ÉCOLE NATIONALE D'INGÉNIEURS DE TUNIS

Département Technologies de l'information et de la Communication

Rapport de stage ouvrier

Conception d'un modèle de vision artificielle pour le tri des amandes

Réalisé par :
Makni Eya
2ème Année informatique 1

Encadré par :
M. Fourati Amine

au sein de la société

SERVICE DIGITAL
Service Digital pour
l'industrie & l'agriculture

Année universitaire : 2020-2021

Signature

Encadrant : M. Fourati Amine

Remerciements

Dans le cadre de ce stage, je tiens à remercier toutes les personnes qui ont collaboré au succès de mon stage ouvrier.

Tout d'abords, je commence par remercier mon maître de stage M.Amine Fourati, le directeur RD de service digital pour l'Industrie pour son accueil, le temps passé ensemble et le partage de son expertise. Grâce aussi à sa confiance, sa disponibilité et son assistance durant toute ma période de stage, j'ai pu m'accomplir toutes mes missions aux temps convenus.

Je tiens à remercier vivement M. Amine Moalla, le directeur du Système Management Qualité qui m'a aidé à postuler dans ce stage qui était en totale adéquation avec mes attentes et qui m'a trouvé à la hauteur de cette candidature.

Je remercie également toute l'équipe de service digital pour l'industrie pour leur accueil, leur esprit d'équipe et en particulier Abir Rachdi ma collègue en 3ème année électronique de communication à l'École Nationale d'Électronique et des Télécommunications de Sfax qui m'a coopéré dans l'élaboration de ce projet lors de sa réalisation de son stage PFE.

Un grand merci à ma mère et mon père, pour leurs conseils, ainsi que leurs soutiens moral et économique.

Mon dernier mot s'adresse à tous les membres du jury pour leur exprimer toute ma gratitude.

Table des matières

Table des figures

Liste des tableaux

Liste des accronymes

DL : Deep Learning

SNN : Standard Neural Network

CNN : Convolutional Neural Network

RNN : Recurrent Neural Network

Hybrid NN : Hybrid Neural Network

Liste des définitions

b : Le bias est une entrée supplémentaire du neurone qui vaut toujours 1, et a son propre poids de connexion. Cela garantit que le neurone sera activé même si toutes les entrées ne sont pas toutes à 0.

w : Le poids est attribué à chacune des entrées, permettant de décider et d'ajuster l'importance de certaines par rapport aux autres. Il représente la force de la connexion entre les unités.

m : La taille du set d'apprentissage ou le nombre d'exemples d'apprentissages.

y : C'est la valeur actuelle de la sortie du neurone.

$\hat{y}$: C'est la valeur prédite de la sortie du neurone.

α : Le taux d'apprentissage détermine à quelle vitesse les poids optimaux du modèle sont calculés.

λ : Le paramètre de régularisation

Introduction générale

Au cours de ces dernières décennies, l'industrie affronte deux problèmes principaux : d'une part, les consommateurs recherchent de plus en plus de personnalisation des produits, de l'autre part le manque de main-d'œuvre devient un problème croissant au niveau de la production. Les industriels doivent s'adapter à un environnement en constante évolution, où la flexibilité et la satisfaction du client sont essentielles.

En effet l'intelligence artificielle se donne comme tâche de créer des machines qui pourront apprendre afin de pouvoir exécuter de façon satisfaisant les tâches qu'un humain pourrait aussi bien accomplir.

C'est dans cette optique d'idée que se situe notre stage ouvrier chez « Service Digital». Il s'agit de développer un système de vision artificielle, rapide, fiable et tend vers "zéro faute" qui sera utilisé dans l'industrie agroalimentaire, plus précisément dans le tri des amandes ; car cette opération reste encore longue et pénible à cause des méthodes manuelles utilisées. La vision artificielle, ou vision par ordinateur ou encore computer vision en anglais a pour but de capturer les images, les analyser, les traiter et de les comprendre afin de pouvoir actionner sur une décision prise.

Ce rapport comporte trois chapitres détaillant le travail réalisé.

— **Chapitre 1 : Contexte du travail et cahier des charges :** ce chapitre sera consacré à la présentation de l'entreprise d'accueil, à la définition du cahier des charges et l'environnement de travail.

— **Chapitre 2 : Formation et Analyse :** il sera focalisé sur la présentation des diverses notions nécessaires pour la conception du projet et la réalisation du travail demandé.

— **Chapitre 3 : Conception technique et réalisation :** dans ce dernier chapitre il y aura une présentation des exigences techniques, collection des données, présentation des modèles, analyse de leurs résultats et enfin une étude comparative.

Chapitre 1

Contexte du travail et cahier de charge

Introduction

Le but de ce chapitre est de présenter l'organisme d'accueil, le cadre de stage et le cahier des charges de ce travail.

1.1 Service Digital pour l'Industrie

Dans cette section, la startup sera définie par une présentation, leurs axes stratégiques, leurs missions et leurs produits.

1.1.1 Service Digital

Service Digital est un bureau de "Recherche et Développement" et de Consulting dans le domaine de l'industrie et l'agriculture. Elle repose sur l'ingénierie et l'implémentation des solutions high tech pour le système de production de demain. Cette société est fondée par M.Amine Fourati, Un data scientist en agronomie et agroalimentaire, et M.Amine Moalla qui est un ingénieur en génie industriel.

FIGURE 1.1 – Logo de la société Service Digital

Cette société a fait un partenariat avec CTRACOM, une société destinée à la construction métallo mécanique d'équipements industriels, en 2020. Ce groupe est appelé CO_UP.

FIGURE 1.2 – Logo CTRACOM

FIGURE 1.3 – Logo du groupe CO_UP

1.1.2 Les axes stratégiques de Service Digital

Les principaux axes de Service Digital sont l'innovation, l'engagement, la confiance, le service après-vente et surtout la qualité.

1.1.3 Les missions de Service Digital

Les missions de cette société se traduisent dans La réalisation des projets innovants et des solutions destinées aux domaines de l'industrie et l'agriculture à fin d'offrir des services et des produits complets, aussi, la mise en place et le déploiement du lean management 5 S, SMED, TPM, KANBAN) et la méthode SCRUM pour la gestion des projets agiles et enfin l'étude de la demande client pour diagnostiquer auditer et mettre en place les systèmes de management selon les référentiels.

1.1.4 Les produits de Service Digital

Le groupe CO_UP est spécialisé dans la production des machines industriel en général et plus spécifiquement dans le domaine agricole et agroalimentaire. Leur produit franchit différentes étapes avant d'être livré chez leurs clients. La première phase est la recherche. À l'issue de cette étape il se trouve avec une solution adaptée, innovante et actualisée technologiquement juste après, il développe un prototype qui matérialise la solution trouvée ces deux étapes se déroulent au sein du département RD. Une fois le prototype est fini le produit passe à l'étape étude, conception et simulation de la machine ensuite vers la mise en place du procédé de fabrication au sein de leur usine. Toutes ces étapes suivent la démarche qualité ISO 9001 en lien avec la méthode AGILE SCRUM et en inspirant de la méthode lean manufacturing.

1.2 Présentation du projet

Cette partie contient une explication du projet, la problématique et les tâches à réaliser.

1.2.1 Contexte du projet

La société Service Digital pour l'Industrie a pour but de réaliser une machine de triage d'amande. Le rôle de cette machine commence après la phase de la machine concasseuse d'amande. Elle consiste à trier les amandes saines, cassées et les déchets. Ce projet contient diverses parties comme la partie électrique, pneumatique, informatique. Pour cette raison ma collègue Abir Rachdi a réalisé une étude détaillée lors de sa préparation de son projet de fin d'année.
En revanche, à cause de la pandémie du virus covid-19, elle n'a pas pu de réaliser toutes les parties de ce projet mais elle a concentré sur le côté informatique. Dans ce stade, le but de ce stage consiste à réaliser un modèle de classification d'amande en utilisant les techniques de deep learning.

