés (SNN), constituent un sous-ensemble de l'apprentissage automatique et sont au cœur des algorithmes d'apprentissage profond. Leur nom et leur structure sont inspirés du cerveau humain, imitant la façon dont les neurones biologiques se signalent les uns aux autres [1].

Les réseaux de neurones artificiels (ANN) sont constitués de couches de nœuds, contenant une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque nœud, ou neurone artificiel, se connecte à un autre et possède un poids et un

seuil associés. Si la sortie d'un nœud individuel est supérieure à la valeur seuil spécifiée, ce nœud est activé, envoyant des données à la couche suivante du réseau. Dans le cas contraire, aucune donnée n'est transmise à la couche suivante du réseau.

1.4 Réseaux neuronaux convolutifs

Dans les réseaux neuronaux, le réseau neuronal convolutif (ConvNets ou CNNs) est l'une des principales catégories pour faire de la reconnaissance d'images, de la classification d'images. La détection d'objets, la reconnaissance de visages sont quelques-uns des domaines où les CNN sont largement utilisés. La classification d'images par CNN prend une image d'entrée, la traite et la classe dans certaines catégories (par exemple, chien, chat, tigre, lion). Les ordinateurs voient une image d'entrée comme un tableau de pixels et cela dépend de la résolution de l'image. En fonction de la résolution de l'image, il verra $h \times w \times d$ (h = hauteur, w = largeur, d = dimension). Par exemple, une image de $6 \times 6 \times 3$ matrice de RVB (3 se réfère aux valeurs RVB) et une image de $4 \times 4 \times 1$ matrice d'échelle de gris [3].

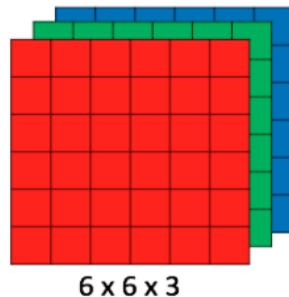


Figure 1.2: Tableau de la matrice RVB
[4]

Techniquement, les modèles CNN d'apprentissage profond pour former et tester, chaque image d'entrée passera par une série de couches de convolution avec des filtres (Kernals), Pooling, couches entièrement connectées (FC) et appliquer la fonction Softmax pour classifier un objet avec des valeurs probabilistes entre 0 et 1. La figure ci-dessous est un flux complet de CNN pour traiter une image d'entrée et classifie les objets en fonction des valeurs .

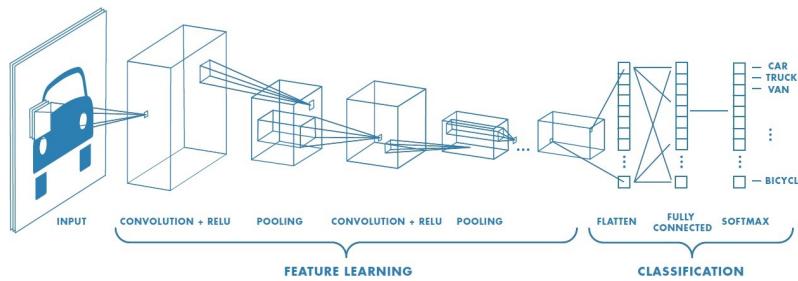


Figure 1.3: Étapes du réseau neuronal convolutif

[4]

1.5 Apprentissage par transfert

Depuis l'introduction de l'apprentissage profond, on assiste à un regain d'intérêt pour les réseaux neuronaux dans toute une série d'applications. L'apprentissage profond a permis de résoudre des problèmes considérés comme impossibles il y a à peine dix ans. Mais les réseaux neuronaux d'apprentissage profond nécessitent de grands clusters de serveurs de calcul, de grandes quantités de données d'entraînement et beaucoup de temps pour entraîner le réseau neuronal profond.

Le réseau neuronal profond est représenté par des dizaines de millions de poids qui relient entre elles les nombreuses couches de neurones du réseau. Ces poids (généralement des valeurs réelles) sont ajustés pendant le processus de formation et appliqués aux entrées (y compris les entrées des couches intermédiaires) pour alimenter une classification de sortie. L'idée de base de l'apprentissage par transfert est donc de commencer avec un réseau d'apprentissage profond qui est pré-initialisé à partir de la formation d'un problème similaire. En utilisant ce réseau, une durée de formation plus courte est nécessaire pour le nouveau problème, mais apparenté.

L'apprentissage par transfert est la méthode qui consiste à partir d'un modèle pré-entraîné et à l'entraîner pour un nouveau domaine problématique - connexe. Le réseau pré-entraîné sert de transfert de connaissances à appliquer dans un autre domaine. Mais de nombreuses options peuvent être utilisées, notamment le transfert de caractéristiques et l'ajustement fin (qui dépendent de la similarité des problèmes en question), en plus de geler certaines couches du réseau et d'en recycler d'autres.

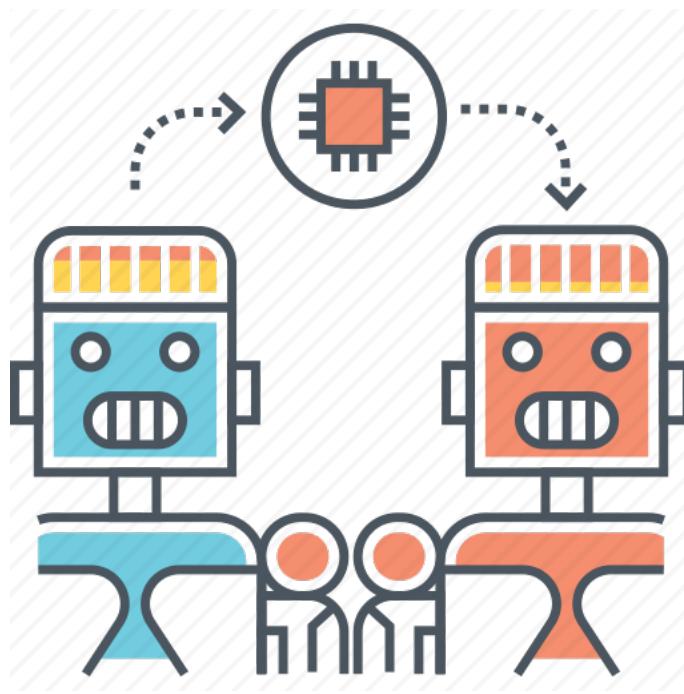


Figure 1.4: Image de transfer learning

1.5.1 Utilisation d'un modèle pré-entraîné

Avec le transfert de fonctionnalités, un modèle est formé pour un problème et réutilisé pour un problème connexe. Mais une autre option consiste à réutiliser un modèle formé par un tiers. Il existe de nombreux modèles pré-entraînés disponibles pour une variété de plateformes et de tâches, communes ou spécialisées [5] :

MobileNet est un modèle entraîné sur la base de données ImageNet (couvrant des millions d'images avec 20 000 classifications) pour l'API de réseaux neuronaux Keras fonctionnant sur Android, IOS et autres. MobileNet peut effectuer la détection d'objets, la reconnaissance de points de repère et d'autres tâches de vision par ordinateur comme la détection d'attributs de visage.

La détection d'objets est un modèle TensorFlow capable de localiser et d'identifier plusieurs objets dans une seule image. Ce modèle a été entraîné sur une variété de jeux de données d'images, y compris le jeu de données Open Images V4. Il est capable de fonctionner sur TensorFlow et TensorFlow Lite (pour les plateformes Android).

En utilisant un modèle pré-entraîné, le modèle peut être affiné par le transfert de caractéristiques (entraînement de la couche de classification) ou en gelant certaines des premières couches convolutives et en ré-entraînant les couches ultérieures. Étant donné que les premières couches convolutionnelles détectent des caractéristiques générales qui

peuvent être indépendantes du problème, il peut être utile de réentraîner (ou d'affiner) les couches convolutionnelles ultérieures où les caractéristiques sont plus spécifiques et dépendent du problème. Cette stratégie est idéale si les domaines de problèmes sont similaires (avec un travail supplémentaire soit dans le transfert de caractéristiques, soit dans le réglage fin avec des données d'entraînement supplémentaires).

1.5.2 Avantages de l'apprentissage par transfert

L'utilisation de l'apprentissage par transfert présente un grand nombre d'avantages, au-delà des économies potentielles de temps et d'énergie. L'un des principaux avantages réside dans la disponibilité d'un ensemble de formation étiqueté suffisant pour votre domaine problématique. Lorsque les données de formation sont insuffisantes, un modèle existant (d'un domaine connexe) peut être utilisé avec une formation supplémentaire pour prendre en charge le nouveau domaine.

Comme nous l'avons vu, un modèle d'apprentissage profond met en œuvre l'extraction de caractéristiques et la classification avec une topologie de réseau neuronal plus petite. Selon le domaine du problème, les sorties (ou la classification) seront généralement différentes entre deux problèmes. Pour cette raison, la couche de classification est généralement remplacée et reconstruite pour le nouveau domaine de problèmes. Cela nécessite beaucoup moins de ressources pour l'apprentissage et la validation, tout en exploitant le pipeline d'extraction de caractéristiques pré-entraînées.

1.5.3 Les défis de l'apprentissage par transfert

Les idées qui sous-tendent l'apprentissage par transfert ne sont pas nouvelles, et elles ont le potentiel de réduire le travail nécessaire pour construire des réseaux neuronaux d'apprentissage profond complexes. L'un des premiers problèmes découverts dans l'apprentissage par transfert est appelé transfert négatif. Le transfert négatif fait référence à la réduction de la précision d'un modèle d'apprentissage profond après le ré-entraînement (biologiquement, cela fait référence à l'interférence des connaissances antérieures avec le nouvel apprentissage). Cela peut être dû à une trop grande dissemblance entre les domaines problématiques ou à l'incapacité du modèle à s'entraîner pour l'ensemble de données du nouveau domaine (en plus du nouvel ensemble de données lui-même). Cela a conduit à l'élaboration de méthodes permettant d'identifier quantitativement la similarité des domaines problématiques afin de comprendre le potentiel de transfert négatif (en plus de l'apprentissage par transfert de viabilité entre domaines).

1.6 EfficientNet

EfficientNet est une architecture de réseau neuronal convolutif et une méthode de mise à l'échelle qui met uniformément à l'échelle toutes les dimensions de profondeur/largeur/résolution en utilisant un coefficient composé. Contrairement à la pratique conventionnelle qui met arbitrairement à l'échelle ces facteurs, la méthode de mise à l'échelle d'EfficientNet met uniformément à l'échelle la largeur, la profondeur et la résolution du réseau avec un ensemble de coefficients d'échelle fixes. Par exemple, si nous voulons utiliser 2^N fois plus de ressources informatiques, nous pouvons simplement augmenter la profondeur du réseau de α^N , la largeur de β^N et la taille de l'image de γ^N , où α, β, γ sont des coefficients constants déterminés par une recherche sur une petite grille sur le petit modèle original. EfficientNet utilise un coefficient composé ϕ pour mettre à l'échelle de manière uniforme la largeur, la profondeur et la résolution du réseau d'une manière fondée sur des principes.

La méthode d'échelle composée est justifiée par l'intuition que si l'image d'entrée est plus grande, alors le réseau a besoin de plus de couches pour augmenter le champ réceptif et plus de canaux pour capturer des motifs plus fins sur l'image plus grande [6].

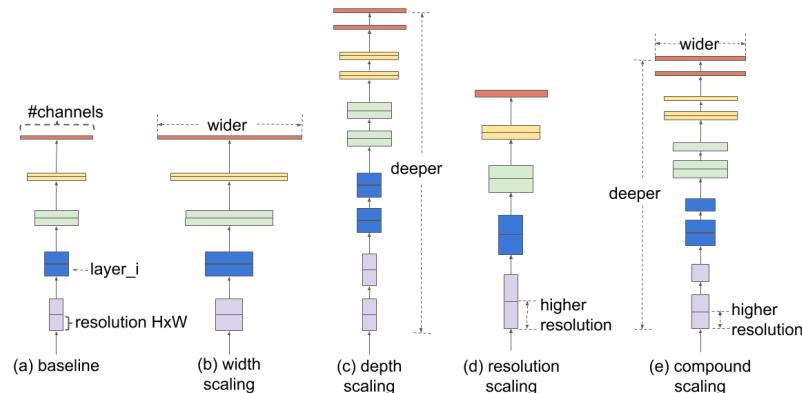


Figure 1.5: Mise à l'échelle du modèle

[6]

1.6.1 EfficientNet Architecture

L'efficacité de la mise à l'échelle du modèle dépend aussi fortement du réseau de base. Ainsi, pour améliorer encore les performances, ils ont également développé un nouveau réseau de base en effectuant une recherche d'architecture neuronale à l'aide du cadre Auto ML MNAS, qui optimise à la fois la précision et l'efficacité (FLOPS). L'architecture qui en résulte utilise la convolution mobile à goulot d'étranglement inversé (MBConv),

similaire à MobileNetV2 et MnasNet, mais elle est légèrement plus grande en raison d'un budget FLOP accru. Ils mettent ensuite à l'échelle le réseau de base pour obtenir une famille de modèles, appelés EfficientNets .

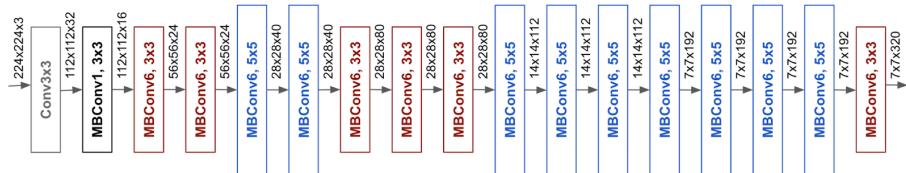


Figure 1.6: EfficientNet Architecture

[6]

1.6.2 Performance d'EfficientNet

Quand ils ont comparé EfficientNets avec d'autres CNNs existants sur ImageNet. En général, les modèles EfficientNet atteignent à la fois une plus grande précision et une meilleure efficacité par rapport aux CNNs existants, en réduisant la taille des paramètres et les FLOPS d'un ordre de grandeur. Par exemple, dans le régime de haute précision, EfficientNet-B7 atteint l'état de l'art 84,4% top-1 / 97,1% top-5 de précision sur ImageNet, tout en étant 8,4x plus petit et 6,1x plus rapide sur l'inférence du CPU que le précédent Gpipe. Comparé à ResNet-50, largement utilisé, EfficientNet-B4 utilise des FLOPS similaires, tout en améliorant la précision top-1 de 76,3% de ResNet-50 à 82,6% (+6,3%) [6].

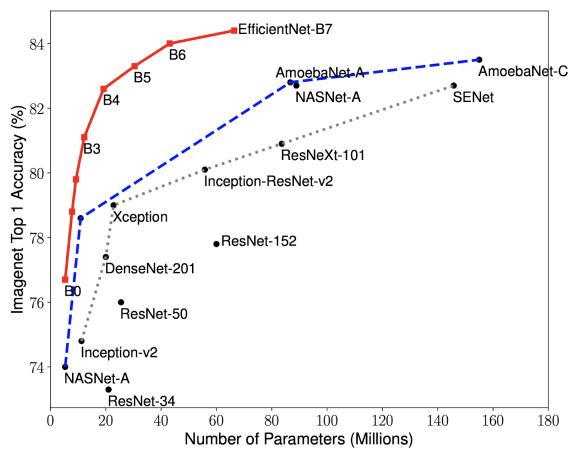


Figure 1.7: Comparaison entre la taille du modèle et sa précision.

[6]

1.6.3 EfficientNet-B0

L'un des problèmes clés de la conception des CNN, comme de tous les autres réseaux neuronaux, est la mise à l'échelle du modèle, c'est-à-dire la décision d'augmenter la taille du modèle afin d'obtenir une meilleure précision. Il s'agit d'un processus fastidieux, qui nécessite des essais manuels jusqu'à ce qu'un modèle suffisamment précis soit produit et satisfasse aux contraintes de ressources. Ce processus est gourmand en ressources et en temps et produit souvent des modèles dont la précision et l'efficacité ne sont pas optimales. Prenant ce problème en considération, les auteurs utilisent la recherche d'architecture neuronale pour construire une architecture de réseau efficace, EfficientNet-B0. Elle atteint une précision de 77,3 % sur ImageNet avec seulement 5,3 millions de paramètres et 0,39 milliard de FLOPS. (Resnet-50 fournit une précision de 76 % avec 26 millions de paramètres et 4,1 milliards de FLOPS). Le principal élément constitutif de ce réseau est le MBConv, auquel est ajoutée l'optimisation par compression et excitation. MBConv est similaire aux blocs résiduels inversés utilisés dans MobileNet v2, qui forment une connexion raccourcie entre le début et la fin d'un bloc convolutif. Les cartes d'activation d'entrée sont d'abord étendues en utilisant des convolutions 1x1 pour augmenter la profondeur des cartes de caractéristiques. Viennent ensuite les convolutions 3x3 en profondeur et les convolutions ponctuelles qui réduisent le nombre de canaux dans la carte des caractéristiques de sortie. Les connexions de raccourci relient les couches étroites tandis que les couches plus larges sont présentes entre les connexions de saut. Cette structure permet de réduire le nombre total d'opérations requises ainsi que la taille du modèle.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 1.8: Architecture d'EfficientNet-B0

[6]

Chapter 2

Plate-forme Web

Cette chapitre explique les connaissances nécessaires pour créer une plate-forme Web contemporaine selon les technologies modernes

2.1 Introduction

Écrire un millier de lignes de code et les transformer en un site web est l'une des choses les plus créatives et compliquées pour les développeurs web. Si vous êtes excité à la vue de nombreux sites web magnifiques et que vous souhaitez vous y mettre, nous devons vous ouvrir les yeux sur certaines choses importantes que vous devez savoir en tant que développeur web. La création d'un site Web qui attire l'attention des utilisateurs ne se limite pas à l'apprentissage de divers langages de programmation, vous devez également apprendre d'autres concepts tels que les outils de développement, les formats de données, les tests, les API, l'authentification et bien d'autres choses encore, une fois que vous vous serez lancé dans ce domaine. Nous allons couvrir ici certaines choses importantes que vous devriez apprendre en tant que développeur web.

2.2 Website

Les sites web sont des fichiers stockés sur des serveurs, qui sont des ordinateurs qui hébergent (terme sophistiqué pour "stocker des fichiers ") des sites web. Ces serveurs sont connectés à un réseau géant appelé Internet... ou World Wide Web "www" (si l'on s'en tient à la terminologie des années 90).