1.2.2 Problématique

Après une discussion prolongée avec les directeurs de Service Digital, nous avons tiré les grands objectifs de ce stage :

— Trouver un modèle de classification qui donne des indices de précision et de perte les plus convenables.

— Avoir un modèle qui ne prend pas beaucoup de temps lors de la classification.

— Le modèle ne doit pas considérer les bonnes amandes comme un déchet.

Pour la meilleure réalisation du projet, il faut suivre des étapes bien définie qui va être divisée tout le long de 2 mois de stage :

— Trouver un modèle de classification qui donne des indices de précision et de perte les plus convenables.

— Avoir un modèle qui ne prend pas beaucoup de temps lors de la classification.

— Le modèle ne doit pas considérer les bonnes amandes comme un déchet.

De nos jours, le souci majeur de toutes les entreprises c'est la satisfaction des clients spécialement lorsqu'il s'agit des entreprises offrant des services en contact direct avec ceux-ci. Connaissant que 60% des gens font leurs courses en dehors des heures de travail et dans le weekend, celles-ci s'équipent par des agents conversationnels à savoir les chatbots pour mieux répondre à ces problématiques et les banques ne font pas exception.

1.2.3 Solution proposée

Afin d'achever les souhaits de WIFAKbankbot et le IWEALTHBot il est nécessaire de fixer les objectifs et étudier la faisabilité technique du projet

1.2.3.1 Objectifs de solution proposée

L'objectif principal de ce projet est de concevoir un chatbot en se basant sur les fonctionnalités suivantes :

— **Une formation :**
Une formation sur l'intelligence artificielle et le deep learning est nécessaire pour avoir un certain niveau de connaissance.

— **Recherche de modèle :**
Une recherche sur les divers modèles de classification d'images déjà utilisées pour d'autres concepts.

— **Choix de modèle :**
Choisir les modèles les plus adéquats suivants des critères bien définis.

— **Création des modèles spécifiques :**
En s'appuyant sur les modèles choisis, créer des modèles spécifiques pour nos besoins.

— **Training et testing :**
Trainer les modèles et tester des exemples.

— **Tuning :**
Faire le tuning des modèles afin d'avoir les meilleurs résultats.

— **Etude comparative :**
Réaliser une étude comparative entre les modèles trouvés.

1.3 L'environnement du travail

Pendant toute ma période de deux mois dans cette startup, j'ai essayé le monde de travail pour ma première fois. C'était une expérience qui me permet de découvrir le vrai sens de la responsabilité, d'être engagé pour faire et rendre les tâches au temps, de travailler en groupe et avec des personnes que je ne les connais pas et de savoir interagir avec mes supérieurs.
Par conséquent, Service Digital a adopté une méthode de gestion de projets appelée Agile Scrum.

1.3.1 Approche Agile

Parmi les méthodes itératives, nous pouvons distinguer les paradigmes AGILES largement utilisés de nos jours à travers le monde. L'état d'esprit AGILE est mené dans un esprit collaboratif avec le client et s'adapte aux approches incrémentales. Elle engendre des produits de haute qualité tout en tenant compte de l'évolution des besoins du client.

1.3.2 La méthode Scrum

SCRUM est le cadre de gestion de projets agiles le plus utilisée de nos jours pour les projets de développement informatique. C'est une méthode d'organisation de projet en cycles de développement itératifs et incrémentaux.
La réunion de la réalisation du backlog était le premier contact avec :

- **Le product owner :** qui est responsable à la réalisation de backlog et la détermination du user story à réaliser avec l'équipe.
- **Le Scrum master :** facilite le dialogue et le travail entre les différents intervenants et responsable du scrum daily meeting et la revue de sprint.
- **L'équipe de développement :** chargés de transformer le besoin du client exprimer sous forme de user stories en fonctionnalité réelle opérationnelle.

L'outil scrum utilisé par digital service pour contrôler le déroulement du projet est un tableau des tâches formé de 5 colonnes : Backlog, A faire, en cours, A vérifier et Terminer. Chaque tâche se déplace d'une colonne à une autre après les réunions quotidiennes où nous avons faite chaque jour pendant 5 minutes avant de commencer le travail.
Je trouve que cette méthode facilite l'organisation et la répartition du travail entre moi et ma collègue. De plus, nous étions toujours en courant des besoins de notre encadrant et de s'adapter aux changements plutôt que le suivie du plan.
D'ailleurs, le suivie de ma progression par mes encadrants était en utilisant l'application **Trello** comme un outil d'organisation collaboratif. En outre, il est utile pour faire une autoévaluation et savoir l'avancement et les capacités personnel quotidiennement.

Conclusion

Pour conclure, je pense que c'était une bonne occasion pour moi de tester la vie professionnelle et d'expérimenter une telle technique utile et populaire surtout dans le domaine de l'informatique.

Chapitre 2

Formation et Analyse

Introduction

Dans ce chapitre, je m'intéresse à la formation que j'ai faite au sein de l'entreprise et fais une analyse au projet.

2.1 Formation deep learning

Pendant le premier mois de ce stage ouvrier, j'ai fait une formation sur la spécialisation en deep learning pour avoir les notions de base nécessaires pour la réalisation de ce projet.

2.1.1 Réseau de neurones et deep learning

Le terme deep learning fait référence à la formation des réseaux de neurones qui peut être simple comme il peut être très grand et compliqué. Un réseau de neurones est formé d'un ensemble d'unités cachées, qui sont les neurones, et qui prennent des données X comme des entrées. Après une ou plusieurs couches de neurones, tous les résultats des unités cachés de l'avant-dernière couche entrent dans la couche de sortie où on a une fonction de régression. Cette dernière fait sortir un seul résultat Y.Alors, en fournissant suffisamment de données sur X et Y et en donnant suffisamment d'entraînement avec X et Y, les réseaux de neurones sont remarquablement bons pour la déterminer les fonctions qui mappent avec précision de X à Y.

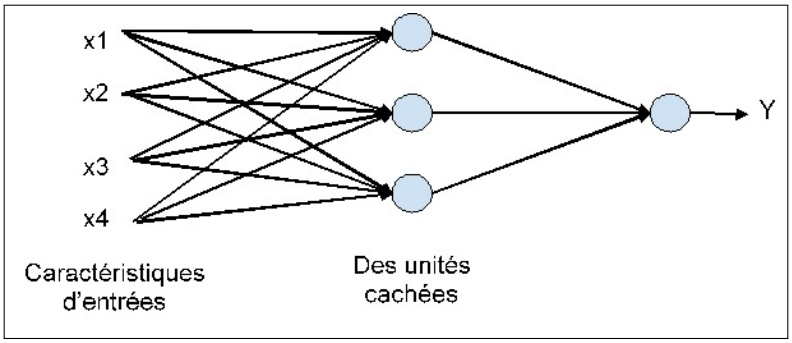

FIGURE 2.1 – un réseau de neurone

2.1.2 Les différents types des réseaux de neurones pour l'apprentissage supervisé

Dans le domaine du deep learning, les chercheurs ont trouvé divers types de réseaux de neurones selon le besoin. Pour cette raison, nous trouvons 4 catégories qui sont :

- **SNN :**qui est un réseau standard acyclique. C'est-à-dire il est formé par des principalement pour des applications comme le choix à visualiser des publicités en ligne pour les internautes ou la détermination des prix des immobiliers.
- **CNN :**est un réseau de neurones dédié essentiellement pour les problèmes qui traitent les images comme la classification d'images ou la reconnaissance faciale.
- **RNN :**ce réseau spécial pour les problèmes de données séquentielles comme l'audio car il est formé par des composantes temporelles et représenté comme une série à une dimension temporelle. Aussi il est adéquat pour les problèmes de translation des langues.

— **Hybrid NN :**ce sont des réseaux plus spécifiques ou ont une architecture plus complexe ou hybride comme les conduites autonomes.

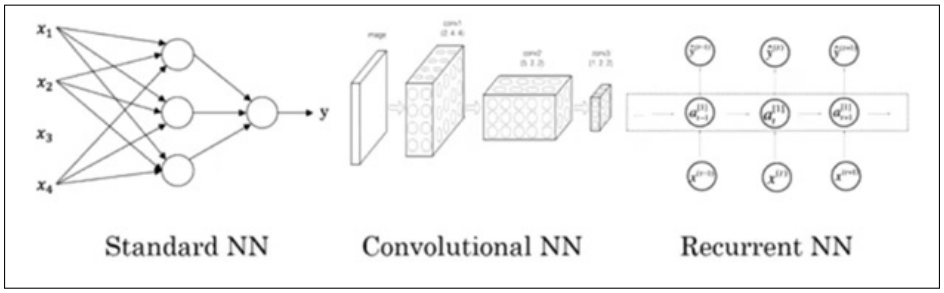

FIGURE 2.2 – les formes des différents réseaux

2.1.3 La regréssion logistique

C'est une méthode de prédire des valeurs prises qualitativement à partir des variables prises quantitativement ou qualitativement. Les fonctions les plus utilisées pendant le processus de la régression logistique sont :

— **La fonction de perte :**
La fonction de perte où Loss function est une fonction qui mesure la performance d'un seul exemple d'apprentissage par notre algorithme. C'est-à-dire elle mesure à quel point notre sortie y hat est bonne quand l'étiquette est y.

$$L(\hat{y}, y) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{2.1}$$

— **La fonction de coût :**
La fonction de coût ou Cost function est une fonction pour mesurer la performance sur le set d'apprentissage en entier.

$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right) \tag{2.2}$$

Lors de l'organisation du calcul dans le réseau de neurones, on utilise un passage direct ou ce qu'on appelle étape de propagation avant, suivi d'un passage en sens inverse appelé propagation inverse ou rétropropagation.
Ces actions sont nécessaires pour trouver la valeur des paramètres w et b du modèle qui donnent la valeur minimum de la fonction du coût. Dans ce cas, la descente de gradient est la bonne méthode à suivre tel que J(w, b) est la surface d'une fonction convexe et elle représente la hauteur de la surface à un certain point.Les bonnes valeurs de w et b correspondent à la valeur minimale de J(w, b). Cette procédure se fait sur 2 étapes :

1. Initialiser des valeurs de w et b soit à zéro soit à des valeurs aléatoires parce que pour la fonction convexe quels que soient w et b initialisés on trouve les mêmes valeurs.
2. Appliquer la descente de gradient plusieurs fois jusqu'à atteindre l'optimum global ou une valeur de proche.

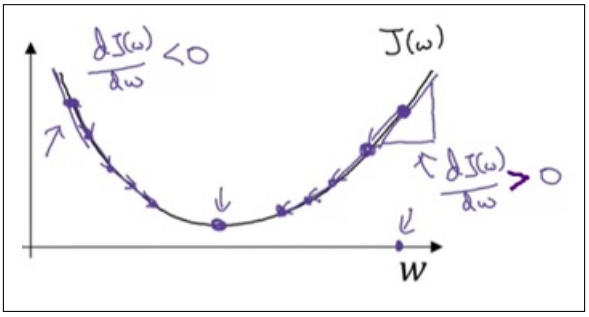

FIGURE 2.3 – l'optimum globale de J(w, b)

La recherche des valeurs des paramètres w et b se fait par les équations suivantes :

$$\text{Pour } J(\omega, b)$$
$$\text{Repeat } \{$$
$$w := w - \alpha\partial\frac{J(w,b)}{\partial w} \tag{2.3}$$
$$b := b - \alpha\frac{\partial J\,(\omega_1 b)}{\partial b}\}$$

2.1.4 Les fonctions d'activation

Il y a plusieurs fonctions d'activation qui peuvent être utilisées dans les réseaux de neurones :

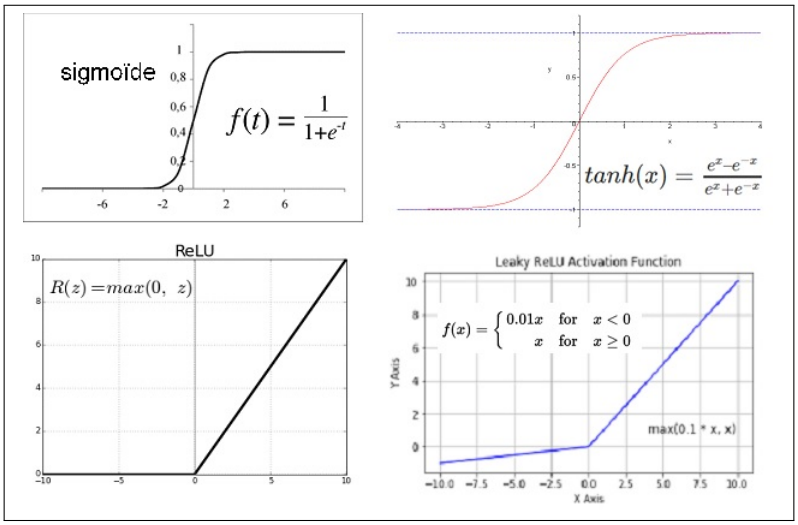

FIGURE 2.4 – Les fonctions d'activation

2.1.5 Les données

Le processus du deep learning est très empirique vu qu'il faut essayer plusieurs trucs et observer la méthode qui fonctionne mieux. Cette démarche nécessite une boucle itérative qui commence par l'obtention d'une idée puis la formule sous forme d'un code et enfin on commence la partie expérimentale. Un seul enchainement peut prendre 10 minutes comme il peut prendre un mois.

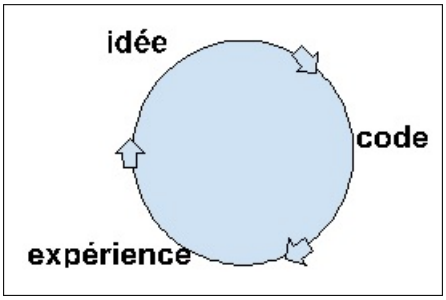

FIGURE 2.5 – Le cycle du processus

En conséquence, il nous faut une base de données pour faire apprendre notre modèle. Cette base doit être divisée en 3 parties :

— Un ensemble d'apprentissage (training set) : C'est un ensemble de données permettant d'adapter les paramètres du modèle. ensemble de validation (Dev set) : cet ensemble permet de décider lequel des différents modèles est le plus performant avec ces données.

— Un ensemble d'essais (test set) : après avoir fini toute la boucle et lorsque le modèle final est prête, on l'évalue avec l'ensemble de données d'essai afin d'obtenir une estimation impartiale de la performance de l'algorithme.

La quantité des données de chaque catégorie doit être bien choisie. Par exemple, à l'ère du big data moderne, lorsqu'il y a des millions d'exemples, 1%s des données pour le Dev set et 1% pour le test set sera largement suffisant à condition qu'ils soient les deux de la même distribution. Mais, le train set doit avoir la plus grande partie des données (98%).