Les navigateurs sont des programmes informatiques qui chargent les sites web via notre connexion internet, comme Google Chrome ou Internet Explorer. Notre ordinateur est également appelé le client.

2.3 HTTP

Le protocole de transfert hypertexte (HTTP) vous relie, vous et votre demande de site web, au serveur distant qui héberge toutes les données du site. C'est un ensemble de règles (un protocole) qui définit la manière dont les messages doivent être envoyés sur l'internet. Il nous permet de sauter entre les pages d'un site et les sites web.

Lorsque nous tapons un site web dans notre navigateur ou que nous recherchons quelque chose via un moteur de recherche, le protocole HTTP fournit un cadre permettant au client (ordinateur) et au serveur de parler le même langage lorsqu'ils se transmettent des demandes et des réponses sur l'internet. Il s'agit essentiellement du traducteur entre vous et l'internet : il lit notre demande de site web, lit le code renvoyé par le serveur et le traduit pour nous sous la forme d'un site web.

2.4 Front-end

Le front-end (ou côté client) est le côté d'un site web ou d'un logiciel que nous voyons et avec lequel nous interagissons en tant qu'internaute. Lorsque les informations d'un site web sont transférées d'un serveur à un navigateur, les langages de codage front-end permettent au site web de fonctionner sans avoir à "communiquer" continuellement avec l'internet.

Le code front-end permet aux utilisateurs comme vous et moi d'interagir avec un site web et de lire des vidéos, d'agrandir ou de réduire des images, de mettre du texte en surbrillance, etc. Les développeurs Web qui travaillent sur le codage front-end travaillent sur le développement côté client.

2.4.1 Langages de programmation front-end

Les langages front-end comprennent HTML, CSS et Javascript. Bien que JQuery soit en train de passer de mode (les navigateurs modernes peuvent maintenant faire le même travail, mais beaucoup plus rapidement que jQuery), de nombreux projets anciens utilisent encore la bibliothèque JavaScript, ne soyez donc pas surpris de la voir au programme d'un bootcamp. Vous apprendrez également beaucoup sur le responsive design, la typographie, la mise en page, le système de grille et la théorie des couleurs. Lorsque vous anticiperez les types de projets sur lesquels vous travaillerez en tant que développeur front-end, pensez à la création et à la refonte de sites Web. Pour être un développeur front-end (parfois même appelé développeur Javascript), vous n'avez pas besoin de compétences en développement back-end. Les sites créés par les développeurs front-end n'interagissent pas avec

les informations stockées dans une base de données pour être fonctionnels. Le contenu sera "fixe", ce qui signifie que de grandes quantités de nouvelles données ne seront pas constamment téléchargées. Les propriétaires de petites entreprises et les restaurants ont généralement d'excellents exemples de sites statiques.

2.5 Back-end

Le développement Back-end fait référence au côté serveur du développement où vous vous concentrez principalement sur le fonctionnement du site. Votre principale responsabilité sera d'effectuer des mises à jour et des modifications, ainsi que de surveiller la fonctionnalité du site. Ce type de développement web se compose généralement de trois parties : un serveur, une application et une base de données. Le code écrit par les développeurs Back-end est ce qui communique les informations de la base de données au navigateur. Tout ce que vous ne pouvez pas voir facilement à l'œil nu, comme les bases de données et les serveurs, est le travail d'un développeur back-end.

2.5.1 Langages de programmation back-end

De nombreux développeurs back-end connaissent les langages front-end tels que HTML et CSS, mais doivent utiliser des langages tels que Java, PHP, Ruby on Rails, Python et .Net pour effectuer le travail back-end. Les développeurs Back-end se concentrent surtout sur la réactivité et la vitesse d'un site. Ces langages sont utilisés pour créer des sites dynamiques, qui se distinguent des sites statiques par le fait que ces types de sites stockent des informations dans des bases de données. Le contenu du site est constamment modifié et mis à jour. Facebook, Twitter et Google Maps sont des exemples de sites dynamiques.

Paramètre	Frontend	Backend
Compétences	Les langages qu'un développeur web front-end doit connaître sont HTML, CSS et JavaScript.	La base de données, le serveur et l'API, etc... .
Équipe	Les développeurs front-end conçoivent l'apparence du site Web en prenant les entrées des utilisateurs et en les modifiant par le biais de tests.	Les développeurs back-end développent une application qui prend en charge le front-end. Il doit également fournir un support, une sécurité et une gestion de contenu.
Service autonome	Le service de développement front-end ne peut pas être proposé indépendamment.	Le développement back-end peut être proposé en tant que service indépendant sous forme de BaaS (Back-end as a service)
But	Les développeurs front-end doivent s'assurer que le site Web est accessible à tous les utilisateurs et qu'il reste réactif dans toutes les vues - mobile et bureau.	L'équipe back-end peut avoir besoin de créer une application autour du front-end et de la prendre en charge. De plus, ils doivent s'assurer que le site Web s'ouvre et fonctionne correctement.
Deux outils les plus populaires	jQuery, HTML5	MySQL, PHP

Table 2.1: Développeur Front-end vs Back-end

2.6 Serveurs de base de données

La quasi-totalité de la technologie en ligne repose entièrement sur les données. Vous pouvez avoir un excellent site Web développé et conçu de manière à être facile à utiliser par n'importe qui en ligne, mais vous devez vous arrêter et penser à l'endroit où iront toutes les informations recueillies par le site ainsi que les données des clients.

C'est là que le serveur de base de données entre en action. Tout contenu ou fichier qui doit être stocké se trouve sur le serveur de base de données, qui fait partie du plan de développement de tout site web. Cela garantit que toutes les données sont sécurisées et organisées.

2.7 Web Design

Une vaste catégorie qui englobe tout ce qui a trait à la conception des éléments visuels et de la convivialité d'un site web. La conception d'interface utilisateur et d'interface utilisateurs, ainsi que de nombreux autres domaines, sont inclus dans la conception web.

Interface utilisateur (IU) : Une spécialisation de la conception Web qui traite des contrôles que les gens utilisent pour interagir avec un site Web ou une application, y compris l'affichage des boutons et les contrôles gestuels.

Expérience de l'utilisateur (UX) : Une autre spécialisation de la conception Web, celle-ci traitant du comportement et du ressenti de l'utilisateur lors de l'utilisation du site ou de l'application. La conception UX englobe de nombreux autres domaines, mais les considère du point de vue de l'utilisateur.

2.8 Responsive web design

Un site web réactif (RWD) est un site web dont la conception vise, grâce à différents principes et techniques, à offrir une consultation confortable sur des écrans de tailles très différentes. L'utilisateur peut ainsi consulter le même site web à travers une large gamme d'appareils (moniteurs d'ordinateur, smartphone, tablettes, télévision...) avec le même confort visuel et sans avoir recours au défilement horizontal ou au zoom avant/arrière sur les appareils tactiles notamment, manipulations qui peuvent parfois dégrader l'expérience utilisateur, tant en lecture qu'en navigation. Un site web réactif est un exemple de plasticité des interfaces.

2.9 API

Les API permettent à votre produit ou service de communiquer avec d'autres produits et services sans connaître les détails de leur mise en œuvre. Elles simplifient le développement d'applications et vous font ainsi gagner du temps et de l'argent. Lorsque vous concevez de nouveaux outils et produits, ou que vous assurez la gestion de ceux qui existent déjà, les API vous offrent plus de flexibilité, simplifient la conception, l'administration et l'utilisation, et vous donnent les moyens d'innover.

Les API sont parfois considérées comme des contrats, avec une documentation qui constitue un accord entre les parties : si la partie 1 envoie une requête à distance selon une structure particulière, le logiciel de la partie 2 devra répondre selon les conditions définies.

Une API REST (également appelée API RESTful) est une interface de programmation d'application (API ou API web) qui respecte les contraintes du style d'architecture REST et permet d'interagir avec les services web RESTful. L'architecture REST (Representational State Transfer) a été créée par l'informaticien Roy Fielding.

2.10 MVC (Model, View, Template)

Le MVT (Model View Template) est un modèle de conception de logiciel. Il s'agit d'une collection de trois composants importants : Modèle, Vue et Template. Le modèle permet de gérer la base de données. Il s'agit d'une couche d'accès aux données qui traite les données.

Le template est une couche de présentation qui gère entièrement l'interface utilisateur. La vue est utilisée pour exécuter la logique métier et interagir avec un modèle pour transporter les données et rendre un template.

2.11 PWA

Une progressive web app est une application web qui consiste en des pages ou des sites web, et qui peuvent apparaître à l'utilisateur de la même manière que les applications natives ou les applications mobiles. Ce type d'applications tente de combiner les fonctionnalités offertes par la plupart des navigateurs modernes avec les avantages de l'expérience offerte par les appareils mobiles.

Une PWA se consulte comme un site web classique, depuis une URL sécurisée mais permet une expérience utilisateur similaire à celle d'une application mobile, sans les contraintes de cette dernière (soumission aux App-Stores, utilisation importante de la mémoire de l'appareil...).

Elles proposent de conjuguer rapidité, fluidité et légèreté tout en permettant de limiter considérablement les coûts de développement : plus besoin de faire des développements spécifiques pour les applications en fonction de chacune des plateformes : IOS, Android...

2.12 SPA

Une application monopage (SPA) est une application web ou un site web qui interagit avec l'utilisateur en réécrivant dynamiquement la page web actuelle avec de nouvelles données provenant du serveur web, au lieu de la méthode par défaut du navigateur web qui charge

de nouvelles pages entières. L'objectif est d'obtenir des transitions plus rapides qui font que le site Web ressemble davantage à une application native.

Dans un SPA, il n'y a jamais de rafraîchissement de la page ; au lieu de cela, tout le code HTML, JavaScript et CSS nécessaire est soit récupéré par le navigateur en un seul chargement de page, soit les ressources appropriées sont chargées dynamiquement et ajoutées à la page si nécessaire, généralement en réponse aux actions de l'utilisateur. La page ne se recharge à aucun moment du processus et ne transfère pas non plus le contrôle à une autre page, bien que le hachage de l'emplacement ou l'API d'historique HTML5 puissent être utilisés pour fournir la perception et la navigabilité de pages logiques distinctes dans l'application.

2.13 MPA

Les applications à pages multiples fonctionnent de manière "traditionnelle". Chaque changement, par exemple l'affichage des données ou le renvoi des données au serveur, nécessite le rendu d'une nouvelle page du serveur dans le navigateur. Ces applications sont grandes, plus grandes que les SPA parce qu'elles doivent l'être. En raison de la quantité de contenu, ces applications ont de nombreux niveaux d'interface utilisateur. Heureusement, ce n'est plus un problème. Grâce à AJAX, nous n'avons plus à nous soucier du fait que les applications volumineuses et complexes doivent transférer beaucoup de données entre le serveur et le navigateur. Cette solution s'améliore et permet de rafraîchir uniquement certaines parties de l'application. D'un autre côté, elle ajoute plus de complexité et est plus difficile à développer qu'une application à page unique.

2.14 ORM

Un mapping objet-relationnel (ORM) est un type de programme informatique qui se place en interface entre un programme applicatif et une base de données relationnelle pour simuler une base de données orientée objet. Ce programme définit des correspondances entre les schémas de la base de données et les classes du programme applicatif. On pourrait le désigner par là, « comme une couche d'abstraction entre le monde objet et monde relationnel ». Du fait de sa fonction, on retrouve ce type de programme dans un grand nombre de frameworks sous la forme de composant ORM qui a été soit développé, soit intégré depuis une solution externe comme : Django.

Part III

Conception de la solution

Chapter 1

Conception du modèle Deep Learning

Dans ce chapitre, je vais parler de la conception de la solution derrière le modèle d'apprentissage profond de classification pour obtenir le meilleur résultat.

Un pipeline d'apprentissage automatique peut être décomposé en trois grandes étapes. La collecte des données, la modélisation des données et le déploiement. Toutes ces étapes s'influencent mutuellement.

Nous pouvons commencer un projet en collectant des données, les modéliser, vous rendre compte que les données que nous avons collectées étaient mauvaises, retourner à la collecte de données, les modéliser à nouveau, trouver un bon modèle, le déployer, constater qu'il ne fonctionne pas, faire un autre modèle, le déployer, constater qu'il ne fonctionne pas à nouveau, retourner à la collecte de données. C'est un cycle...

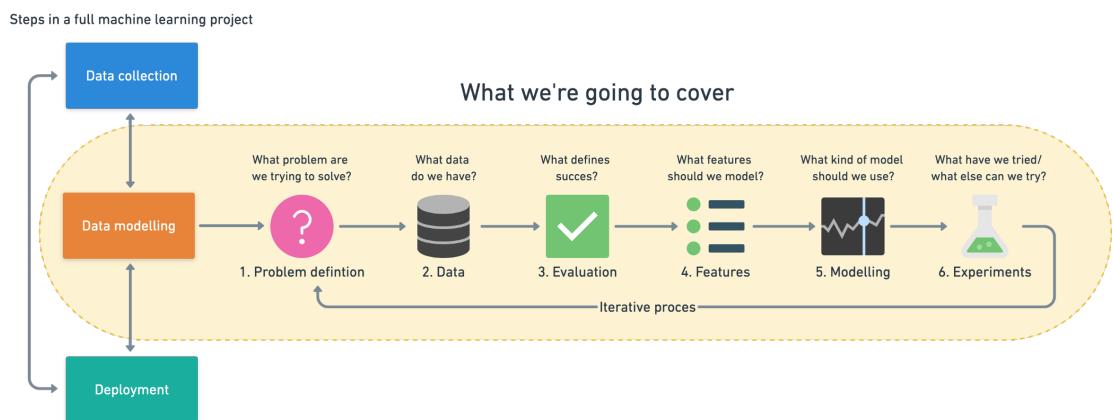


Figure 1.1: Les étapes d'un projet machine learning

Ainsi, comme nous le voyons dans ce chapitre, nous allons couvrir les 6 étapes pour obtenir le meilleur résultat avec le modèle de classification alimentaire par apprentissage profond :

1. Définition du problème : le problème nous essayons de résoudre et comment le formuler en tant que problème d'apprentissage automatique .
2. Données : Si l'apprentissage automatique permet d'obtenir des informations à partir de données, quelles sont les données dont nous disposons ? Comment correspondent-elles à la définition du problème ? Nos données sont-elles structurées ou non structurées ? Statiques ou en flux ?
3. Évaluation : Qu'est-ce qui définit le succès ? Un modèle d'apprentissage automatique précis à 95 % est-il suffisant ?
4. Caractéristiques : Quelles parties de nos données allons-nous utiliser pour notre modèle ? Comment ce que nous savons déjà peut-il influencer cela ?
5. Modélisation : Quel modèle choisir ? Comment pouvez-vous l'améliorer ? Comment le comparer à d'autres modèles ?
6. Expérimentation : Que pourrions-nous essayer d'autre ? Le modèle que nous avons déployé répond-il à nos attentes ? Comment les autres étapes changent-elles en fonction de ce que nous avons découvert ?

Plongeons un peu plus profondément dans chacun d'eux pour mon modèle.

1.1 Définition du problème

Pour décider si mon projet peut ou non utiliser l'apprentissage profond, la première étape consiste à faire correspondre le problème que j'essaie de résoudre à un problème d'apprentissage automatique .

Les quatre principaux types d'apprentissage automatique sont l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé, l'apprentissage par transfert et l'apprentissage par renforcement. Les trois plus utilisés dans les applications commerciales sont l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par transfert et nous pouvons les décomposer davantage en classification, régression et recommandation.

Dans mon modèle, je vais utiliser l'apprentissage par transfert parce que je m'occupe d'un modèle de classification computer vision qui nécessite des ressources puissantes et la formation d'un modèle d'apprentissage automatique à partir de zéro peut être coûteuse et prendre du temps. Ainsi, lorsque les algorithmes d'apprentissage automatique trouvent

des modèles dans un type de données, ces modèles peuvent être utilisés dans un autre type de données.

C'est pourquoi j'ai choisi d'utiliser EfficientNet B0 pour mon problème parce qu'il est déjà pré-entraîné avec d'énormes images et tout ce dont j'ai besoin est de reproduire les entrées et les sorties pour qu'il classifie les aliments dans l'image.

1.2 Data

Afin de reconnaître de la nourriture dans une image, nous devons d'abord entraîner le modèle avec un jeu de données, c'est pourquoi j'ai choisi le dataset FOOD-101 qui est un jeu de données plus important qui comprend 101 catégories principales basées sur les plats les plus populaires. Chacune de ces catégories comporte 1000 images de nourriture, ce qui donne au dataset un total de 101 000 images. Le dataset n'a pas été nettoyé et peut donc contenir des images avec des étiquettes erronées.

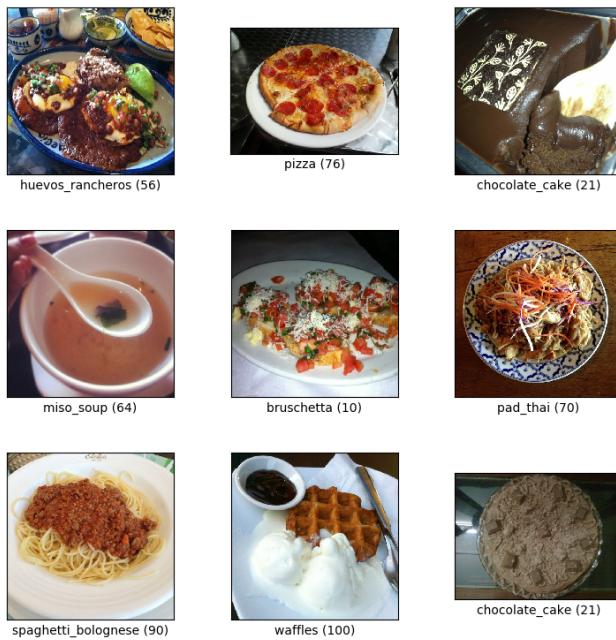


Figure 1.2: Food-101 Dataset

1.3 Évaluation

J'ai défini mon problème en termes d'apprentissage machine et j'ai un jeu de données. j'ai le dataset. Maintenant, qu'est-ce qui définit le succès.