2.1.6 Biais et Variance

Biais :
Le biais est un neurone supplémentaire dans chaque couche et stocke la valeur « 1 » pour chaque action. L'erreur de biais est une erreur d'hypothèses erronées dans l'algorithme d'apprentissage.

Variance :
La variance est une erreur de sensibilité à de petites fluctuations dans l'ensemble d'apprentissage.

caption

	Elevé	**Faible**
Biais	Le modèle ne s'adapte pas bien à l'ensemble d'apprentissage et l'errer de formation (training error) sera importante.	Le modèle est bien accordé et l'erreur de formation est faible.
Variance	Le modèle ne peut pas faire des prédictions précises sur l'ensemble de validation et l'erreur de validation sera importante.	Le modèle capable de sortir de ses données d'entraînement.

TABLE 2.1 – La variation du biais et de la variance

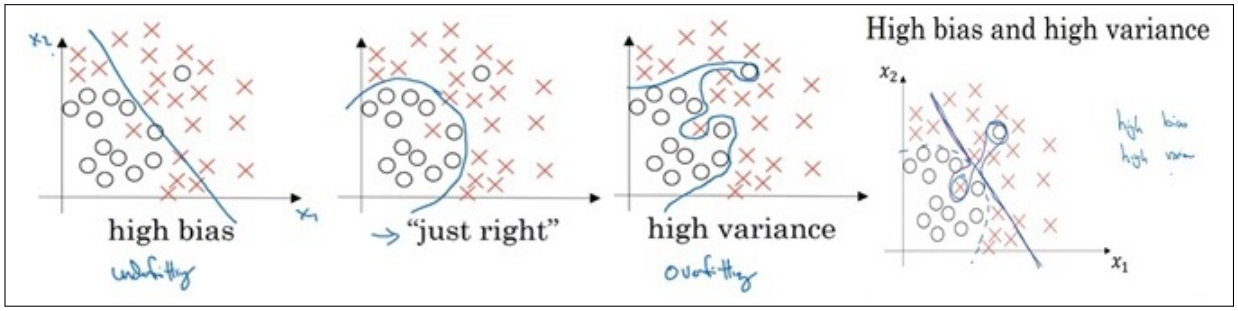

FIGURE 2.6 – biais de réseau neuronal - faible variance et variance élevée

Les méthodes de correction de biais élevé :

— Utiliser un réseau de neurones plus grand.

— Faire plus d'entraînement ou essayer des algorithmes d'optimisation plus avancés.

— Rechercher d'autres architectures de réseau neuronal.

Les méthodes de correction de la variance élevée :

— Ajouter plus de données.

— Faire la régularisation.

— Rechercher d'autres architectures de réseau neuronal.

2.1.7 La régularisation et ces méthodes

La régularisation est un changement au niveau de la fonction de coût ou qui s'appelle aussi la fonction d'erreur pour fixer les problèmes de surajustement ou de sous-ajustement des données d'apprentissage.
Les méthodes de la régularisation sont :

2.1.7.1 L2 regularization

$$f(w,b) = \frac{1}{m}\sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right) + \frac{\lambda}{2m} * \|w\|_2^2 \tag{2.4}$$

2.1.7.2 Dropout

En outre de L2 regularization, il y a la méthode de dropout. Elle consiste à :

1. Pour chaque couche du réseau, on définit une probabilité aléatoire d'élimination des nœuds.
2. Supprimer toutes les sorties des nœuds éliminés.
3. Refaire la boucle d'apprentissage en ajoutant et supprimant d'autres nœuds.

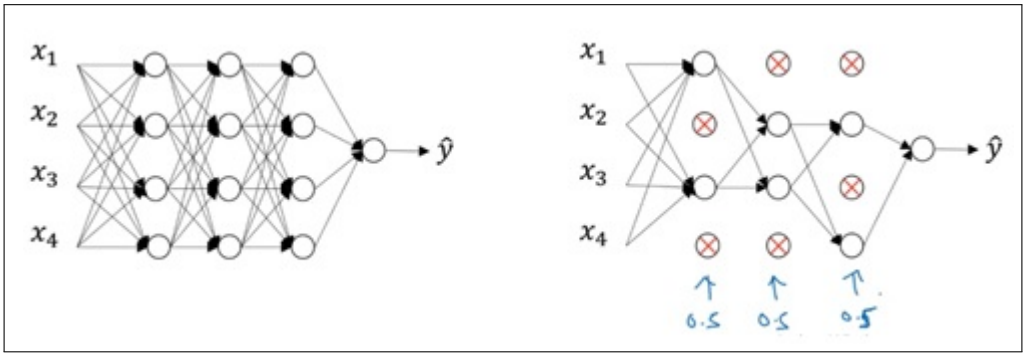

FIGURE 2.7 – Dropout

2.1.7.3 Augmentation des données (Data augumentation)

Pour augmenter notre base de données en cas d'insuffisance, on peut retourner des images horizontalement, prendre des recadrages aléatoires de l'image, prendre des distorsions aléatoires et des traductions pour les caractères optiques. . . .

2.1.7.4 Early stopping

Early stopping ou l'arrêt précoce est une méthode de régularisation qui permet d'ignorer le surajustement lors de l'utilisation d'une méthode itérative. Après l'initialisation du paramètre w, on commence l'apprentissage du modèle. Au début, les performances du modèle s'améliorent mais après un certain nombre d'itérations il reste inchangeable. Dans ce cas, on utilise la méthode de l'arrêt précoce.

2.1.8 Les méthodes d'optimisation de la descente de gradient

Pour atteindre le minimum de la descente de gradient, on obtient des oscillations lentes et encombrées ce qu'il empêche l'utilisation d'un taux d'apprentissage important. Pour cette raison, il y a plusieurs méthodes d'optimisation de la descente de gradient qui diminue le nombre d'oscillations et visent l'objectif directement comme :

— Momentum gradient descent

— RMSprop

— Adam gradient descent

2.2 Architecture d'un CNN

Une architecture CNN est formée par une succession de couches différentes et variées telle que chaque une à un rôle spécifique :

— La couche de convolution qui traite les données d'un champ récepteur.

— La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).

— La couche de correction (Relu), souvent appelée par abus "Relu'" en référence à la fonction d'activation (Unité de rectification linéaire).

— La couche "entièrement connectée" (Fully Connected : FC), qui est une couche de type perceptron.

— La couche de perte (LOSS).

2.2.1 Couche de convolution

La couche de convolution est le bloc de construction de base d'un CNN, son but principal est d'extraire des entités de l'image d'entrée. La convolution préserve la relation spatiale entre les pixels en apprenant les caractéristiques de l'image à l'aide de petits carrés de données d'entrée. Quand on lui présente une nouvelle image, le CNN ne sait pas exactement si les fonctionnalités seront présentes dans l'image où elles pourraient exister, il essaie donc de les trouver n'importe où dans l'image entière. En calculant dans toute l'image si une caractéristique est présente, nous faisons un filtrage. En utilisant un filtre sur les données d'entrée, nous créons une carte de fonctionnalité comme montre la figure.

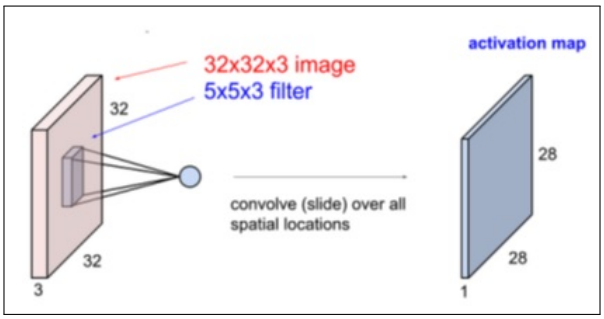

FIGURE 2.8 – La couche de convolution

2.2.2 Couche de ReLU

La partie importante de l'ensemble du processus est l'unité linéaire rectifiée ou ReLU. Le calcul derrière ce concept est très simple : chaque fois qu'il y a une valeur négative dans le pixel, elle est remplacée par 0. Cette fonction force les neurones à retourner des valeurs positives.

2.2.3 Couche de Pooling (POOL)

Le Pooling, comme la montre la figure 2.8, est une méthode permettant de réduire la taille des images, tout en préservant les informations les plus importantes qu'elles contiennent, mais aussi de s'affranchir du phénomène d'overfitting ». Tout comme la convolution, le calcul derrière ce concept n'est pas très compliqué. En fait, il suffit de faire glisser progressivement une petite fenêtre sur toutes les parties de l'image et de prendre la valeur maximale de la fenêtre à chaque étape. Après avoir procédé au pooling, l'image ne représente qu'un quart de son nombre de pixels de départ.Parce qu'il garde à chaque pas la valeur maximale contenue dans la fenêtre et il conserve les meilleures caractéristiques de l'image. En ayant moins d'informations spatiales, on obtient plus des performances de calcul, moins de paramètres, donc moins de chances de sur-apprentissage.

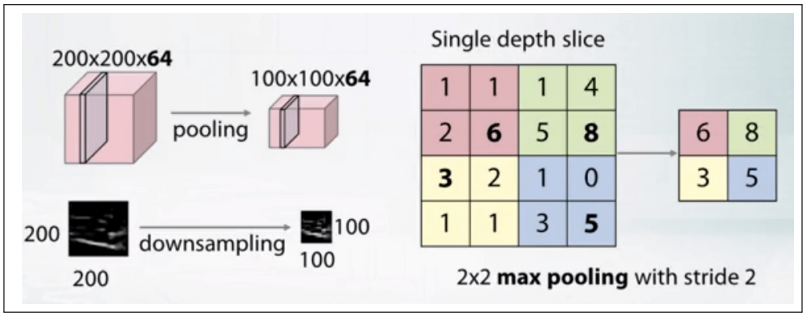

FIGURE 2.9 – La couche de pooling

2.2.4 Couche entièrement connectée (FC)

Une couche entièrement connectée dans un réseau neuronal est une couche dans laquelle toutes les entrées d'une couche sont connectées à chaque unité d'activation de la couche suivante. Dans les modèles d'apprentissage automatique les plus courants, la dernière couche est une couche entièrement connectée qui compile les données extraites par la couche précédente pour former la sortie finale.

2.2.5 Couche de perte (LOSS)

La fonction de perte fournit non seulement une représentation statique des performances du modèle, mais également la manière dont l'algorithme ajuste les données en premier lieu. La plupart des algorithmes d'apprentissage automatique utilisent une sorte de fonction de perte dans le processus d'optimisation ou de recherche des meilleurs paramètres (poids) des données. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La fonction « Softmax » permet de calculer la distribution de probabilités sur les classes de sortie.

Conclusion

Ces connaissances des notions basiques sur le deep learning, les réseaux de neurones et plus précisément les réseaux de neurones convolutives sont nécessaires pour la réalisation de projets de stage. Pour cette raison, cette étude a été entamé dans ce chapitre.

Chapitre 3

Conception technique et réalisation

Introduction

Dans ce chapitre, nous allons se concentrer sur le côté conception et réalisation du projet. Ainsi, nous allons commencer par donner les outils utilisés pour aboutir à notre objectif suivi d'une analyse du travail fourni et du résultat.

3.1 Environnement du travail

Dans cette section nous allons énumérer les environnements matériels et logiciels utilisés pour la réalisation de projets.

3.1.1 Environnement de développement matériel

L'environnement de notre travail est une machine du type PC Lenovo dont les caractéristiques sont les suivantes :

— Processeur : Intel (R) Core^{TM} i5-8250U CPU @ 1.60GHz 1.80GHz

— RAM : 8,00Go

— Système d'exploitation : Windows 10 Professionnel

3.1.2 Environnement de développement logiciel

Dans cette section, nous nous intéressons à présenter les outils et les frameworks utilisés pour aboutir à la réalisation de notre approche.

3.1.2.1 Langages

— **Python :** est un langage de programmation de haut niveau largement utilisé pour la programmation générale, créé par Guido van Rossum et publié en 1991. Un langage interprété, Python met l'accent sur la lisibilité du code et surtout l'optimisation des nombres des lignes de code chose n'est pas disponible par d'autres langages comme C++ et Java.

FIGURE 3.1 – Logo Python

— **Kaggle :** est une communauté en ligne de data scientists et de praticiens de machine learning. Il permet aux utilisateurs de trouver et de publier des ensembles de données, d'explorer et de créer des modèles dans un environnement de science de données basés sur le web, de travailler avec d'autres chercheurs en données et ingénieurs en apprentissage automatique, et de participer à des compétitions pour résoudre les défis de la science de données.Il fournit également un GPU Nvidia P100 gratuit pour entraîner ses modèles, ce qui est essentiel pour l'analyse d'image si on veut avoir un temps de calcul qui reste raisonnable. Cela élimine également le besoin de poste de travail pour développer et terminer tous les travaux via un ordinateur portable

connecté à Internet.Il est possible d'exporter le travail effectué sur cet environnement de développement afin de l'utiliser où l'on le souhaite.
L'export est soit un fichier .ipynb ou un script Python.

FIGURE 3.2 – Logo Kaggle

— **GitHub :** C'est un outil gratuit pour héberger du code open source, et propose également des plans payants pour les projets de code privés.

FIGURE 3.3 – Logo GitHub

3.1.2.2 Langages

— **TensorFlow :** TensorFlow TM est une bibliothèque de logiciels opens source pour le calcul numérique utilisant des graphiques de flux de données. Les noeuds dans le graphe représentent des opérations mathématiques, tandis que les arêtes de graphe représentent les matrices de données multidimensionnelles (tenseurs) communiquées entre eux. L'architecture flexible nous permet de déployer le calcul sur un ou plusieurs processeurs ou GPU dans un poste de travail, un serveur ou un périphérique mobile avec une seule API.

FIGURE 3.4 – Logo TensorFlow

— **Keras :** est une API de réseaux neuronaux de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow, CNTK ou Theano comme backend. Il a été développé dans le but de permettre une expérimentation rapide. Pouvoir passer de l'idée au résultat avec le moins de retard possible est la clé pour faire de bonnes recherches.

FIGURE 3.5 – Logo Keras

— **Numpy :** NumPy est le package fondamental pour l'informatique scientifique avec Python. Il contient entre autres choses :
  — Un puissant objet tableau N-dimensional

— Fonctions sophistiquées (diffusion)
— Des outils pour intégrer le code C / C ++ et Fortran
— Algèbre linéaire utile, transformée de Fourier et capacités de nombres aléatoires.

Outre ses utilisations scientifiques évidentes, NumPy peut également être utilisé comme un conteneur multidimensionnel efficace de données génériques. Des types de données arbitraires peuvent être définis. Cela permet à NumPy de s'intégrer facilement et rapidement à une grande variété de bases de données.

FIGURE 3.6 – Logo Numpy

— **OpenCV :** est un open source Computer vision Library : C'est une bibliothèque logicielle qui a été conçue pour une meilleure efficacité de calcul et avec une forte concentration sur les applications en temps réel. Cette bibliothèque regroupe un large panel de fonctions permettant de faciliter la reconnaissance d'images en mouvement.