Il existe différentes mesures d'évaluation pour les problèmes de classification, de régression et de recommandation. Celle que nous choisirons dépendra de notre objectif. Dans mon cas, il s'agit d'un problème de classification et je dois donc utiliser des méthodes spécifiques :

- Faux négatifs : le modèle prédit un négatif, en réalité un positif. .
- Faux positifs : le modèle prédit un résultat positif, alors qu'il est en réalité négatif.
- Vrais négatifs : le modèle prédit un négatif, il est en fait négatif.
- Vrais positifs : le modèle prédit un positif, réellement positif.
- Précision : Quelle est la proportion de prédictions positives qui étaient réellement correctes ? Un modèle qui ne produit aucun faux positif a une précision de 1,0.
- Rappel : Quelle est la proportion de vrais positifs qui ont été prédits correctement ? Un modèle qui ne produit aucun faux négatif a un rappel de 1,0.
- Score F1 : Une combinaison de la précision et du rappel. Plus il est proche de 1,0, mieux c'est.
- Courbe ROC (Receiver operating characteristic) et aire sous la courbe (AUC) : la courbe ROC est un graphique comparant le taux de vrais positifs et de faux positifs. La métrique AUC est l'aire sous la courbe ROC. Un modèle dont les prédictions sont fausses à 100 % a une AUC de 0,0, un modèle dont les prédictions sont justes à 100 % a une AUC de 1,0.

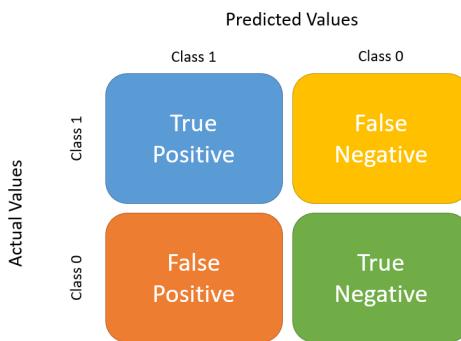


Figure 1.3: Confusion matrix

1.4 Features

Comme nous le savons, toutes les données ne sont pas les mêmes. Et lorsque nous entendons quelqu'un parler de caractéristiques, il s'agit de différents types de données au sein des données.

Les trois principaux types de caractéristiques sont catégoriques, continues (ou numériques) et dérivées.

Dans notre cas, les images peuvent également constituer une caractéristique. Quoi qu'il en soit, elles sont transformées en tableau de nombres avant qu'un algorithme d'apprentissage automatique puisse les modéliser et nous appelons ce tableau tenseur comme nous le voyons dans la figure suivante :

color image is 3rd-order tensor

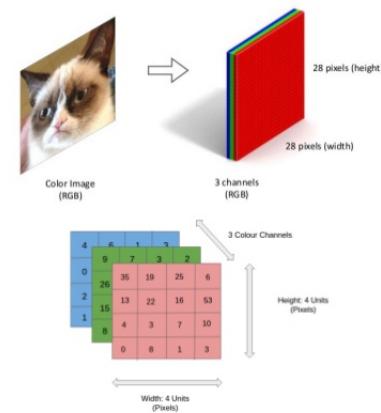


Figure 1.4: Conversion d'images en tenseurs

et lorsqu'il s'agit de fonctionnalités, nous devons nous souvenir que Gardez-les identiques pendant l'expérimentation (training) et la production (test), car un modèle d'apprentissage automatique doit être entraîné sur des caractéristiques qui représentent le plus fidèlement possible ce à quoi il sera utilisé dans un système réel.

1.5 Modélisation

Une fois que j'ai défini le problème, préparé les données, les critères d'évaluation et les caractéristiques, il est temps maintenant de modéliser.

Je vais diviser la phase de modélisation en trois parties : choisir un modèle, améliorer un modèle, le comparer à d'autres.

1.5.1 Le choix d'un modèle

Lors du choix du modèle, je prendrai en considération l'interprétabilité et la facilité de débogage, la quantité de données, les limites d'entraînement et de prédiction.

Interprétabilité et facilité de débogage : je devrais d'abord réfléchir à la raison pour laquelle le modèle a fait ces prédictions et comment les erreurs peuvent être corrigées.

Quantité de données et limites de l'entraînement : je dois donc aussi penser à la quantité de données, de temps et de ressources dont nous disposons pour la formation et la prédiction. Comme je l'ai mentionné dans la section précédente, j'ai 101000 exemples dans le dataset, ce qui consomme beaucoup de ressources, en particulier avec le modèle d'apprentissage par transfert, car il a beaucoup de paramètres. C'est pourquoi j'utilise EfficientNetB0, qui n'a que 4M de paramètres, avec une bonne précision par rapport à EfficientNetB7 ou l'autre.

1.5.2 Ajuster et améliorer un modèle

Après avoir choisi le modèle, je dois l'améliorer en modifiant les hyperparamètres tels que le taux d'apprentissage ou l'optimiseur. Ou des facteurs d'architecture spécifiques au modèle, tels que le type de couches pour l'architecture de mes réseaux neuronaux. Mais l'utilisation d'un modèle pré-entraîné "EfficientNet B0" par apprentissage par transfert présente l'avantage supplémentaire que toutes ces étapes ont été effectuées pour les couches cachées, de sorte que je n'ai plus qu'à me concentrer sur les couches d'entrée et de sortie.

1.5.3 Comparaison des modèles

La dernière étape de la phase de modélisation consiste à comparer les résultats des modèles et à choisir le meilleur, en utilisant tensorBoard

1.6 Expérimentation

Maintenant on a dans l'étape de l'expérimentation, cette étape implique toutes les autres étapes. L'apprentissage automatique étant un processus hautement itératif, je dois m'assurer que mes expériences sont exploitables. Toutes les expériences doivent être menées sur différentes parties des données.

Dataset d'apprentissage : j'utilise cet ensemble pour l'apprentissage du modèle, 70% des données .

Dataset de validation : j'utilise ce dataset pour la mise au point du modèle, 15% des données .

Dataset de test : J'utilise ce dataset pour le test et la comparaison des modèles, 15% des données .

Ensuite, je définis un ensemble de variables expérimentales qui comprend une liste de variables que je fais varier sur un certain nombre d'essais. Ces variables peuvent être des hyperparamètres tels que la taille du lot, le taux d'apprentissage, les optimiseurs, ou les architectures de modèle ou tout autre facteur qui peut avoir un effet sur la réponse, puis je vais boucler à travers les ensembles et créer un essai et un travail de formation pour chaque ensemble, enfin je peux analyser les résultats de l'expérience et choisir le meilleur.

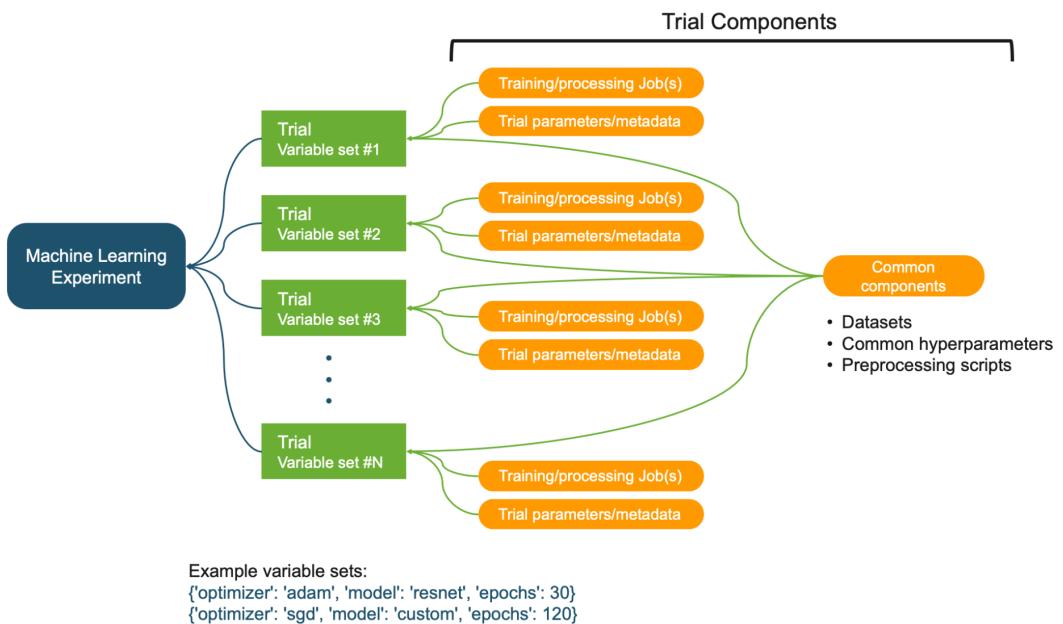


Figure 1.5: Conversion d’images en tenseurs

Chapter 2

Conception du Plate-forme Web

L'objectif de ce chapitre est de finaliser la conception de la plate-forme web en utilisant les outils UML . L'étude comporte ainsi la réalisation de plusieurs diagrammes : diagramme use case , diagramme de class , diagramme de déploiement et le modèle relationnel .

2.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est une représentation graphique des interactions possibles d'un utilisateur avec un système. Un diagramme de cas d'utilisation montre différents cas d'utilisation et différents types d'utilisateurs du système et sera souvent accompagné d'autres types de diagrammes. Les cas d'utilisation sont représentés par des cercles ou des ellipses. Les acteurs sont souvent représentés par des bâtons.

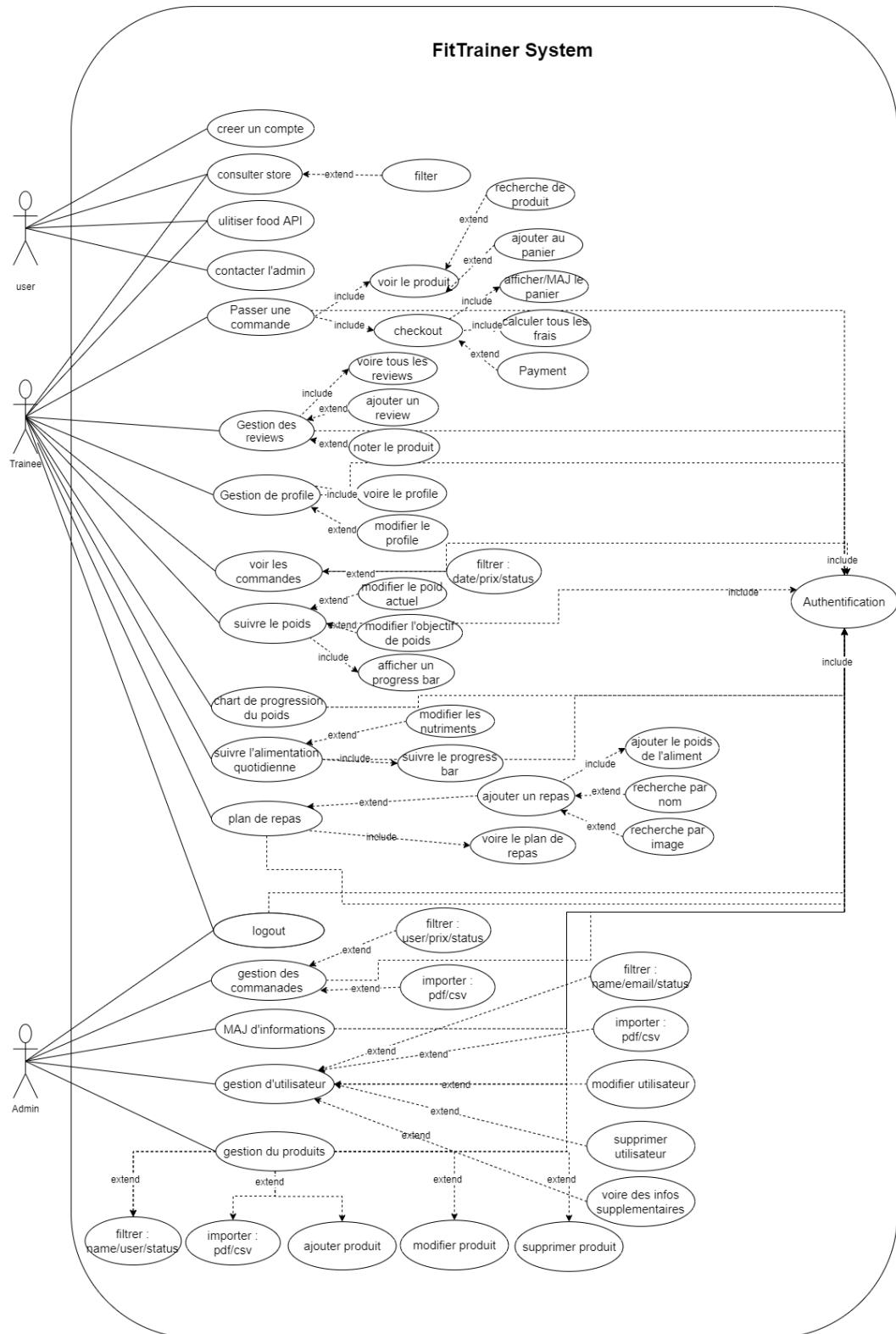


Figure 2.1: Use Case Diagram

2.2 Diagramme de classe

Un diagramme de classes est un type de diagramme et une partie d'un langage de modélisation unifié (UML) qui définit et fournit la vue d'ensemble et la structure d'un système en termes de classes, d'attributs et de méthodes, ainsi que les relations entre les différentes classes.

Il est utilisé pour illustrer et créer un diagramme fonctionnel des classes du système et sert de ressource de développement du système dans le cadre du cycle de vie du développement logiciel.

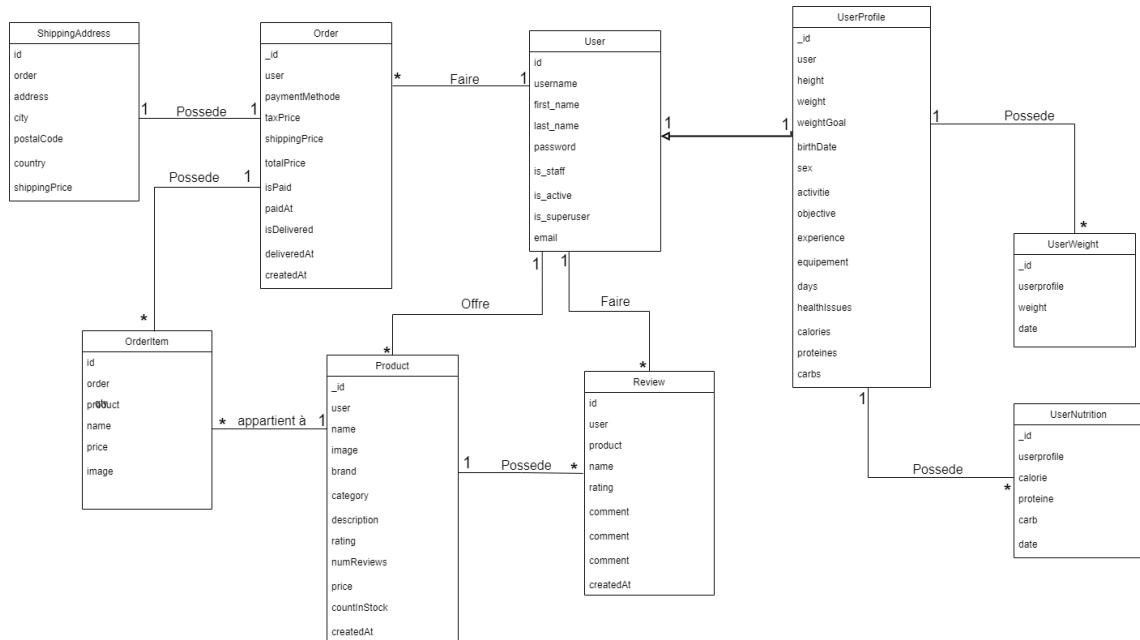


Figure 2.2: Diagramme de classe

2.3 Modèle Relationnel

Le modèle relationnel (MR) représente la base de données comme une collection de relations. Une relation n'est rien d'autre qu'un tableau de valeurs. Chaque ligne de la table représente une collection de valeurs de données liées. Ces lignes de la table dénotent une entité ou une relation du monde réel.

Le nom de la table et les noms des colonnes sont utiles pour interpréter la signification des valeurs de chaque ligne. Les données sont représentées comme un ensemble de relations. Dans le modèle relationnel, les données sont stockées sous forme de tableaux. Cependant, le stockage physique des données est indépendant de la manière dont les données sont organisées logiquement.

User (id,username,firstName,lastName,password,isStaff,isActive,isSuperuser,email)

UserProfile (id,#user_id,height,weight,weightGoal,birthDate,sex,activitie,objective,experience,equipement,days,healthIssues,calories,proteines,carbs)

UserWeight (id,#userprofile_id,weight,date)

UserNutrition (id,#userprofile_id,calorie,proteine,carb,date)

Product (id,#user_id,name,image,brand,category,description,rating,numRevies,price,countInStock,createdAt)

Order (id,#user_id,paymentMethode,taxPrice,shippingPrice,totalPrice,isPaid,isDelivered,paidAt,deliveredAt,createdAt)

OrderItem (id,#order_id,#product_id,name,qty,price,image)

ShippingAddress (id,#order_id,address,city,postalCode,country,shippingPrice)

Review (id,#user_id,#product_id,name,rating,comment,createdAt)

2.4 Architecture du plate-forme

FitTrainer est une application pleine page. J'aimerais construire ce kit de démarrage basé sur l'architecture d'une application web pleine page. L'architecture de FitTrainer utilise ce type de technologie, React + Flux (Redux) + django REST framework + RDBMS(Postgresql). Cette pile est également présentée sous forme de diagramme.