FIGURE 3.7 – Logo OpenCV

3.2 Travail réalisé

Dans cette section, nous allons exposer le travail réalisé en détaillant les étapes de création du projet.

3.2.1 Architecture de la solution

Notre application comporte trois principales parties comme le montre la figure 3.8 :

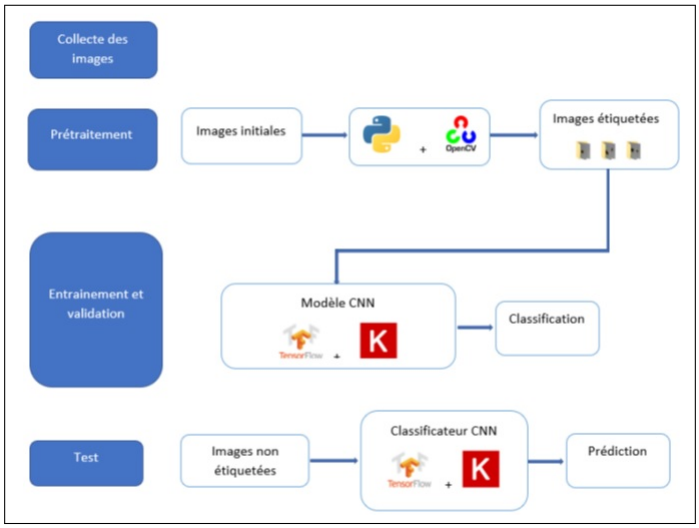

FIGURE 3.8 – Architecture fonctionnelle de classification des images

3.2.2 Recherche de jeu de données «dataset»

Afin de pouvoir distinguer les différentes classes d'amandes, nous allons créer une base de données à partir d'une collecte d'images d'amandes déjà utilisée dans un projet de classification des amandes. Toutes les images sont capturées dans des conditions égales (même appareil photo numérique, même position, même arrière-plan). Notre base se compose d'un total de 1250 images. Le tableau 3.1 montre la répartition du nombre d'images pour chaque catégorie.

Catégorie	**Nombre d'images**
Good-almond	664
Bad-almond	375
Not-almond	211

TABLE 3.1 – Nombre d'images de chaque catégorie

La figure ci-dessous 3.9 montre le diagramme circulaire de cette répartition :

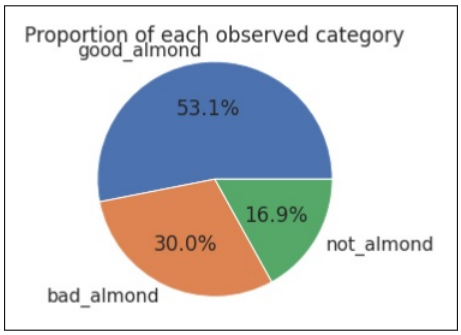

FIGURE 3.9 – Répartition des données

Après avoir trouvé notre base de données, nous avons la divisé sur trois dossiers : un pour la partie d'entraînement, un pour le test et l'autre pour la phase de validation, chaque dossier est décomposé à son tour en trois autres dossiers. Ces trois derniers identifient les trois classes de notre algorithme de Deep Learning.La répartition des données dans ces trois ensembles dépend de la quantité des données à disposition. Cependant, de manière générale, nous réserverons environ 60% pour l'ensemble d'entraînement, 20% pour l'ensemble de validation et 20% pour l'ensemble de tests. La figure 3.10 présente un aperçu sur la structure d'une base de données.

FIGURE 3.10 – Structure de la base de données

3.2.3 Création du modèle de classification

Afin de construire un modèle de classification des images on a suivi une succession d'étapes. L'étape 1 se focalise sur la création du modèle, ce modèle est une suite de blocs de convolution destinée à l'extraction et l'identification des fonctionnalités des images suivies d'un bloc entièrement connecté qui classifie les images à l'aide de poids ajustés pendant l'entraînement. Puis, le modèle génère une prédiction.Les poids sont ajustés pour trouver des modèles afin de faire de meilleures prévisions. Nous utilisons Keras et TensorFlow qui sont la base de construction de la plupart des modèles d'apprentissage en profondeur. Nous avons créé plusieurs architectures de CNN afin de faire une étude comparative.

3.2.4 Les modèles

— **Modèle 01 :** la figure 3.11 montre l'architecture de réseau de neurones :

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu', input_shape = (150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Dropout(rate=0.25),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dropout(rate=0.5),
    tf.keras.layers.Dense(3, activation=tf.nn.softmax)
])
```

FIGURE 3.11 – Architecture du modèle 01

Le type de modèle que nous allons utiliser est séquentiel. La méthode séquentielle est le moyen le plus simple de créer un modèle dans Keras. Il permet de construire un modèle couche par couches. Chaque couche a des poids qui correspondent à la couche qui la suit.Notre réseau est composé de deux couches de convolution (chacune est suivie d'une fonction d'activation ReLU), deux couches de maxpooling et deux couches de fully connected.
L'image en entrée est de taille 150*150, elle passe d'abord par la première couche de convolution (composée de 32 filtres de taille 148*148), ensuite on applique le Maxpooling. À la sortie de cette couche, nous aurons 32 features maps de taille 74*74. Ces 32 features map sont donnés en entrée de la deuxième couche et on répète les mêmes travaux qu'on a faits pour la première convolution pour avoir par la suite 32 features maps de taille 36*36. Ensuite, nous utilisons un réseau de neurones composé de deux couches fully connected. La première a 128 neurones où la fonction d'activation utilisée est le Relue, et la deuxième est un softmax qui permet de calculer la distribution de probabilité des 3 classes.

— **Modèle 02 :** Le deuxième modèle que nous présentons dans la figure 3.12 est composé de six couches de convolution, deux couches de maxpooling et quatre couches de fully connected.

```
model = Models.Sequential()

model.add(Layers.Conv2D(200,kernel_size=(3,3),activation='relu',input_shape=(150,150,3)))
model.add(Layers.Conv2D(180,kernel_size=(3,3),activation='relu'))
model.add(Layers.MaxPool2D(5,5))
model.add(Layers.Conv2D(180,kernel_size=(3,3),activation='relu'))
model.add(Layers.Conv2D(140,kernel_size=(3,3),activation='relu'))
model.add(Layers.Conv2D(100,kernel_size=(3,3),activation='relu'))
model.add(Layers.Conv2D(50,kernel_size=(3,3),activation='relu'))
model.add(Layers.MaxPool2D(5,5))
model.add(Layers.Flatten())
model.add(Layers.Dense(180,activation='relu'))
model.add(Layers.Dense(100,activation='relu'))
model.add(Layers.Dense(50,activation='relu'))
model.add(Layers.Dropout(rate=0.5))
model.add(Layers.Dense(3,activation='softmax'))
```

FIGURE 3.12 – Architecture du modèle 02

— **Modèle 03 :** Le troisième modèle que nous présentons dans la figure 3.13 est composé de quatre couches de convolution, deux couches de maxpooling et deux couches de fully connected.

```
first_Mod = Sequential()

first_Mod.add(Conv2D(64,(3,3),activation='relu',input_shape=input_shape))
first_Mod.add(Conv2D(64,(3,3),activation='relu'))
first_Mod.add(MaxPool2D(pool_size=(2,2)))
first_Mod.add(Dropout(0.5))

first_Mod.add(Conv2D(128,(3,3),activation='relu'))
first_Mod.add(Conv2D(128,(3,3),activation='relu'))
first_Mod.add(MaxPool2D(pool_size=(2,2)))
first_Mod.add(Dropout(0.5))

first_Mod.add(Flatten())
first_Mod.add(Dense(128,activation='relu'))
first_Mod.add(Dropout(0.5))
first_Mod.add(Dense(num_classes,activation='softmax'))
```

FIGURE 3.13 – Architecture du modèle 03

— **Modèle 04 :** Le quatrième modèle que nous présentons dans la figure 3.14 est composé de deux couches de convolution, deux couches de maxpooling et deux couches de fully connected.