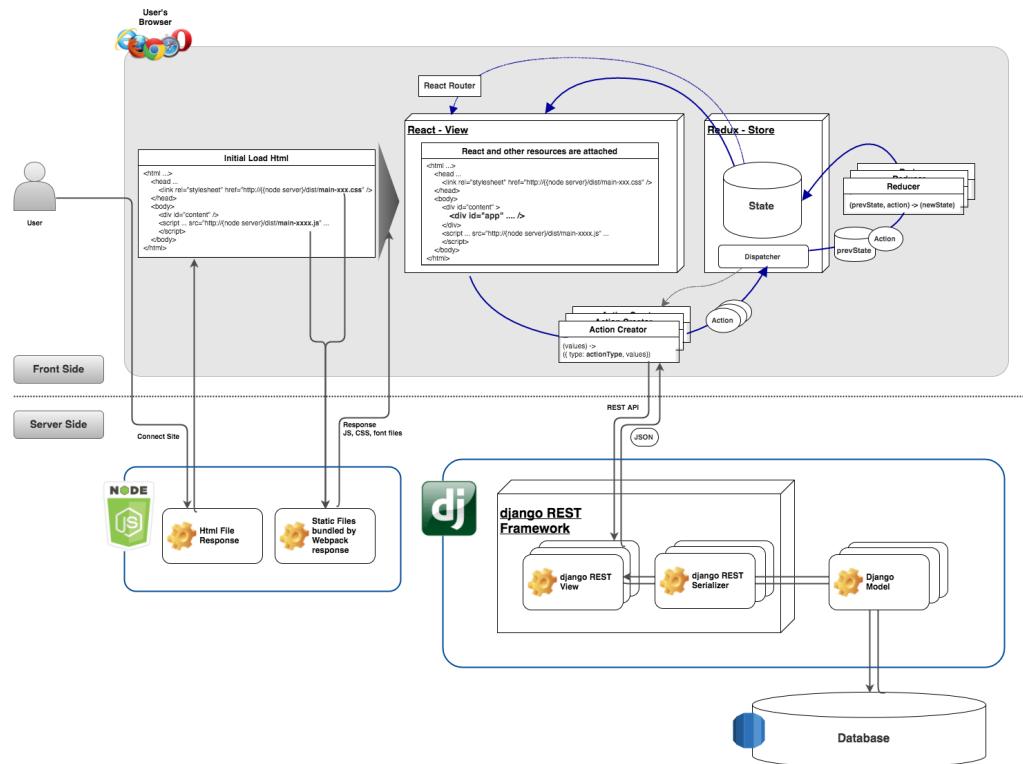


Figure 2.3: Architecture du plate-forme

Part IV

Mise en œuvre de la solution

Chapter 1

Technologies utilisées

Après avoir présenté ma méthodologie d'analyse et de conception de ma solution, je passe maintenant à sa réalisation.

Je vais d'abord présenter les différentes technologies utilisées pour réaliser le modèle de classification alimentaire par apprentissage profond et les différents technologies utilisées pour construire la plateforme web, la tester et la déployer.

1.1 Deep Learning Model

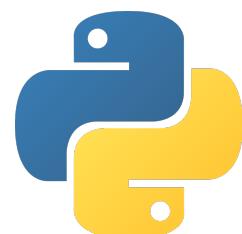
1.1.1 Environnement

Pour effectuer les différents tests et réaliser le développement des modules mentionnés dans le chapitre précédent, j'ai dû utiliser plusieurs technologies. Mon choix s'est porté sur des technologies largement utilisées dans le domaine de la science des données et facilitant le prototypage.

1.1.1.1 Python

Python est devenu en quelques années un langage majeur dans le monde des applications centrées sur le traitement des données. Il s'agit d'un langage orienté objet permettant d'exprimer des concepts très complexes en quelques lignes de code.

Et facilite donc grandement le prototypage, ce qui explique sa popularité et son utilisation dans l'apprentissage automatique.



1.1.2 Les bibliothèques utilisées

1.1.2.1 TensorFlow 2.0

Dans mon modèle, j'ai utilisé TensorFlow qui est l'une des meilleures bibliothèques disponibles pour travailler avec l'apprentissage automatique sur Python proposée par Google, TensorFlow facilite la création de modèles ML pour les débutants comme pour les professionnels.

Avec TensorFlow, vous pouvez créer et entraîner des modèles ML non seulement sur des ordinateurs, mais aussi sur des appareils mobiles et des serveurs en utilisant TensorFlow Lite et TensorFlow Serving qui offrent les mêmes avantages [7], mais pour les plateformes mobiles et les serveurs haute performance.



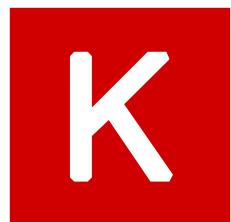
1.1.2.2 TensorFlow Hub

Comme je l'ai mentionné dans le chapitre précédent, j'ai utilisé l'aspect de l'apprentissage par transfert avec TensorFlow Hub qui est un référentiel de modèles d'apprentissage automatique formés, prêts à être optimisés et déployés partout [7]. Vous pouvez réutiliser des modèles formés comme BERT et Faster R-CNN et EfficientNet comme on a fait dans ce projet avec seulement quelques lignes de code.



1.1.2.3 Keras

Keras est l'une des bibliothèques de réseaux neuronaux les plus populaires et les plus libres pour Python. Conçue à l'origine par un ingénieur de Google pour ONEIROS, abréviation de Open-Ended Neuro Electronic Intelligent Robot Operating System, Keras a rapidement été prise en charge par la bibliothèque centrale de TensorFlow, ce qui la rend accessible au-dessus de TensorFlow.



Keras étend la facilité d'utilisation de TensorFlow avec ces fonctionnalités supplémentaires pour la programmation ML et DL. Avec une communauté utile et un canal Slack dédié, il est facile d'obtenir de l'aide. Il existe également un support pour les réseaux neuronaux convolutionnels et récurrents, ainsi que pour les réseaux neuronaux standard [8].

1.1.2.4 Pandas

Pour manipuler mes données pendant le prétraitement des données, j'ai utilisé la bibliothèque Pandas. Pandas est une bibliothèque d'analyse de données Python et est utilisé principalement pour la manipulation et l'analyse des données. Elle entre en jeu avant que l'ensemble de données ne soit préparé pour l'apprentissage. Pandas permet aux programmeurs d'apprentissage automatique de travailler sans effort avec des séries chronologiques et des données multidimensionnelles structurées.



Pandas utilise les DataFrames, qui ne sont qu'un terme technique pour désigner une représentation bidimensionnelle des données, en offrant aux programmeurs des objets DataFrame.

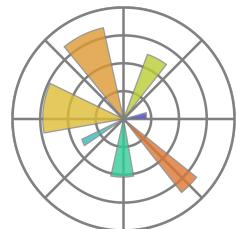
1.1.2.5 Numpy

La bibliothèque NumPy pour Python se concentre sur la gestion des données multidimensionnelles étendues et des fonctions mathématiques complexes opérant sur ces données. NumPy offre une rapidité de calcul et d'exécution de fonctions complexes travaillant sur des tableaux c'est pourquoi j'ai utilisé spécialement quand il s'agit de traiter les tensors.



1.1.2.6 Matplotlib

Pour visualiser mes données et mes résultats lors de l'entraînement du modèle, j'ai utilisé Matplotlib.



Matplotlib est une bibliothèque Python très populaire pour la visualisation de données. Comme Pandas, elle n'est pas directement liée à l'apprentissage automatique. Elle s'avère particulièrement utile lorsqu'un programmeur souhaite visualiser les modèles dans les données. Il s'agit d'une bibliothèque de traçage en 2D utilisée pour créer des graphiques et des tracés en 2D. Un module appelé pyplot facilite la tâche des programmeurs en matière de traçage, car il fournit des fonctionnalités permettant de contrôler les styles de lignes, les propriétés des polices, le formatage des axes, etc. Il fournit différents types de graphiques et de tracés pour la visualisation des données, à savoir des histogrammes, des diagrammes d'erreur, des barres de données, etc.

1.1.3 Les Outils

1.1.3.1 Jupyter Notebook

Le Jupyter Notebook est une application web open source que vous pouvez utiliser pour créer et partager des documents contenant du code, des équations, des visualisations et du texte. Jupyter Notebook est maintenu par les personnes du projet Jupyter [9].

Jupyter Notebooks est un projet dérivé du projet IPython, qui avait lui-même un projet IPython Notebook. Le nom, Jupyter, vient des principaux langages de programmation qu'il supporte : Julia, Python et R.

Jupyter est livré avec le noyau IPython, qui vous permet d'écrire vos programmes en Python, mais il existe actuellement plus de 100 autres noyaux que vous pouvez également utiliser.



1.1.3.2 Google Colab

Comme nous le savons, la formation à l'apprentissage profond nécessite un ordinateur puissant, j'ai donc dû utiliser google Colab.

Colaboratory est un environnement Jupyter notebook gratuit qui ne nécessite aucune installation et fonctionne entièrement dans le nuage. Avec Colaboratory, vous pouvez écrire et exécuter du code, enregistrer et partager vos analyses et accéder à de puissantes ressources informatiques, le tout gratuitement depuis votre navigateur [10].



Comme son nom l'indique, Google Colab intègre la collaboration dans son produit. En fait, il s'agit d'un carnet de notes Jupyter qui exploite les fonctionnalités de collaboration de Google Docs. Il fonctionne également sur les serveurs de Google qui donne un accès gratuit aux ressources informatiques de Google telles que les GPU et les TPU et vous n'avez pas besoin d'installer quoi que ce soit. De plus, les notebooks sont enregistrés sur votre compte Google Drive.

1.1.3.3 Anaconda

La plateforme de science des données d'entreprise pour les scientifiques des données, les professionnels de l'informatique et les dirigeants d'entreprise. Une distribution gratuite et open-source des langages de programmation Python et R pour le calcul scientifique,



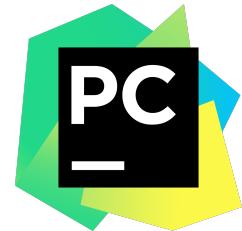
qui vise à simplifier la gestion et le déploiement des paquets. Les versions des paquets sont gérées par le système de gestion des paquets conda.

Anaconda Python est comme un sac de bibliothèques Python et R. Plus de 7500+ bibliothèques utile surtout lorsque vous voulez apprendre l'apprentissage automatique ou effectuer certaines tâches d'apprentissage automatique [11]. En effet, ses bibliothèques intégrées vous feront gagner le temps que vous auriez dû consacrer à la recherche et au téléchargement manuels.

1.1.3.4 Pycharm

Quand j'implémente mon modèle, il fonctionne bien dans colab mais quand j'essaie de l'exécuter dans django, je rencontre de nombreux problèmes c'est pourquoi j'ai utilisé pycharm pour vérifier les erreurs.

PyCharm est un IDE (Integrated Development Environment) spécialement conçu pour Python. Il est utilisé pour développer des applications basées sur Python. Il permet aux utilisateurs de créer un environnement virtuel où l'utilisateur peut travailler avec différents paquets indépendamment des autres paquets installés sur le système.



PyCharm offre une interface utilisateur très interactive qui le rend intéressant pour travailler avec Python.

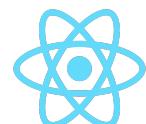
1.1.4 TensorBoard

En apprentissage automatique, pour améliorer quelque chose, il faut souvent pouvoir le mesurer. TensorBoard est un outil qui fournit les mesures et les visualisations nécessaires au cours du flux de travail de l'apprentissage automatique. Il permet de suivre les métriques d'expérience telles que la perte et la précision, de visualiser le graphe du modèle, de projeter les incorporations dans un espace de dimension inférieure, et bien plus encore .

1.2 Plate-forme Web

1.2.1 React Js

React.js est une bibliothèque JavaScript open-source utilisée pour créer des interfaces utilisateur spécifiquement pour les applications à page unique. Elle est utilisée pour gérer la couche de visualisation



des applications web et mobiles. React nous permet également de créer des composants d’interface utilisateur réutilisables [12].

React permet aux développeurs de créer de grandes applications web qui peuvent modifier les données, sans recharger la page. L’objectif principal de React est d’être rapide, évolutif et simple. Il ne fonctionne que sur les interfaces utilisateurs de l’application. Cela correspond à la vue dans le modèle MVC. Il peut être utilisé avec une combinaison d’autres bibliothèques ou frameworks JavaScript, comme Angular JS dans MVC.

1.2.1.1 Caractéristiques de React.js

- JSX : dans React, au lieu d’utiliser le JavaScript ordinaire pour la création de modèles, on utilise JSX un simple JavaScript qui permet de citer le HTML et utilise la syntaxe de ces balises HTML pour rendre les sous-composants.
- React Native : React a des bibliothèques natives qui ont été annoncées par Facebook en 2015, qui fournit l’architecture react aux applications natives comme IOS, Android et UPD.
- Flux de données à sens unique : dans React, un ensemble de valeurs immuables sont transmises au moteur de rendu du composant en tant que propriétés dans ses balises HTML. Le composant ne peut pas modifier directement les propriétés mais peut passer une fonction de rappel à l’aide de laquelle nous pouvons effectuer des modifications.
- Modèle d’objet de document virtuel : React crée un cache de structure de données en mémoire qui calcule les modifications apportées, puis met à jour le navigateur.

1.2.1.2 React vs Plain JS

La meilleure façon de comprendre pourquoi j’ai utilisé React est de le comparer à un simple JS. Un simple JS fonctionne sans aucune bibliothèque. Fondamentalement, JavaScript est un langage de script et il ne succombe à aucune règle. Vous êtes libre d’écrire du JS où vous voulez et de l’exécuter dans votre navigateur. Il ne suit aucune structure. Une fois qu’il est reçu dans votre navigateur, il est exécuté de haut en bas, ligne par ligne.

Cependant, dans react, la façon dont les données circulent a des règles clairement définies. Les applications React ont une structure. Chaque morceau de code qui s’exécute seul est contraint à l’intérieur d’un composant. Ces composants travaillent de manière isolée mais communiquent entre eux pour faire fonctionner l’application web.

	Plain js	React js
Création d'une interface utilisateur	Créé en HTML sur le serveur	Crée en jsx défini sur le navigateur
La distribution des fonctionnalités dans l'application	Aucune exigence quant à la manière de diviser les fonctionnalités ou les composants de l'interface utilisateur dans une application	Fait en sorte que votre application soit divisée en composants et que chacun de ces composants maintienne tout le code nécessaire.
Stockage des données	Généralement stockées dans le DOM	Composants contrôlés
Mises à jour de UI	Doivent se faire en trouvant le nœud du DOM à mettre à jour et en ajoutant ou supprimant manuellement des éléments	Met automatiquement à jour l'interface utilisateur en fonction de la définition et de la modification de l'état du composant.

Table 1.1: Différences entre React.js et Plain js

1.2.2 Django

Django est un framework web python open-source utilisé pour le développement rapide de sites web pragmatiques, maintenables, de conception propre et sécurisés. Un framework d'application web est une boîte à outils regroupant tous les composants nécessaires au développement d'une application [13].



L'objectif principal du framework Django est de permettre aux développeurs de se concentrer sur les composants de l'application qui sont nouveaux au lieu de passer du temps sur des composants déjà développés. Django est plus complet que de nombreux autres frameworks sur le marché. Il prend en charge un grand nombre de problèmes liés au développement Web et permet aux utilisateurs de se concentrer sur le développement des composants nécessaires à leur application.

1.2.2.1 Pourquoi Django

- Intégration MI : Django nous offre un moyen rapide et fiable de mettre en œuvre des modèles d'apprentissage automatique, en utilisant le même langage de programmation que celui dans lequel ils ont été écrits. Grâce à cela, la mise en œuvre de solutions d'apprentissage automatique devient facile .

- Philosophie DRY : la philosophie DRY (don't repeat yourself) de Django est ce qui aide les développeurs à réaliser un développement rapide. Vous pouvez travailler sur plus d'une itération d'une application à la fois sans devoir reprendre le code à zéro. Vous pouvez également réutiliser le code existant pour de futures applications tout en conservant l'authenticité et les fonctionnalités uniques.
- Mesures de sécurité : les systèmes de sécurité constituent l'une des plus grandes priorités de la plate-forme Django. Vous n'avez pas à mettre en œuvre manuellement des fonctions de sécurité pour poursuivre le développement web. Vous n'avez pas à vous soucier des problèmes de sécurité courants tels que l'injection SQL, le cross-site scripting et le clickjacking en utilisant la plate-forme. Des correctifs de sécurité sont fréquemment déployés sur la plate-forme afin de garantir le respect des dernières normes de sécurité.
- Fiabilité : Django existe depuis un certain temps déjà, et sa grande communauté rend la plate-forme encore meilleure. Il existe des sites Web dédiés à la plate-forme où vous pouvez trouver de l'aide pour tous les problèmes que vous pouvez rencontrer. Si vous avez besoin d'aide pour vos projets, vous pouvez toujours compter sur le soutien de la communauté.

1.2.2.2 Django Rest framework

Le framework Django REST (DRF) est une boîte à outils puissante et flexible permettant de créer des API Web. Son principal avantage est qu'il facilite grandement la sérialisation.

Le framework Django REST est basé sur les vues basées sur les classes de Django, c'est donc une excellente option si vous êtes familier avec Django. Il adopte des implémentations telles que les vues basées sur les classes, les formulaires, le validateur de modèle, QuerySet, etc.



Le framework Django Rest vous permet de créer des API RESTful : Un moyen de transférer des informations entre une interface et une base de données de manière simple. Il sépare l'interface utilisateur et le stockage des données et fait communiquer l'utilisateur et la base de données en envoyant un fichier .json.

1.2.2.3 Pourquoi Django Rest Framework

De nombreux frameworks vous permettent de créer facilement des API pour vos applications, mais le framework Django REST (DRF) est pratique à bien des égards et offre les avantages suivants :

- Son API navigable sur le Web est un énorme gain de convivialité pour vos développeurs.
- Les politiques d'authentification comprennent des paquets pour OAuth1 et OAuth2.
- La sérialisation prend en charge les sources de données ORM et non ORM.
- Le système est personnalisable jusqu'au bout. Il suffit d'utiliser des vues régulières basées sur des fonctions si vous n'avez pas besoin des fonctionnalités les plus puissantes.
- Il dispose d'une documentation complète et d'un excellent support communautaire.
- Il est utilisé par des entreprises de renommée internationale, dont Mozilla, Red Hat, Heroku et Eventbrite, qui lui font confiance.

Dans la figure ci-dessous, je vais montrer comment la communication entre React et Django est effectuée par le framework Rest.