```
from keras import models, layers
model = models.Sequential()
model.add(layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', input_shape=X_train.shape[1:]))
model.add(layers.MaxPool2D(pool_size=(2, 2)))
model.add(layers.Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(layers.MaxPool2D(pool_size=(2, 2)))
model.add(layers.Dropout(rate=0.25))
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dropout(rate=0.5))
model.add(layers.Dense(3, activation='softmax'))
```

FIGURE 3.14 – Architecture du modèle 04

3.2.4.1 Paramètres du modèle

Afin de créer le modèle et implémenter la base de données, on initialise les différentes variables pertinentes dans l'algorithme :

- **Nombre d'époques :** est le nombre de fois où le modèle parcourt les données. Plus on a d'époques, plus le modèle s'améliorera, jusqu'à un certain point. Ensuite, le modèle cessera de s'améliorer. En outre, plus le nombre d'époques est important, plus le modèle sera long à exécuter.
- **Fonction de perte "loss" :** On a utilisé "sparce-categhorical-crossentropy" pour notre fonction de perte. C'est le choix le plus courant pour la classification en catégories. Plus la valeur de loss est faible plus le modèle fonctionné mieux.
- **Metrics :** Pour rendre les choses encore plus faciles à interpréter, on a utilisé la mesure de précision «accuracy» pour voir le score de précision du jeu de validation défini à la fin de chaque époque.
- **Optimiseur :** L'optimiseur contrôle le taux d'apprentissage. On a utilisé "Adam", un optimiseur de taux d'apprentissage adaptatif. Il ajuste le taux d'apprentissage tout au long de la formation. Le taux d'apprentissage détermine à quelle vitesse les poids optimaux du modèle sont calculés.Un taux d'apprentissage plus faible peut conduire à des poids plus précis mais le temps nécessaire pour calculer les poids sera plus long.
- **Les valeurs de Poids et des biais :** sont initialisées d'une façon aléatoire.

3.2.4.2 Compilation et formation de modèle

La formation du modèle est faite en passant les données d'entraînement, de validation et de test au réseau construit. Et la compilation du modèle prend les deux paramètres : optimiseur et perte.

3.2.5 Prédiction des nouvelles données

Pour faire des prédictions pour de nouvelles données avec le modèle, on utilise la fonction "predict". Aussi on a créé une fonction qui permet de prédire un ensemble de jeux de donnés.

3.2.6 Evaluation du modèle

Les deux pires problèmes qui pourraient arriver à un modèle d'apprentissage automatique sont soit de construire des connaissances inutiles, soit de ne rien apprendre de pertinent d'un jeu de données de formation. Dans la théorie de l'apprentissage automatique, ces deux phénomènes sont décrits en utilisant respectivement les termes surajustement et sous ajustement et ils constituent deux plus grands défis des solutions modernes d'apprentissage en profondeur.

3.2.6.1 Les métriques d'évaluation de performance de modèle

Il existe un certain nombre de mesures de performance pour les problèmes de classification. Il est de la plus haute importance de réaliser que le choix de la mesure de performance est profondément spécifique à un domaine et à une question. Dans le cas d'un jeu de données avec des classes équilibrées ; comme dans notre cas ; la précision de notre classificateur dépendra de l'efficacité de l'algorithme choisi, de la façon dont nous l'avons appliqué et de

la quantité de données de formation utilisées. Après la formation du modèle, l'évaluation de performance se fait via les métriques telles que :

- **Accuracy :** est le nombre de prédictions correctes effectué par le modèle sur le nombre total d'enregistrements. La meilleure précision est de 100%, indiquant que toutes les prédictions sont correctes.