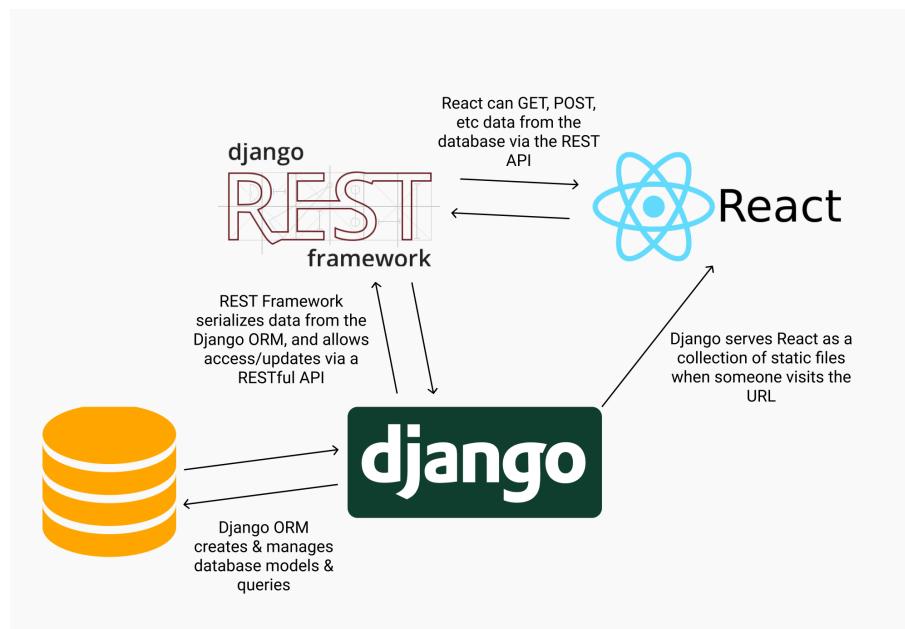


Figure 1.1: La communication entre Django et React js

1.2.3 Base de donnée

1.2.3.1 Postgresql

PostgreSQL est une base de données relationnelle open source avancée, de classe entreprise, qui prend en charge les requêtes SQL (relationnelles) et JSON (non relationnelles) [14]. Il s'agit d'un système de gestion de base de données très stable, soutenu par plus de 20 ans de développement communautaire qui a contribué à ses niveaux élevés de résilience, d'intégrité et de correction. PostgreSQL est utilisé comme magasin de données principal ou entrepôt de données pour de nombreuses applications web, mobiles, géospatiales et analytiques.

PostgreSQL a une histoire riche en matière de prise en charge de types de données avancés et supporte un niveau d'optimisation des performances qui est commun à ses homologues commerciaux, comme Oracle et SQL Server.



1.2.3.2 Pourquoi Postgresql

- Open Source DBMS : only PostgreSQL provides enterprise-class performance and functions among current Open Source DBMS with no end of development possibilities.
- ACID et Transaction : PostgreSQL supporte ACID (Atomicité, Cohérence, Isolation, Durabilité).
- Diverses techniques d'indexation : PostgreSQL ne fournit pas seulement des techniques d'indexation par arbre B+, mais aussi divers types de techniques telles que GIN (Generalized Inverted Index) et GiST (Generalized Search Tree), etc.
- Intégration dans BI et ML : PostgreSQL fonctionne bien avec les applications de BI. Cependant, il est plus adapté aux applications d'entreposage et d'analyse des données qui nécessitent des vitesses de lecture et d'écriture rapides.
- Intégration avec django : L'utilisation conjointe de PostgreSQL et de Django offre de nombreux avantages car Django fournit un certain nombre de types de données qui ne fonctionnent qu'avec PostgreSQL et dispose de `django.contrib.postgres` pour effectuer des opérations de base de données sur PostgreSQL.

Paramètre	SQL	NOSQL
Définition	Les bases de données SQL sont principalement appelées SGBDR ou bases de données relationnelles.	Les bases de données NoSQL sont principalement appelées bases de données non relationnelles ou distribuées.
Signe pour	Les SGBDR traditionnels utilisent la syntaxe et les requêtes SQL pour analyser et obtenir les données afin de mieux les comprendre. Ils sont utilisés pour les systèmes OLAP.	Le système de base de données NoSQL se compose de différents types de technologies de base de données. Ces bases de données ont été développées en réponse aux demandes présentées pour le développement de l'application moderne.
Langage de requête	Langage d'interrogation structuré (SQL)	Aucun langage d'interrogation déclaratif
Schema	SQL databases have a predefined schema	NoSQL databases use dynamic schema for unstructured data.
Schema	SQL databases have a predefined schema	NoSQL databases use dynamic schema for unstructured data.
Meilleure utilisation pour	La base de données RDBMS est la bonne option pour résoudre les problèmes ACID.	NoSQL est la meilleure option pour résoudre les problèmes de disponibilité des données.

Table 1.2: Comparaison entre SQL et NOSQL

1.2.4 Langues

1.2.4.1 HTML 5

Le Framework React js utilise la dernière version du HTML. La version HTML 5 a été finalisée en octobre 2014. Le html est un langage balisé permettant de structurer les données d'une page Internet.

1.2.4.2 CSS 3

Le framework React js arrive avec Css 3 .Les feuilles de style en cascade (CSS) sont un mécanisme simple qui permet aux développeurs de styliser plus facilement les éléments HTML. Ils sont moins dépendants des fichiers d'image et peuvent réaliser le style CSS avec moins de lignes de code.

1.2.4.3 SASS

Le Framework React js embarque aussi avec lui le SCSS avec la bibliothèque node-sass. Ce format de fichier à la même particularité que les fichiers CSS pour formater le texte. Cependant, il dispose de fonctionnalités syntaxiques en plus, comme des variables par exemple. Il est à noter qu'un fichier CSS est un fichier SCSS valide car le SCSS n'est qu'une extension de ce format.

1.2.4.4 Js

JavaScript est couramment utilisé pour créer des pages Web. Il permet d'ajouter un comportement dynamique à mon plateforme et d'y ajouter des effets spéciaux. Il est principalement utilisé à des fins de validation dans mon React code .

1.2.4.5 Python

Python est l'un des choix les plus populaires en matière de programmation backend. Il est relativement nouveau et dispose d'une énorme bibliothèque, c'est pourquoi je l'ai utilisé dans le backend de django pour profiter de la puissance de ce langage.

1.2.4.6 Json

J'ai utilisé JSON pour la communication entre react js et django "frontend to backend" parce que c'est extrêmement léger à envoyer et à recevoir dans les requêtes et réponses HTTP en raison de la petite taille du fichier. Il est facile à lire par rapport à quelque chose comme le XML car il est beaucoup plus propre et il n'y a pas autant de balises d'ouverture et de fermeture dont il faut se préoccuper [15].

1.2.5 Outils

1.2.5.1 Vs code

Vs code est l'éditeur de texte que j'utilise quand je fais du développement web. Il a un nombre important de fonctionnalités autres que le simple traitement de texte comme par exemple des auto-complétions des fonctions de base de plusieurs langages, des beautifiers, des correcteurs syntaxiques, un gestionnaire de projet, etc. On peut vraiment y installer ce qu'on veut, puisqu'il y a la possibilité d'installer les packages que l'on veut.

1.2.5.2 Redux devtools

Pour gérer mon état de réaction, j'ai utilisé ReduxDevTools, qui me fournit une plateforme de débogage pour les applications Redux. Il me permet d'effectuer le débogage du voyage dans le temps et l'édition en direct. Il me permet d'inspecter chaque état et charge utile d'action et de revenir en arrière dans le temps en "annulant" des actions.

1.2.5.3 React Profiler

J'ai parfois des problèmes avec les performances de React. J'ai donc utilisé react profiler pour mesurer la fréquence de rendu d'une application React et le "coût" du rendu. Son but est d'aider à identifier les parties d'une application qui sont lentes et qui peuvent bénéficier d'optimisations telles que la mémorisation.

1.2.5.4 Lighthouse

Lighthouse est l'extension que j'ai utilisée pour obtenir un rapport sur ma plateforme web. Il s'agit d'un outil automatisé à code source ouvert permettant d'améliorer les performances, la qualité et l'exactitude d' applications web.

Lors de l'audit d'une page, Lighthouse exécute une série de tests sur la page, puis génère un rapport sur les résultats de la page. À partir de là, je peux utiliser les tests défaillants comme indicateurs de ce que vous pouvez faire pour améliorer l'application.

1.2.6 bibliothèques

1.2.6.1 React-redux

Comme je l'ai mentionné, React js est un cadre de flux de données à sens unique parce que l'état se déplace uniquement du parent à l'enfant, je dois donc utiliser Redux pour la gestion de l'état de l'application. Pour résumer, Redux maintient l'état de l'application entière dans un seul arbre d'état immuable (objet), qui ne peut pas être changé directement. Lorsque quelque chose change, un nouvel objet est créé (en utilisant des actions et des réducteurs).

1.2.6.2 React-hook

Quand je travaillais avec des composants de classe React, j'ai été confronté à de nombreux problèmes avec la logique de classe, c'est pourquoi je suis passé à des composants

fonctionnels avec React Hook, dans lesquels je n'ai pas à refactoriser un composant fonctionnel en un composant de classe quand il se développe, ce qui rend les composants plus réutilisables. Il permet également de partager la logique d'état entre les composants.

1.2.6.3 Axios

Dans mes applications React, j'ai souvent besoin de récupérer des données à partir d'API externes afin de les afficher dans mes pages Web. Une façon de construire cette fonctionnalité est d'utiliser l'API Javascript Fetch. Fetch est tout à fait capable de récupérer des données externes, mais il a quelques limitations, c'est pourquoi j'ai utilisé la bibliothèque Axios.

Axios est conçue pour gérer les demandes et les réponses http. Elle est plus souvent utilisée que Fetch car elle offre un ensemble de fonctionnalités plus large et prend en charge les anciens navigateurs.

Axios traite les réponses à l'aide de Promises, ce qui en fait un outil rationalisé et facile à utiliser dans notre code. Axios utilise des méthodes comme get() et post() qui effectuent des requêtes http GET et POST pour récupérer ou créer des ressources.

1.2.6.4 JWT

Il y a différents choix pour le type de jeton d'accès à utiliser, et les jetons Web JSON (JWTs) sont une excellente option que j'ai utilisée pour l'authentification de mon application react et django.

JSON web token (JWT), est un standard ouvert (RFC 7519) [16] qui définit un moyen compact et autonome de transmettre de manière sécurisée des informations entre parties sous forme d'objet JSON.

En raison de sa taille relativement petite, un JWT peut être envoyé par une URL, par un paramètre POST ou à l'intérieur d'un en-tête HTTP, et il est transmis rapidement. Un JWT contient toutes les informations nécessaires sur une entité pour éviter d'interroger une base de données plus d'une fois. Le destinataire d'un JWT n'a pas non plus besoin d'appeler un serveur pour valider le jeton.

1.2.6.5 UI Libraries

Pour réaliser mon design d'interface utilisateur, j'ai utilisé deux bibliothèques d'interface utilisateur populaires, compatibles avec react js.

Bootstrap est un framework CSS, HTML et JS puissant qui permet de créer des applications Web réactives. Il s'agit du framework le plus populaire pour la création

d'applications mobiles et de bureau. Son nom original est Twitter Blueprint, car Bootstrap a été développé par l'équipe de Twitter en tant que framework interne.

Material UI est un framework React UI qui suit les principes du Material design. Material UI est basé sur le framework React de Facebook et contient des composants qui sont réalisés selon les directives de Material. Material design a été développé par Google en 2014 tandis que Material UI a été développé par une petite équipe dévouée et passionnée en 2017.

1.2.7 Déploiement

1.2.7.1 Docker

Dans ma plateforme web, j'ai tellement de dépendances qu'il faut les installer pour qu'elle fonctionne. J'ai donc utilisé un conteneur docker qui inclut les dépendances logicielles nécessaires à la plateforme, comme des versions spécifiques des runtimes de langages de programmation et d'autres bibliothèques logicielles.

1.2.7.2 Heroku

Pour déployer ma plateforme web, j'ai utilisé le serveur Heroku, une plateforme en tant que service (PaaS) basée sur des conteneurs, que j'utilise pour déployer, gérer et mettre à l'échelle mon application.

Heroku est élégant, flexible et facile à utiliser, offrant le chemin le plus simple pour mettre l'application sur le marché rapidement et à faible coût.

Chapter 2

Méthodes et résultats

Dans ce chapitre, je vais détailler la solution que j'ai utilisée pour mettre en œuvre ma conception du modèle d'apprentissage profond et de la plateforme web.

2.1 Mise en œuvre du deep learning modal

2.1.1 Importation de dataset

Pour travailler avec le dataset food101, j'ai utilisé TFDS. qui est un endroit pour les ensembles de données d'apprentissage automatique préparés et prêts à l'emploi, car il charge les données déjà au format tenseur et permet d'expérimenter différentes techniques de modélisation sur un dataset cohérent.

```
import tensorflow_datasets as tfds

# Load in Food101 data from TFDS (takes about 5-6 minutes in Google Colab)
(train_data, test_data), ds_info = tfds.load(name="food101",
                                             split=["train", "validation"],
                                             shuffle_files=True,
                                             as_supervised=True,
                                             with_info=True)
```

Figure 2.1: Importation de dataset avec TFDS

2.1.2 Prétraitement des données

Puisque j'ai téléchargé les données à partir des datasets TensorFlow, il y a quelques étapes de prétraitement que je dois effectuer avant qu'elles soient prêtes à être modélisées.

Plus précisément, mes données sont actuellement :

- En type de données uint8.
- Composées de tous les tenseurs de tailles différentes (images de tailles différentes)
- Non mises à l'échelle (les valeurs des pixels sont comprises entre 0 et 255).

Alors que les modèles aiment que les données soient :

- Dans le type de données float32
- Avoir tous les tenseurs de la même taille (les lots exigent que tous les tenseurs aient la même forme, par exemple (224, 224, 3)).
- Mis à l'échelle (valeurs entre 0 et 1), également appelés normalisés.

Pour s'en occuper, je vais créer une fonction preprocess_img() qui :

- Redimensionne un tenseur d'image d'entrée à une taille spécifiée en utilisant tf.image.resize().
- Convertit le type de données actuel d'un tenseur d'image en entrée en tf.float32 à l'aide de tf.cast().

Remarque : les modèles EfficientNetB0 pré-entraînés dans tf.keras.applications.efficientnet (que je vais utiliser) ont un redimensionnement intégré.

Batch prepare datasets :

Avant de pouvoir modéliser les données, je dois les transformer en lots.

Pourquoi ?

Parce que le calcul sur des lots est efficace en termes de mémoire.

Je transforme mes données de 101 000 tenseurs d'images et étiquettes (train et test combinés) en lots de 32 paires d'images et d'étiquettes, ce qui leur permet de tenir dans la mémoire de notre GPU.

Pour y parvenir de manière efficace, je vais utiliser un certain nombre de méthodes de l'API tf.data.

Plus précisément, je vais utiliser :

- map() - fait correspondre une fonction prédéfinie à un dataset cible (par exemple, preprocess_img() à nos tenseurs d'images).

- shuffle() - mélange aléatoirement les éléments d'un dataset cible dans une taille de tampon (idéalement, la taille de tampon est égale à la taille du dataset, mais cela peut avoir des implications sur la mémoire).
- batch() - transforme les éléments d'un dataset cible en lots (taille définie par le paramètre batch_size).
- prefetch() - prépare les lots de données suivants pendant que d'autres lots de données sont en cours de calcul (améliore la vitesse de chargement des données mais coûte de la mémoire).
- cache() - met en cache les éléments d'un dataset cible, ce qui permet de gagner du temps de chargement (uniquement si le dataset est suffisamment petit pour tenir en mémoire, les instances standard de Colab ne disposent que de 12 Go de mémoire).

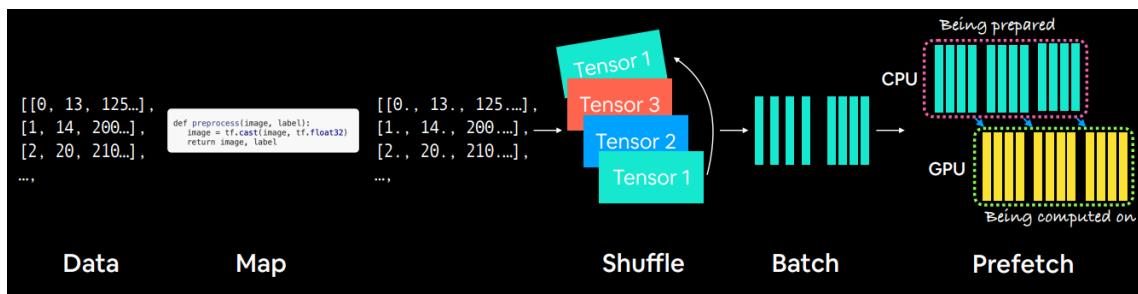


Figure 2.2: Étapes du prétraitement des données

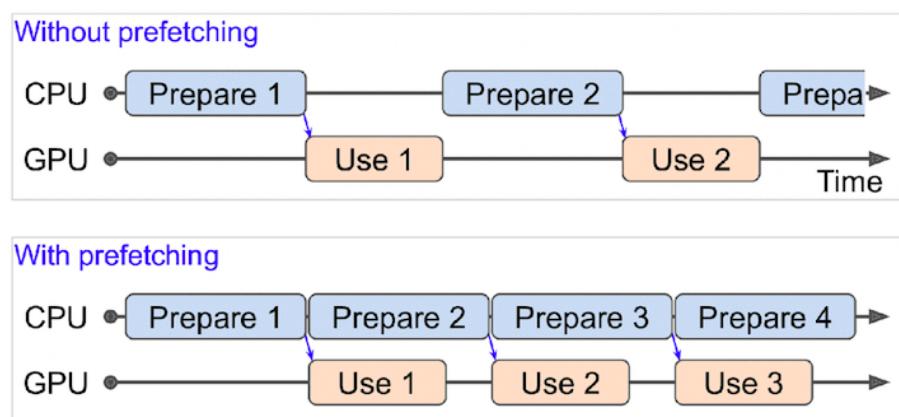


Figure 2.3: Prefetching vs not using prefetching

2.1.3 Crédit du modelling callbacks

Étant donné que je vais m'entraîner sur une grande quantité de données et que l'entraînement pourrait prendre beaucoup de temps, c'est une bonne idée de mettre en place quelques callbacks de modélisation afin d'être sûr que des choses comme les journaux d'entraînement de notre modèle soient suivis et que mon modèle soit checkpointed (sauvegardé) après diverses étapes d'entraînement.

Pour chacune de ces opérations, je vais utiliser les callbacks suivants :

- `tf.keras.callbacks.TensorBoard()` : permet de garder la trace de l'historique d'entraînement du modèle afin que je puisse l'inspecter ultérieurement.
- `tf.keras.callbacks.ModelCheckpoint()` : enregistre la progression du modèle à différents intervalles afin que je puisse le charger et le réutiliser ultérieurement sans avoir à le réentraîner.

Le pointage est également utile pour pouvoir commencer à affiner mon modèle à une époque particulière et revenir à un état antérieur si l'affinage ne présente aucun avantage.

2.1.4 Mise en place mixed precision training

Normalement, les tenseurs dans TensorFlow utilisent par défaut le type de données float32 (sauf indication contraire). Les GPU ont également une mémoire limitée et ne peuvent donc gérer qu'un certain nombre de tenseurs float32 en même temps, c'est là qu'intervient l'apprentissage en précision mixte (MPT), qui consiste à utiliser un mélange de tenseurs float16 et float32 pour mieux utiliser la mémoire de votre GPU.

Comme nous l'avons déjà mentionné, lorsque vous utilisez l'entraînement à précision mixte, le modèle utilise les types de données float32 et float16 pour utiliser moins de mémoire si possible et s'exécuter plus rapidement (utiliser moins de mémoire par tenseur signifie que plus de tenseurs peuvent être calculés simultanément).

En conséquence, l'utilisation de l'apprentissage en précision mixte améliore mes performances sur les GPU de Colab (avec un score de capacité de calcul de 7.0+) jusqu'à 3x.

	Prefetch and mixed precision	No prefetch and no mixed precision
Time per epoch	280-300s	1127-1397s

Table 2.1: Résultat du temps par époche en utilisant MPT

2.1.5 Construire le modèle



Figure 2.4: Architecture de modèle

Maintenant que les callbacks sont prêts à être utilisés et que la précision mixte est activée, nous allons construire le modèle.

Pour construire le modal, je vais :

- Utiliser EfficientNetB0 de `tf.keras.applications` pré-entraîné sur ImageNet comme modèle de base, je le téléchargerai sans les couches supérieures en utilisant le paramètre `include_top=False` pour pouvoir créer mes propres couches de sortie.
- Geler les couches du modèle de base afin que nous puissions utiliser les modèles préappris que le modèle de base a trouvé sur ImageNet.
- Rassemblez les couches d'entrée, de base, de mise en commun et de sortie dans un modèle fonctionnel.
- Compilez le modèle fonctionnel en utilisant l'optimiseur Adam et la fonction de perte sparse categorical crossentropy (puisque nos étiquettes ne sont pas codées à un coup).
- Ajuster le modèle en utilisant les callbacks TensorBoard et ModelCheckpoint.

Mais le problème ici est de savoir quand je dois arrêter la formation.

Idéalement, lorsque mon modèle cesse de s'améliorer. Mais là encore, en raison de la nature du deep learning, il peut être difficile de savoir quand exactement un modèle cessera de s'améliorer. Heureusement, il existe une solution : le rappel EarlyStopping.

Le rappel EarlyStopping surveille une métrique de performance de modèle spécifiée (par exemple, `val_loss`) et lorsqu'elle cesse de s'améliorer pendant un nombre spécifié d'époques, il arrête automatiquement la formation.

En utilisant le rappel EarlyStopping combiné avec le rappel ModelCheckpoint qui sauvegarde automatiquement le modèle le plus performant, je peux continuer à entraîner mon modèle pendant un nombre illimité d'époques jusqu'à ce qu'il cesse de s'améliorer.

Un autre callback que nous allons implémenter : Comme je l'ai mentionné dans la partie précédente, le taux d'apprentissage est l'hyperparamètre de modèle le plus important que nous pouvons régler car il surveille une métrique spécifiée et lorsque cette métrique cesse de s'améliorer, il réduit également le taux d'apprentissage par un facteur spécifié (par exemple, il divise le taux d'apprentissage par 10 dans ma modale).

Remarque : comme j'utilise un apprentissage à précision mixte, mon modèle a besoin d'une couche de sortie séparée avec un dtype=float32 codé en dur, par exemple, layers.Activation("softmax", dtype=tf.float32). Cela garantit que les sorties de mon modèle sont renvoyées au type de données float32 qui est numériquement plus stable que le type de données float16 (important pour les calculs de perte).

```

from tensorflow.keras import layers
from tensorflow.keras.layers.experimental import preprocessing

# Create base model
input_shape = (224, 224, 3)
base_model = tf.keras.applications.EfficientNetB0(include_top=False)
base_model.trainable = False # freeze base model layers

# Create Functional model
inputs = layers.Input(shape=input_shape, name="input_layer")
x = preprocessing.Rescaling(1./255)(x)
x = base_model(inputs, training=False)
x = layers.GlobalAveragePooling2D(name="pooling_layer")(x)
x = layers.Dense(len(class_names))(x) # want one output neuron per class
# Separate activation of output layer so we can output float32 activations
outputs = layers.Activation("softmax", dtype=tf.float32, name="softmax_float32")(x)
model = tf.keras.Model(inputs, outputs)

# Compile the model
model.compile(loss="sparse_categorical_crossentropy", # Use sparse_categorical_crossentropy when labels are *not* one-hot
              optimizer=tf.keras.optimizers.Adam(),
              metrics=["accuracy"])

Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0_notop.h5
16711680/16785208 [=====] - 0s/step
  
```

Figure 2.5: Création de modèle

Bon, le modèle a été compilé, maintenant ajustons-le sur toutes les données. Je vais le configurer pour qu'il fonctionne jusqu'à 100 époques. Comme je vais utiliser le rappel EarlyStopping, il pourrait s'arrêter avant d'atteindre 100 époques.

```

history_101_food_classes_all_data_fine_tune = loaded_gs_model.fit(train_data,
                                                               epochs=100, # fine-tune for a maximum of 100 epochs
                                                               steps_per_epoch=len(train_data),
                                                               validation_data=test_data,
                                                               validation_steps=int(0.15 * len(test_data)), # validation during training on 15% of test data
                                                               callbacks=[create_tensorboard_callback("training_logs", "efficientbo_101_classes_all_data_fine_tuning"),
                                                               model_checkpoint, # save only the best model during training
                                                               early_stopping, # stop model after X epochs of no improvements
                                                               reduce_lr]) # reduce the learning rate after X epochs of no improvements

Saving TensorBoard log files to: training_logs/efficientbo_101_classes_all_data_fine_tuning/20210317-034947
Epoch 1/100
2368/2368 [=====] - 304s 123ms/step - loss: 0.9813 - accuracy: 0.7357 - val_loss: 0.7932 - val_accuracy: 0.7842
INFO:tensorflow:Assets written to: fine_tune_checkpoints/assets
INFO:tensorflow:Assets written to: fine_tune_checkpoints/assets
Epoch 2/100
2368/2368 [=====] - 287s 121ms/step - loss: 0.5896 - accuracy: 0.8363 - val_loss: 0.8097 - val_accuracy: 0.7820
Epoch 3/100
2368/2368 [=====] - 287s 121ms/step - loss: 0.3337 - accuracy: 0.9068 - val_loss: 0.8791 - val_accuracy: 0.7850
Epoch 00003: ReduceLROnPlateau reducing learning rate to 1.9999999494757503e-05.
Epoch 4/100
2368/2368 [=====] - 287s 121ms/step - loss: 0.1099 - accuracy: 0.9724 - val_loss: 0.9399 - val_accuracy: 0.8008

```

Activate Windows
Get 10% off to activate Windows.

Figure 2.6: Ajustement du modèle

Note : l'ajustement du modèle avec des pipelines de données non optimisés et sans entraînement de précision mixte me prend un temps assez long par époque (environ 15-20 minutes sur les GPU de Colab). Mais avec le code optimisé que j'ai écrit ci-dessus, le modèle fonctionne beaucoup plus rapidement (plutôt 4-5 minutes par époque).

2.1.6 Résultat

Maintenant, après avoir monté mon modèle, je peux enfin voir s'il bat l'état de l'art que j'ai lu. J'ai donc besoin de l'évaluer :

```

# Evaluate mixed precision trained loaded model
results_loaded_gs_model_fine_tuned = loaded_gs_model.evaluate(test_data)
results_loaded_gs_model_fine_tuned

790/790 [=====] - 55s 69ms/step - loss: 0.9397 - accuracy: 0.7964
[0.9397100210189819, 0.7963564395904541]

```

Figure 2.7: Évaluation du modèle

Je peux voir que mon modèle obtient un excellent résultat "79.63% de précision" qui bat beaucoup d'articles de recherche que j'ai étudiés avec le temps minimum pour l'entraînement "seulement 15min" et le minimum de ressources fournies.

Puisque j'ai suivi les journaux d'entraînement de réglage fin de mon modèle à l'aide du rappel TensorBoard, téléchargeons-les et inspectons-les sur TensorBoard.dev.

Visualisons maintenant la précision de mes données de formation et de test pour vérifier si le modèle fonctionne bien ...

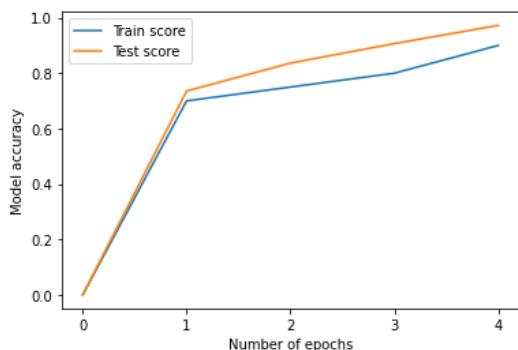


Figure 2.8: Train VS Test data accuracy

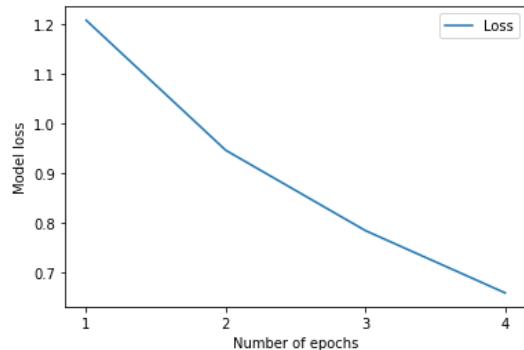


Figure 2.9: Visualisation de loss function

Comme nous le voyons, mes données d’entraînement ont une précision de "97%" tandis que la précision de mes données de test est de "79,88%", donc ici j’ai un peu surapprentissage mais cela n’affecte pas la prédiction. De plus, la fonction de perte diminue avec les époques, ce qui prouve que l’apprentissage se déroule correctement.

2.2 Mise en œuvre de la plate-forme Web

2.2.1 Structure du projet

Dans cette sous-section, j’expliquerai l’arborescence principale (répertoires, structure) de l’ensemble du projet à l’aide d’une capture d’écran d’illustration.

Le dossier de mon projet contient deux dossiers importants : backend et frontend et je vais détailler les sous-dossiers de chacun d’eux.

Il faut savoir que toute cette arborescence est créée automatiquement par le Framework django lors de la création d’un nouveau projet avec la commande **django-admin startproject [projectname]**. Cela semble très complexe mais je vais vous expliquer les principaux éléments :

- Le répertoire racine externe est un conteneur pour mon projet .
- Manage.py : Un utilitaire en ligne de commande qui permet d’interagir avec ce projet Django de différentes manières.
- Le backend interne est le paquetage Python réel pour le projet. Son nom est le nom du paquetage Python que je dois utiliser pour importer tout ce qui s’y trouve (par exemple setup.urls).
- **__init__.py** : Un fichier vide qui indique à Python que ce répertoire doit être considéré comme un paquetage Python.

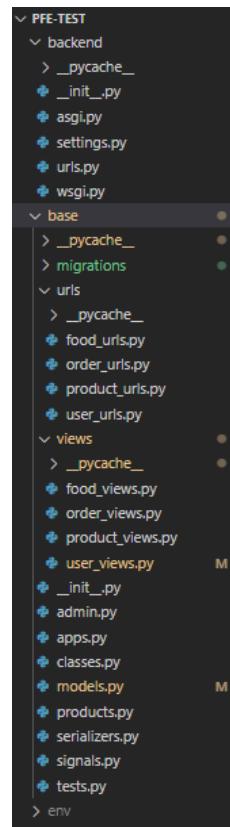


Figure 2.10: Contenu du dossier backend

- Settings.py : Paramètres/configuration pour ce projet Django.
- Urls.py : Les déclarations d’URL pour ce projet Django ; une "table des matières" de mon site alimenté par Django.
- Asgi.py : Un point d’entrée pour les serveurs web compatibles avec ASGI pour servir le projet.
- Wsgi.py : Un point d’entrée pour les serveurs web compatibles WSGI pour servir le projet.
- Admin.py : Enregistrement de mes modèles ici pour qu’ils puissent être ajoutés et personnalisés sur le tableau de bord d’administration de Django.
- Apps.py : Contient le code de configuration des applications
- Models.py : Définir mes modèles ici. Ce sont les tables utilisées pour stocker les données et définir leurs relations.
- Tests.py : Écriture de tests pour l’application

- Views.py : Définir les données qui seront rendues sur les modèles
- Env directoire : a folder contains all the necessary executables to use the packages that my project would need

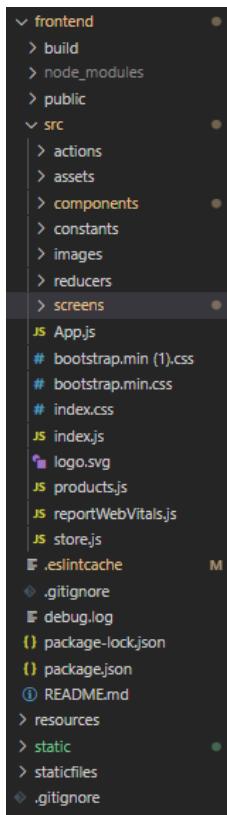


Figure 2.11: Contenu du dossier frontend

Toute cette arborescence frontale est créée automatiquement par le framework React js lorsqu'un nouveau projet est créé avec la commande **npx create-react-app**

- Node_modules : contient toutes les dépendances nécessaires pour une application react initiale fonctionnelle.
- .Gitignore : ce fichier spécifie les fichiers non suivis intentionnellement que Git doit ignorer.
- Package.json : ce fichier contient diverses métadonnées qui sont pertinentes pour mon projet. Il spécifie les dépendances utilisées dans le projet, ce qui aide npm à configurer le même environnement sur différentes machines pour mon projet.

- README.md : ce fichier peut être utilisé pour définir l'utilisation, les instructions de construction, le résumé de mon projet, etc. Il utilise le langage de balisage markdown pour créer le contenu.
- Package-lock.json : il garantit que mon paquet est cohérent sur plusieurs machines en stockant les versions des dépendances installées avec votre paquet.
- Public : Dossier racine qui est servi comme notre application react.
- src : Dans sa forme la plus simple, c'est mon dossier react app c'est-à-dire contenant les composants, les tests, les fichiers css, etc. C'est l'esprit de mon application.
- App.js : Ce fichier contient un composant react très basique défini qui peut être remplacé par mon propre composant racine.
- App.css : Contient les styles de mon composant react (App.js)
- Actions : contenir toutes les actions Redux .
- Reducers : contenir toutes les Redux reducers
- Store : un fichier qui garde la trace de l'état actuel de l'application.
- Screens : contient tous les composants correspondants contenant de la logique. Chaque conteneur aura un ou plusieurs composants
- Components : contient tous les composants " simples " ou de présentation, constitués uniquement de HTML et de style.
- Assets : dossier qui contient tous les autres fichiers à exécuter dans mon projet comme les images, les polices ...

2.2.2 Mise en œuvre des parties intéressantes des cas d'utilisation et Ecrans principaux de l'application

Dans cette deuxième sous-section, j'expliquerai plus en détail l'implémentation des parties intéressantes de certains exemples de cas d'utilisation et les ecrans principaux de l'application.

2.2.2.1 Utilisation de Food AI

Comme nous le voyons dans les captures d'écran, l'utilisateur doit télécharger son plat dans mon cas j'ai pris une photo réelle dans Mega Pizza restaurant à Annaba pour tester un cas réel, puis en utilisant mon modèle de deep learning déployé sur django dont j'ai parlé dans la section précédente, je peux prédire le plat téléchargé après avoir effectué quelques étapes de prétraitement de l'image comme indiqué dans l'image en-dessus et cela pour rendre l'image d'entrée identique à l'entrée de mon modèle.

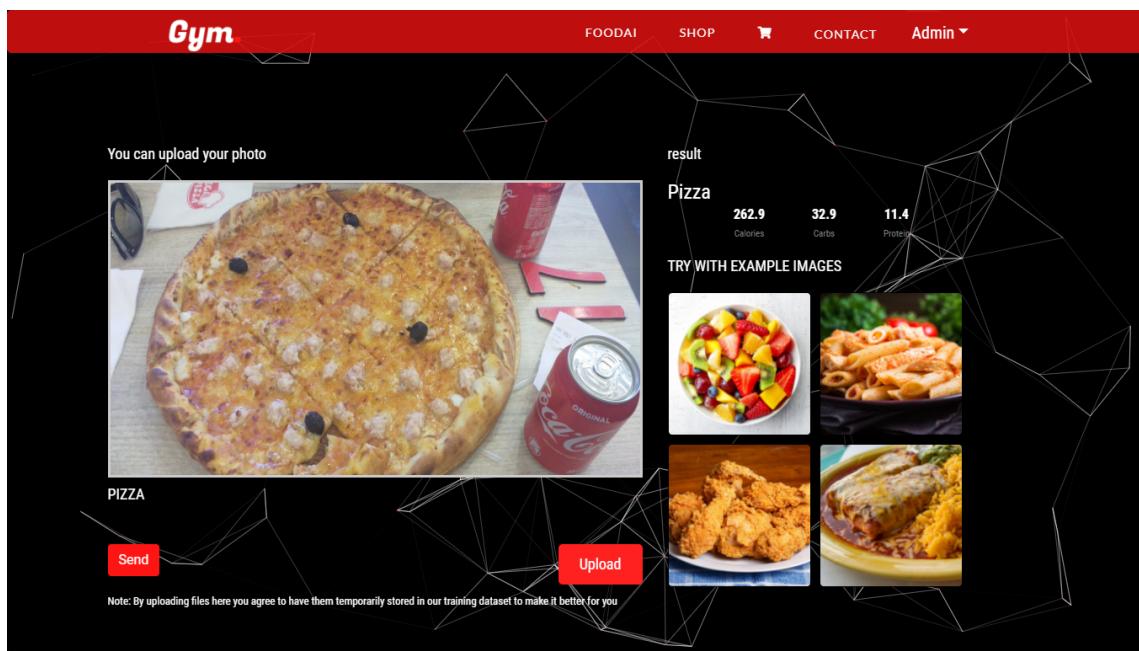
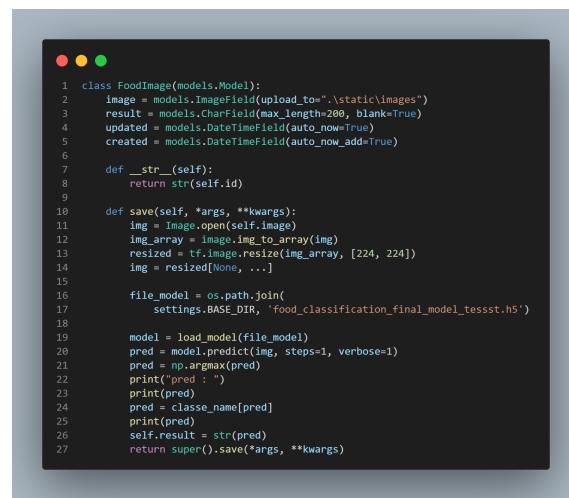


Figure 2.12: Food AI avec cas réel



```
1  class FoodImage(models.Model):
2      image = models.ImageField(upload_to=".\\static\\images")
3      result = models.CharField(max_length=200, blank=True)
4      updated = models.DateTimeField(auto_now=True)
5      created = models.DateTimeField(auto_now_add=True)
6
7      def __str__(self):
8          return str(self.id)
9
10     def save(self, *args, **kwargs):
11         img = Image.open(self.image)
12         img_array = image.img_to_array(img)
13         resized = tf.image.resize(img_array, [224, 224])
14         img = resized[None, ...]
15
16         file_model = os.path.join(
17             settings.BASE_DIR, 'food_classification_final_model_tessst.h5')
18
19         model = load_model(file_model)
20         pred = model.predict(img, steps=1, verbose=1)
21         pred = np.argmax(pred)
22         print("pred : ")
23         print(pred)
24         pred = classe_name[pred]
25         print(pred)
26         self.result = str(pred)
27
28     return super().save(*args, **kwargs)
```

Figure 2.13: Food AI in Django

2.2.2.2 Authentification de l'athlète

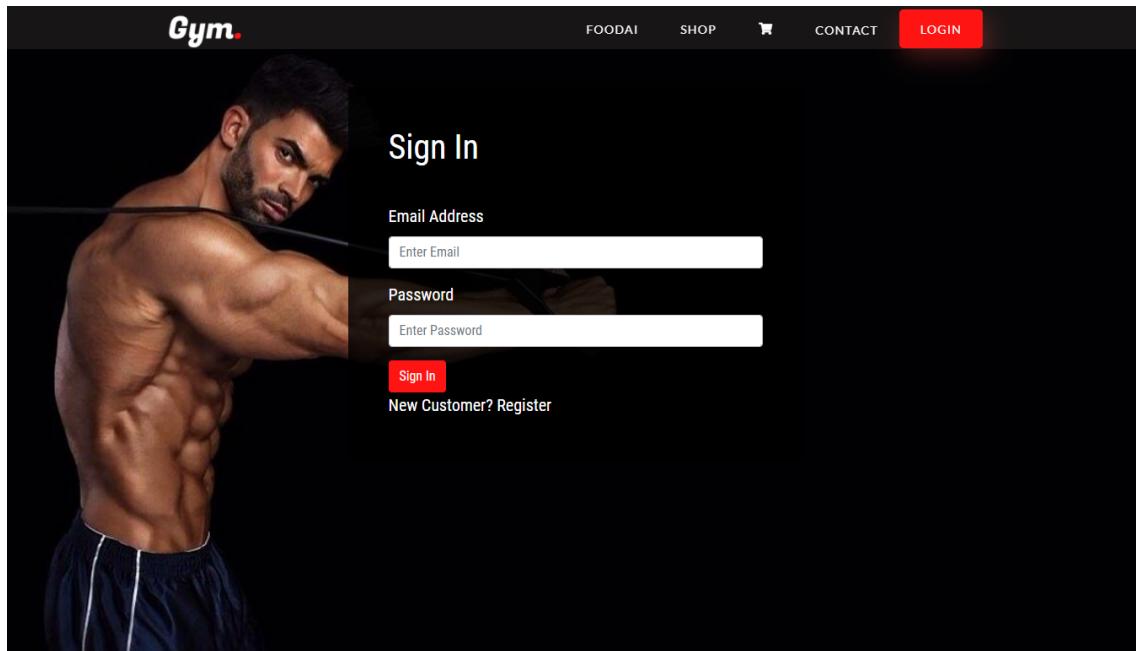


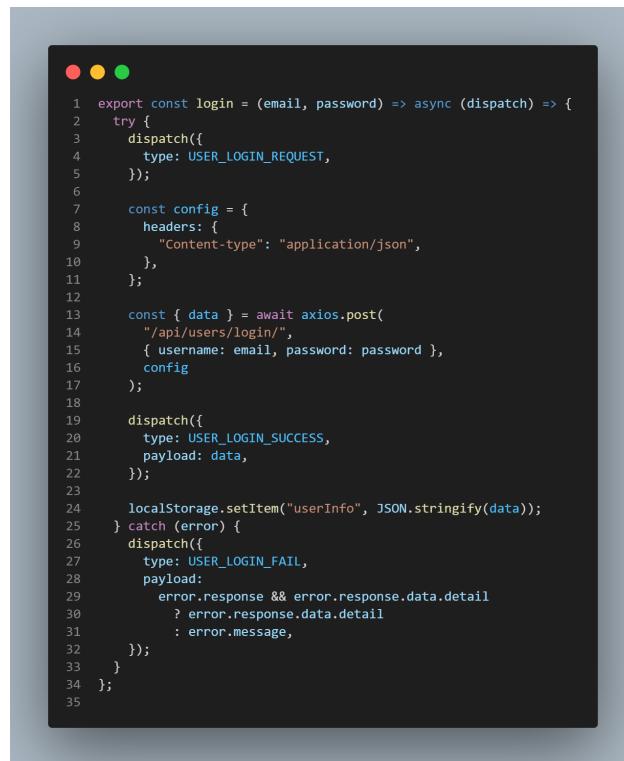
Figure 2.14: Login screen design

D'abord, je dois envoyer une action de login avec l'email et le mot de passe de l'utilisateur avec la fonction suivant.



Figure 2.15: Login screen in React screen

Ensuite, dans le fichier d'actions de l'utilisateur, j'ai la fonction `login()` qui envoie une requête post avec axios au django view comme indiqué ci-dessous.



```

1 export const login = (email, password) => async (dispatch) => {
2   try {
3     dispatch({
4       type: USER_LOGIN_REQUEST,
5     });
6
7     const config = {
8       headers: {
9         "Content-type": "application/json",
10      },
11    };
12
13     const { data } = await axios.post(
14       "/api/users/login/",
15       { username: email, password: password },
16       config
17     );
18
19     dispatch({
20       type: USER_LOGIN_SUCCESS,
21       payload: data,
22     });
23
24     localStorage.setItem("userInfo", JSON.stringify(data));
25   } catch (error) {
26     dispatch({
27       type: USER_LOGIN_FAIL,
28       payload: {
29         error.response && error.response.data.detail
30         ? error.response.data.detail
31         : error.message,
32       },
33     });
34   }
35 }

```

Figure 2.16: Authentication in React Action

Enfin, le request sera sérialisée par la fonction UserSerializerWithToken() dans le serializer de django, puis traitée par MyTokenObtainPairSerializer() dans django user views.



```

1 class UserSerializerWithToken(serializers.ModelSerializer):
2   token = serializers.SerializerMethodField(read_only=True)
3   userProfile = serializers.SerializerMethodField(read_only=True)
4
5   class Meta:
6     model = User
7     fields = '__all__'
8
9   def get_token(self, obj):
10     token = RefreshToken.for_user(obj)
11     return str(token.access_token)
12
13   def get_userProfile(self, obj):
14
15     try:
16       profile = UserProfileSerializer(
17         obj.userprofile, many=False).data
18
19     except:
20       profile = False
21     return profile
22

```

Figure 2.17: Authentication in django

2.2.2.3 User Consulting Store

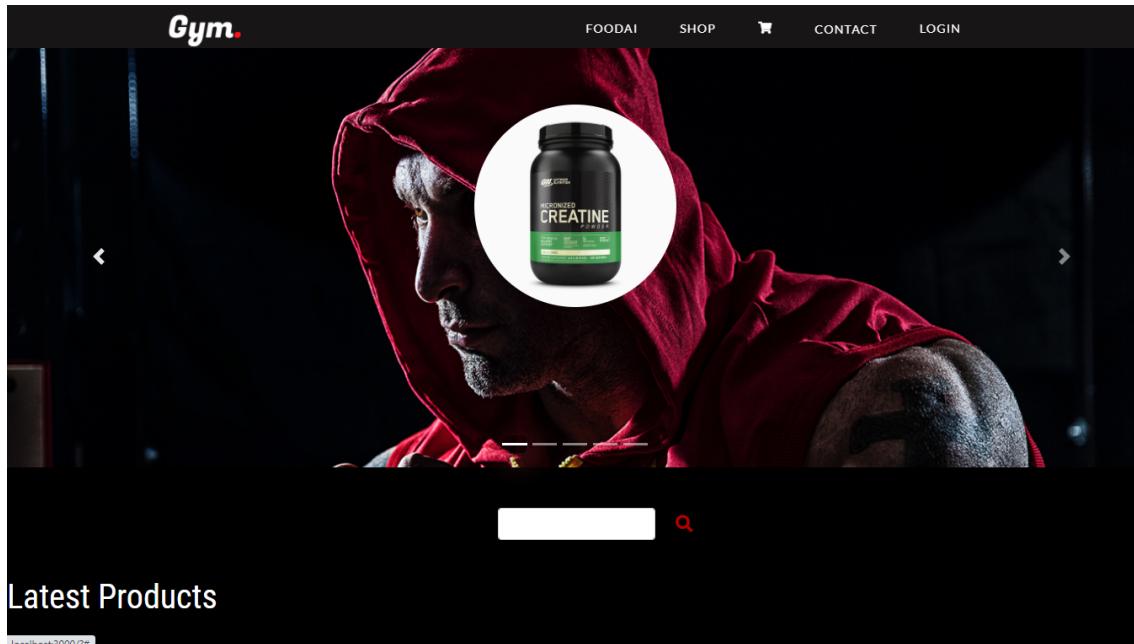
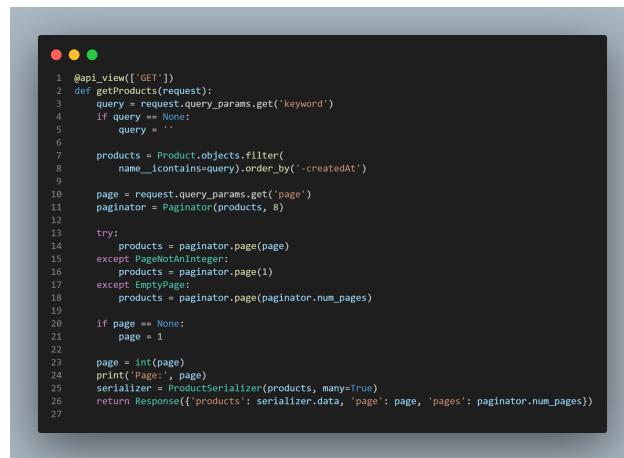


Figure 2.18: Shop screen design

```
1  useEffect(() => {
2    dispatch(listProducts(keyword));
3  }, [dispatch, keyword]);
4
5  return (
6    <div>
7      {!keyword && <ProductCarousel />}
8      <SearchBox />
9      <h1>Latest Products</h1>
10     {loading ? (
11       <Loader />
12     ) : error ? (
13       <Message variant="danger">{error}</Message>
14     ) : (
15       <Container>
16         <Row>
17           {products.map((product) => (
18             <Col key={product._id} sm={12} md={6} lg={4} xl={3}>
19               <Product product={product} />
20             </Col>
21           )));
22         </Row>
23         <Paginate page={page} pages={pages} keyword={keyword} />
24       </Container>
25     )}
26   </div>
27 );
```

Figure 2.19: Shop screen in React



```

1  @api_view(['GET'])
2  def getproducts(request):
3      query = request.query_params.get('keyword')
4      if query == None:
5          query = ''
6
7      products = Product.objects.filter(
8          name__icontains=query).order_by('-createdAt')
9
10     page = request.query_params.get('page')
11     paginator = Paginator(products, 8)
12
13     try:
14         products = paginator.page(page)
15     except PageNotAnInteger:
16         products = paginator.page(1)
17     except EmptyPage:
18         products = paginator.page(paginator.num_pages)
19
20     if page == None:
21         page = 1
22
23     page = int(page)
24     print('Page:', page)
25     serializer = ProductSerializer(products, many=True)
26     return Response({'products': serializer.data, 'page': page, 'pages': paginator.num_pages})
27

```

Figure 2.20: Shop screen in Django

Dans les captures d'écran ci-dessus, j'ai montré les fonctions frontend et backend utilisées pour afficher tous les produits du magasin à l'utilisateur.

2.2.2.4 Plan de repas de l'utilisateur

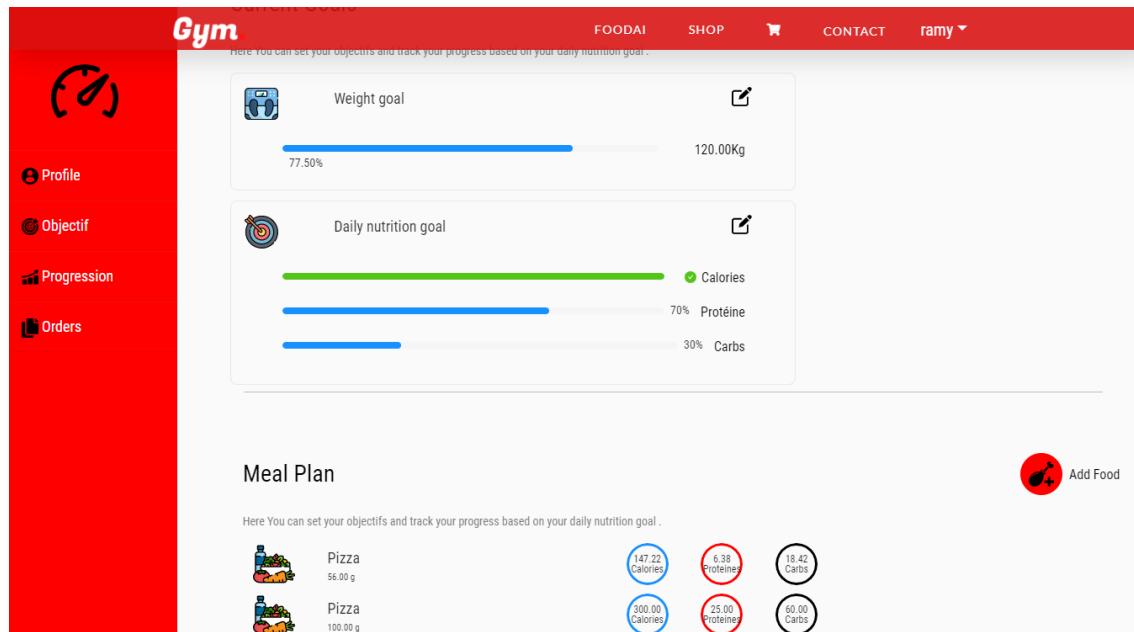


Figure 2.21: Plan-repas

À partir de cette page, l'athlète peut vérifier son objectif nutritionnel quotidien (calories, protéines, glucides) et ajouter son plan de repas en téléchargeant une image de

l'aliment ou en utilisant une simple recherche, puis comparer ses nutriments à l'objectif nutritionnel suggéré par l'entraîneur.

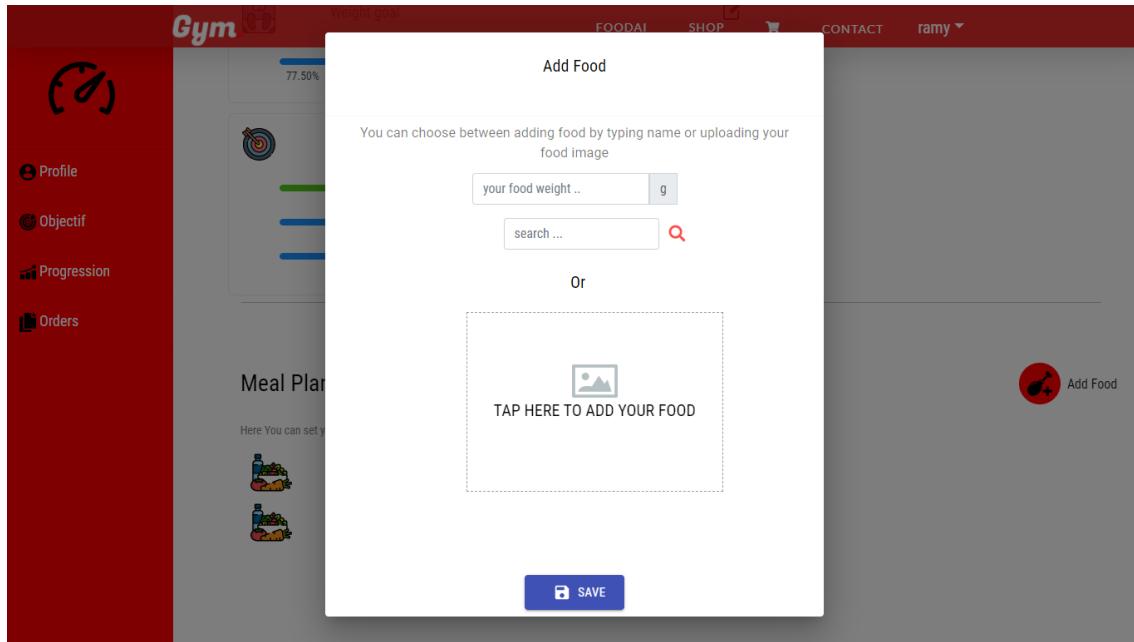


Figure 2.22: Ajouter un repas

2.2.2.5 Admin Gestion des utilisateurs

ID	Name	Email	Admin	Actions
>17	admin3@gmail.com	admin3@gmail.com	ADMIN	
>18	ramy2	ramyabadi3@gmail.com	USER	
>84	user	uspo@gmail.com	USER	
>85	user	uspo@gmail.co	USER	
>87	ramy	ramy@gmail.com	USER	

Figure 2.23: Table de tous les utilisateurs

L'entraîneur avec cette page a le contrôle total des utilisateurs, il peut rechercher n'importe quel utilisateur par (nom, email, id) puis voir toutes ses informations personnelles et il peut aussi les modifier ou les supprimer.

Pour le code source complet, consultez mon répertoire github et suivez toutes les instructions pour exécuter le projet : <https://github.com/Ramy-Ab/PFEV2>

Part V

Gestion de projet

Chapter 1

Travail Collaboratif

1.1 Introduction

Dans ce chapitre, je vais relater différents aspects liés à la gestion de projet. Je vais détailler le suivi du projet, notamment les outils utilisés, les interactions avec les différents superviseurs et le personnel d'Artec int. Je détaillerai également les livrables qui ont permis de tenir au chef de projet d'être informé de l'avancement du projet.

1.2 Outils collaboratifs

Je présente dans ce qui suit les différents outils collaboratifs utilisés pour permettre le partage de certaines ressources et le suivi détaillé du projet. Ressources avec mes superviseurs et mes collègues de l'esi-sba lorsque je cherche de l'aide.

1.2.1 Google Drive

Google Drive est un service de stockage et de partage de fichiers basé sur le cloud lancé par la société Google. Il inclut Google Docs, Sheets et Slides et Drawings, est une suite bureautique permettant d'éditer des documents, des feuilles de calcul, Je l'ai utilisé pour partager différents documents, y compris des rapports et diverses présentations.



1.2.2 GIT

Git est un logiciel de contrôle de version open source qui permet de suivre l'évolution du code source (principalement) et du travail d'équipe [17]. Je l'ai

utilisé pour faciliter l’application de la méthodologie de prototypage et pour garder suivi des modifications apportées. Je l’ai utilisé dans sa version 2.32.0 tout au long du cours.

1.2.3 Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), 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. Sans donc avoir besoin d’installer quoi que ce soit sur notre ordinateur à l’exception d’un navigateur. Je l’ai utilisé pour partager le code de mon modèle avec les superviseurs autorisés et pour entraîner mon modèle.

1.3 Suivi du projet

Ce projet a été suivi par deux entités : L’École Nationale d’Informatique de Sidi Bel Abbes (ESI-SBA) représentée par le responsable pédagogique (M. Belkacem Khaldi), et par la société société Artec int , représentée par le superviseur (M. Ahmed Bendjelloul).

1.3.1 Supervision à l’ESI-Sba

Mon objectif principal durant ce stage est de réaliser un travail complet et conforme aux qualités d’un travail réalisé par un ingénieur d’état de l’ESI-Sba, donc la supervision du côté de l’ESI-Sba était primordiale. Le suivi par le superviseur pédagogique (M. Khaldi) s’est fait par le biais de rapports résumant l’ensemble des tâches effectuées et à suivre, les résultats obtenus et les justifications théoriques les concernant, en utilisant Google Meet. Le rôle du superviseur de l’ESI-Sba était de valider les grandes lignes directrices du stage, valider les concepts théoriques, relire et valider le mémoire et surtout de s’assurer que le travail effectué était conforme aux exigences imposées par l’ESI-Sba.

1.3.2 Supervision à Artec Int

La supervision du côté d’Artec Int, plus précisément au sein du chef de projet M. Ahmed Bendjelloul, s’est déroulée en respectant les trois composantes suivantes respectant les deux composantes ci-dessous :

- L'assistance quotidienne : je me suis rendu au travail 3 jours par semaine, et j'ai consacré les autres jours à avancer sur tout ce qui concerne la recherche bibliographique. Progresser sur tout ce qui est recherche bibliographique et à apprendre quelques cours pour réaliser mon travail.
- Suivi des tâches : le superviseur nous accompagne moi pratiquement de 8h à 16h, et supervisait en moyenne, une réunion par semaine. Le superviseur vérifie l'avancement des tâches effectuées. Il participe également à la prise de décision sur les aspects conceptuels.

Chapter 2

Méthode de gestion

Au début de chaque projet, il est nécessaire de choisir une méthode de gestion à suivre qui permet de superviser le projet afin de respecter les délais et d'atteindre les objectifs fixés. Deux grandes approches sont utilisées dans le domaine de l'informatique : l'approche classique ou traditionnelle et l'approche agile. Une approche traditionnelle de type cycle en V ou cascade attend généralement une expression détaillée et validée par le client au début du projet, laissant peu de place aux changements. La réalisation dure le temps estimé sans que le client ait une idée de son produit. Souvent, un cahier des charges mal spécifications détaillées, un calendrier mal géré ou des risques non étudiés au début du projet risquent de le mettre en échec. Au contraire, l'approche Agile propose de réduire considérablement l'absence du client pendant la période de mise en œuvre en donnant plus de visibilité, en impliquant le client du début à la fin du projet. Les méthodes agiles adoptent un processus itératif et incrémental et elles doivent respecter quatre valeurs fondamentales représentées par l'agile :

- Les personnes et les interactions plutôt que les processus et les outils, - Des logiciels fonctionnels plutôt qu'une documentation cohérente, - Réagir au changement plutôt que de suivre un plan. J'ai opté pour une méthode agile pour la gestion de mon projet pour les raisons suivantes : - Le besoin n'était pas bien formulé au début du projet ; l'entreprise n'a pas fourni de spécifications détaillées mais seulement un aperçu. Des spécifications détaillées mais seulement une vue d'ensemble. Il est donc nécessaire d'impliquer le chef de projet tout au long du projet pour anticiper ses besoins.

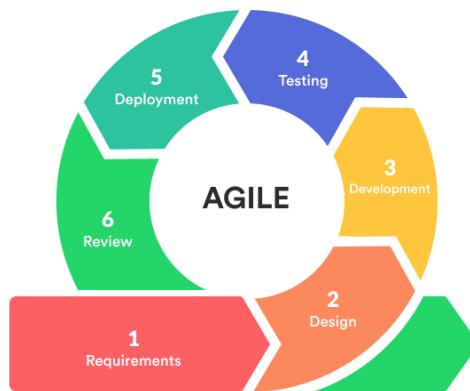


Figure 2.1: Le processus de méthode Agile

2.1 Scrum Method

Il existe plusieurs méthodes agiles, la plus connue étant la méthode Scrum que j'ai adopté.

Scrum est une méthode très simple et très efficace adoptée par les géants du marché : Microsoft, Google, Nokia ..., elle se caractérise par des sprints (itérations courtes) de (2) à (4) semaines avec des développements continus. dédié à la société Artec qui joue le rôle du client. La figure ci-dessous illustre le schéma général de cette méthode :

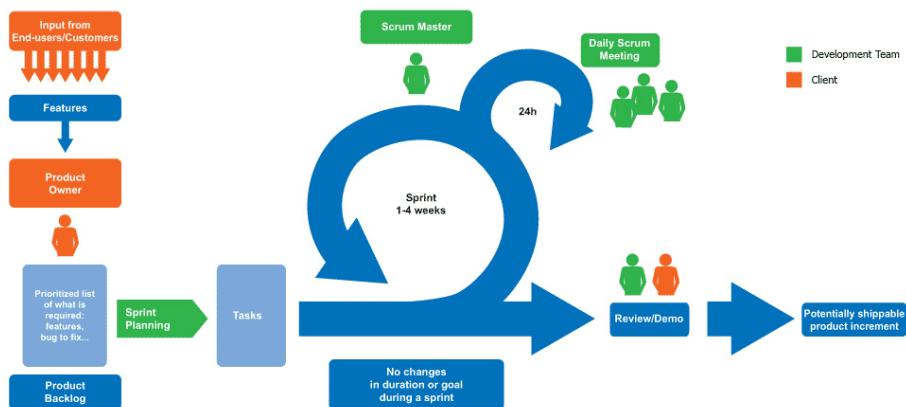


Figure 2.2: Le processus de scrum

2.2 Diagramme de Gantt

La macro-planification du projet est présentée par le diagramme de Gantt suivant développé lors du lancement du projet et modifié si nécessaire tout au long du projet. Également les principales étapes de la planification et leurs dates sont indiquées dans la deuxième figure.

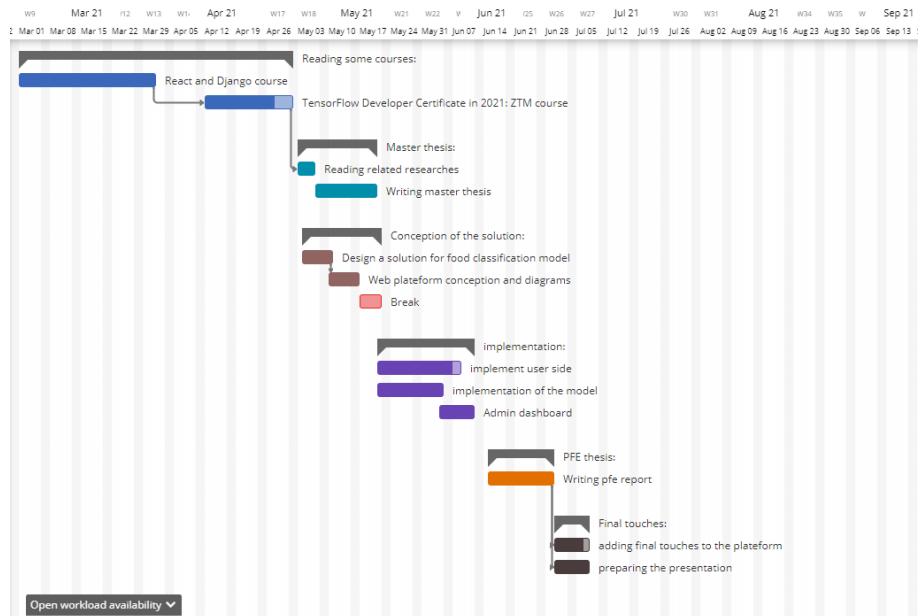


Figure 2.3: Diagramme de Gant

		ASSIGNEE	EH	START	DUET	%	
<input type="checkbox"/>	Search tasks...						←
<input checked="" type="checkbox"/>	Reading some courses:		-	01/Mar	01/May	92%	
1	<input checked="" type="checkbox"/> React and Django course	Unassigned	-	01/Mar	31/Mar	100%	
2	<input checked="" type="checkbox"/> TensorFlow Developer Certi...	Unassigned	-	12/Apr	01/May	80%	
	+ Add task + Add section						
<input checked="" type="checkbox"/>	Master thesis:		-	03/May	20/May	100%	
5	<input checked="" type="checkbox"/> Reading related researches	Unassigned	-	03/May	06/May	100%	
6	<input checked="" type="checkbox"/> Writing master thesis	Unassigned	-	07/May	20/May	100%	
	+ Add task + Add section						
<input checked="" type="checkbox"/>	Conception of the solution:		-	04/May	21/May	67%	
9	<input checked="" type="checkbox"/> Design a solution for food c...	Unassigned	-	04/May	10/May	100%	
10	<input checked="" type="checkbox"/> Web plateform conception ...	Unassigned	-	10/May	16/May	100%	
11	<input checked="" type="checkbox"/> Break	Unassigned	-	17/May	21/May	0%	
	+ Add task + Add section						
<input checked="" type="checkbox"/>	implementation:		-	21/May	11/Jun	96%	
14	<input checked="" type="checkbox"/> implement user side	Unassigned	-	21/May	08/Jun	90%	
15	<input checked="" type="checkbox"/> implementation of the model	Unassigned	-	21/May	04/Jun	100%	
16	<input checked="" type="checkbox"/> Admin dashboard	Unassigned	-	04/Jun	11/Jun	100%	
	+ Add task + Add section						
<input checked="" type="checkbox"/>	PFE thesis:		-	15/Jun	29/Jun	100%	
19	<input checked="" type="checkbox"/> Writing pfe report	Unassigned	-	15/Jun	29/Jun	100%	
	+ Add task + Add section						
<input checked="" type="checkbox"/>	Final touches:		-	30/Jun	07/Jul	93%	
22	<input checked="" type="checkbox"/> adding final touches to the ...	Unassigned	-	30/Jun	07/Jul	85%	
23	<input checked="" type="checkbox"/> preparing the presentation	Unassigned	-	30/Jun	07/Jul	100%	
	+ Add task + Add section						

Figure 2.4: Les étapes de planification

Chapter 3

Conclusion et travaux futurs

Ce chapitre résume les objectifs et les résultats atteints dans le projet. Les travaux futurs sera également décrit pour une meilleure solution au travaille .

3.1 Conclusion

Dans ce projet, j'ai créé une plateforme web spéciale pour un entraîneur personnel professionnel afin qu'il puisse proposer des programmes aux athlètes en fonction de leurs objectifs et qu'ils puissent suivre leurs progrès à travers la plateforme et connaître la valeur nutritionnelle de leurs aliments, puis la comparer avec la valeur recommandée par l'entraîneur en utilisant un algorithme de reconnaissance d'images alimentaires basé sur l'apprentissage profond pour améliorer la précision de l'évaluation diététique.

Les technologies évoluent et l'on est sans cesse obligé d'en apprendre des nouvelles pour rester à jour. C'est ce qui m'a motivé à l'accomplissement de ce projet avec les Frameworks React js ,django ,Rest framwork et aussi l'intégration d'un model deep learning dans mon plate-forme . En effet, je ne connaissais au début ce Framework que de nom, mais maintenant je peux me débrouiller et créer des plate-forme web facilement.

Dans ce rapport, j'ai fait une synthèse bibliographique des travaux qui m'ont paru pertinents dans ce domaine de la création d'un plateforme web pour des entraîneur professionnel et de la classification des aliments. A travers cette étude, j'ai constaté la richesse des travaux réalisés en termes de techniques,frameworks et d'algorithmes utilisés.

Pendant mon stage chez Artec Int, j'ai pu mieux comprendre comment travailler sur un projet réel. J'ai apprécié de travailler avec l'équipe d'Artec pour concevoir et mettre en œuvre différents problèmes logiciels. Cependant, il me reste encore beaucoup de chemin à parcourir pour acquérir les compétences nécessaires dans le domaine du développement.

Dans l'ensemble, j'ai trouvé l'expérience du stage en marketing positive, et je suis sûr que je pourrai utiliser les compétences acquises dans ma carrière ultérieure.

Ce travail est le dernier que je ferais à l'école national supérieur d'informatique et je pense avoir fièrement montré tout ce dont je suis capable, ainsi qu'une bonne partie des compétences apprises durant ces cinq années d'école.

3.2 Appréciation personnelle

Ce stage de fin d'études chez Artec Int a été une expérience très enrichissante, tant sur le plan personnel que professionnel. Et professionnelle. Il m'a permis, à la fois, de m'immiscer dans le monde de la recherche mais aussi dans celui du travail en grande entreprise. mais aussi dans le monde du travail au sein d'entreprises professionnelles. Elle m'a permis d'appliquer les connaissances acquises lors de ma formation à l'ESI-sba et de traiter des problèmes concrets du monde réel. Ayant acquis de nouvelles connaissances et compétences, je pense que ce stage a concrètement renforcé et amélioré ma formation d'ingénieur. renforcé et amélioré ma formation d'ingénieur dans mon école.

3.3 Travaux futurs

Cette section propose un cours de développement futur pour des améliorations à la solution actuelle ainsi que de nouvelles fonctionnalités pour améliorer les capacités de la solution et voila mes travaux futures :

- **Augmenter Dataset :** comme nous le savons, dans la formation d'un modèle d'apprentissage profond, l'ensemble de données a un effet important sur la précision du modèle. C'est pourquoi, pour les travaux futurs, il serait préférable d'utiliser plus d'ensembles de données dans la formation : UECFOOD256, UECFOOD100, FOOD50, FOOD85 ...
- **Food API :** comme je l'ai mentionné dans la partie précédente, j'ai déployé mon modèle dans django mais le problème est qu'à chaque fois que j'essaie de prédire un aliment, il charge le modèle à partir de zéro, ce qui prend beaucoup de temps. C'est pourquoi je pense l'héberger seul sur un serveur AWS et en faisant cela, je peux offrir une API gratuite pour tout le monde.
- **Amélioration du modèle :** mon modèle ne peut classer que les aliments. Dans le futur, je prévois de le rendre capable de segmenter l'image pour détecter tous les

aliments qu'elle contient, puis d'estimer le poids de chacun d'entre eux avec leur valeur nutritive et de produire le résultat final.

- **Système de recommandation :** je prévois d'ajouter un autre apprentissage profond pour ma plate-forme web afin de recommander au stagiaire le supplément approprié en fonction de nombreux faits pour obtenir les meilleurs résultats et aussi pour augmenter les ventes du coach .
- **Payement avec BaridiMob :** jusqu'à présent, le paiement se fait toujours à l'aide de la version d'essai de Paypal, mais j'essaie de le rendre plus pratique, comme l'entreprise me l'a demandé, en utilisant le paiement par BaridiMob.
- **Version Mobile :** en utilisant la technique de l'application web progressive qui offre une interface compatible avec les mobiles, ce qui nous permet de la faire fonctionner correctement sans avoir à développer une autre application mobile séparée.

References

- [1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [2] Lei Zhou, Chu Zhang, Fei Liu, Zhengjun Qiu, and Yong He. Application of deep learning in food: a review. *Comprehensive Reviews in Food Science and Food Safety*, 18(6):1793–1811, 2019.
- [3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.
- [4] Yufei Ma, Yu Cao, Sarma Vrudhula, and Jae-sun Seo. Optimizing the convolution operation to accelerate deep neural networks on fpga. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(7):1354–1367, 2018.
- [5] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.
- [6] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.
- [7] Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.
- [8] Nikhil Ketkar. Introduction to keras. In *Deep learning with Python*, pages 97–111. Springer, 2017.
- [9] Bernadette M Randles, Irene V Pasquetto, Milena S Golshan, and Christine L Borgman. Using the jupyter notebook as a tool for open science: An empirical study. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–2. IEEE, 2017.
- [10] Jeffrey M Perkel. Why jupyter is data scientists’ computational notebook of choice. *Nature*, 563(7732):145–147, 2018.

- [11] Akhil Kadiyala and Ashok Kumar. Applications of python to evaluate environmental data science problems. *Environmental Progress & Sustainable Energy*, 36(6):1580–1586, 2017.
- [12] Sanchit Aggarwal. Modern web-development using reactjs. *International Journal of Recent Research Aspects*, 5(1):133–137, 2018.
- [13] Jeff Forcier, Paul Bissex, and Wesley J Chun. *Python web development with Django*. Addison-Wesley Professional, 2008.
- [14] Behandelt PostgreSQL. Postgresql. Web resource: <http://www.PostgreSQL.org/about>, 1996.
- [15] Kazuaki Maeda. Performance evaluation of object serialization libraries in xml, json and binary formats. In *2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP)*, pages 177–182. IEEE, 2012.
- [16] Michael Jones, Brain Campbell, and Chuck Mortimore. Json web token (jwt) profile for oauth 2.0 client authentication and authorization grants. *May-2015.{Online}*. Available: <https://tools.ietf.org/html/rfc7523>, 2015.
- [17] Jon Loeliger and Matthew McCullough. *Version Control with Git: Powerful tools and techniques for collaborative software development*. " O'Reilly Media, Inc.", 2012.