- **Precision :** possibilité d'un modèle de classification de ne renvoyer que des instances pertinentes.
- **Recall :** capacité d'un modèle de classification a identifié toutes les instances pertinentes.
- **F1-score :** mesure simple combinant rappel et précision en utilisant la moyenne harmonique.
- **Support :** est le nombre d'occurrences de chaque classe.
- **Matrice de confusion « Confusion matrix » :** est l'un des meilleurs moyens d'évaluer les performances du modèle. Elle permet de visualiser les prédictions sous la forme d'une matrice. La matrice de confusion est composée par :
  - **Vrai positifs « TP »** C'est le résultat du modèle qui prédit correctement la catégorie positive.
  - **Faux positifs « FP » :** C'est le résultat du modèle prédit incorrectement la catégorie positive.
  - **Vrai négatifs « TN » :** C'est le résultat du modèle qui prédit correctement la catégorie négative.
  - **Faux négatifs « FN » :** C'est le résultat du modèle prédit incorrectement la catégorie négative.

3.2.6.2 Résultats obtenus

- **Modèle 01 :** Après l'analyse des résultats obtenus, On constate les remarques suivantes : d'après la figure 3.15, la précision de l'apprentissage et de test augmente avec le nombre d'époques, ceci reflète qu'à chaque époque le modèle apprend plus d'informations. Si la précision diminue alors on aura besoin de plus d'informations pour faire apprendre notre modèle et par conséquent on doit augmenter le nombre d'époques et vice-versa. De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'époques.

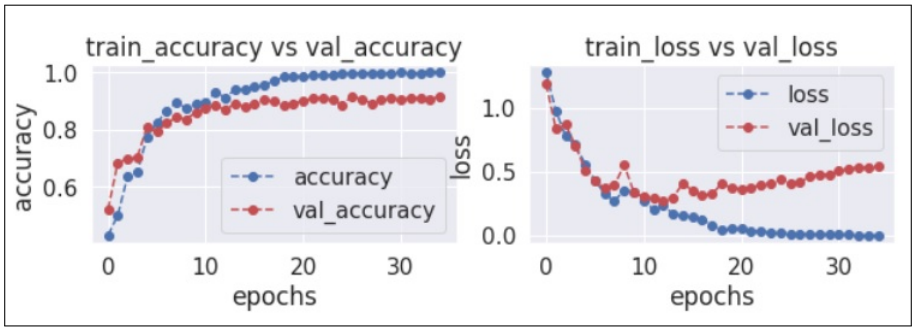

FIGURE 3.15 – Résultat du modèle 01

La matrice de confusion permet d'évaluer la performance de notre modèle, puisqu'elle reflète les métriques du Vrai positif, Vrai négatif, Faux positif et Faux négatif. La

figure 3.16 illustre de près la position de ces métriques pour chaque classe. Pour cet exemple le modèle a bien classé les images.

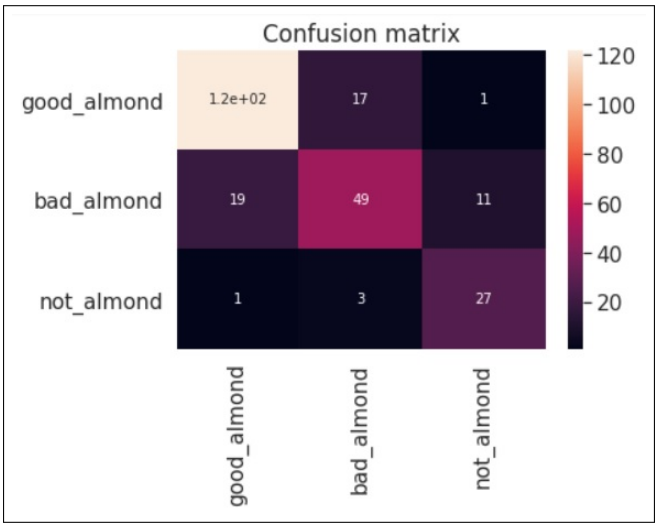

FIGURE 3.16 – Matrice de confusion du modèle 01

— **Modèle 02 :** La figure 3.17 montre le résultat de précision et de perte de modèle 02.

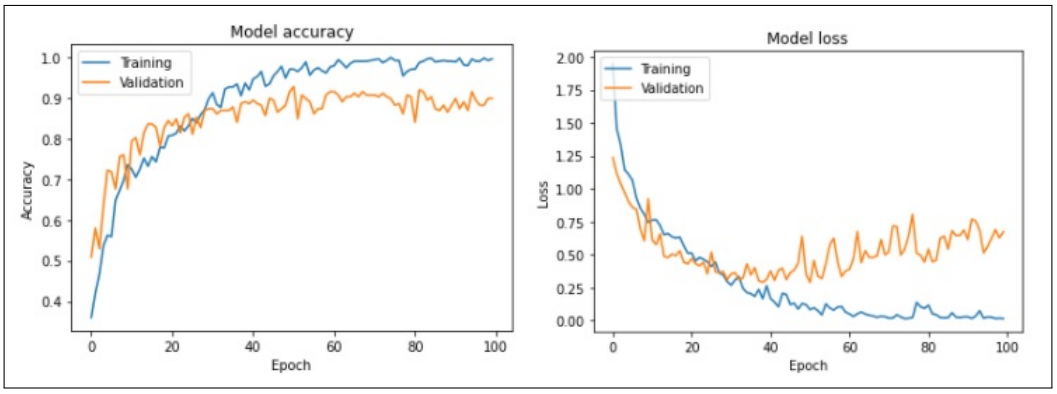

FIGURE 3.17 – Résultat du modèle 02

La figure 3.18 montre la matrice de confusion de modèle 02.

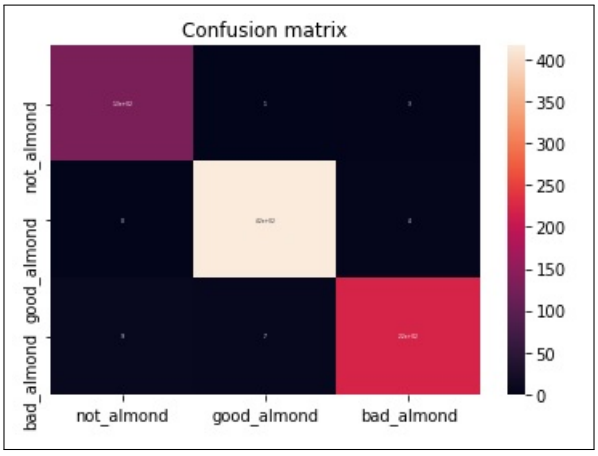

FIGURE 3.18 – Matrice de confusion du modèle 02

— **Modèle 03 :** La figure 3.19 montre le résultat de précision et de perte de modèle 03.

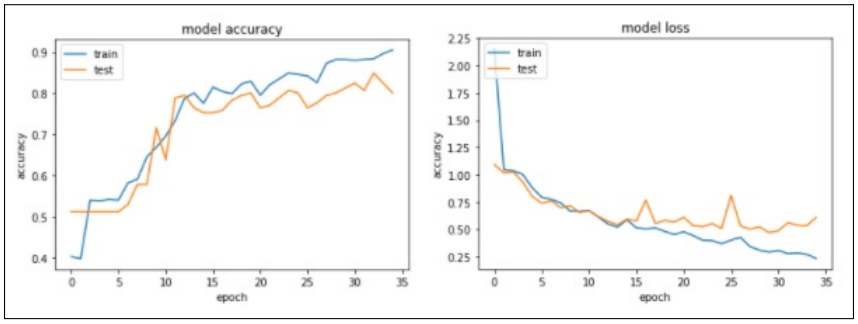

FIGURE 3.19 – Résultat du modèle 03

— **Modèle 04 :** La figure 3.20 montre le résultat de précision et de perte de modèle 04.

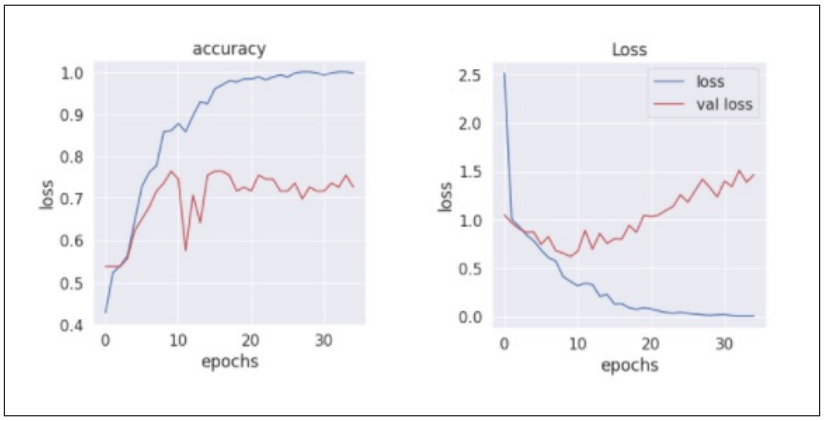

FIGURE 3.20 – Résultat du modèle 04

La figure 3.21 montre la matrice de confusion de modèle 04.

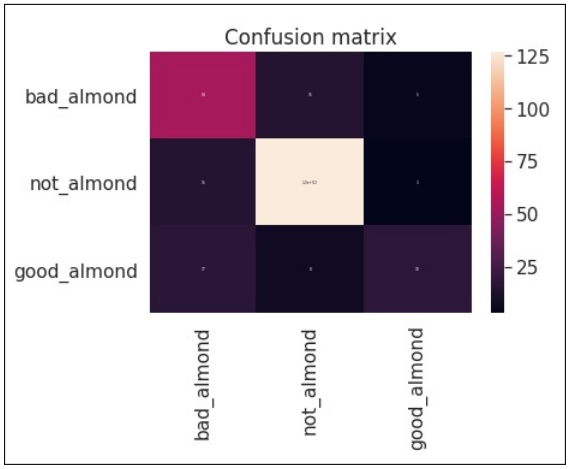

FIGURE 3.21 – Matrice de confusion du modèle 04

3.2.7 Étude comparative entre les quatre modèles réalisés

Le choix de modèle à utiliser se fait selon une étude comparative des modèles en variant les réseaux de neurones. Nous avons créé quatre modèles CNN dont en variant chaque fois

l'architecture.
Le tableau montre les différents résultats des précisions des modèles.

Modèle	**Précision d'entraînement**	**Précision de validation**	**Précision de test**
Modèle 01	100%	91.8%	79.2%
Modèle 02	99.6%	89.9%	76.8%
Modèle 03	98.6%	80.1%	63.6%
Modèle 04	99.7%	72.6%	75.9%

TABLE 3.2 – Comparaison des précisions des modèles

Le modèle candidat est choisi selon la valeur de précision de test la plus proche de 100La figure 3.22 montre un exemple de prédiction d'une image appartenant à la classe "good-almond" avec le modèle choisi.

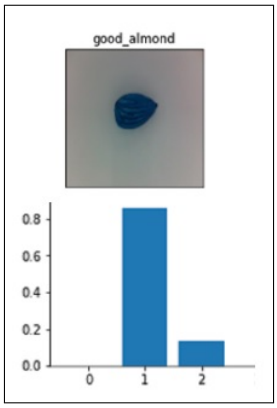

FIGURE 3.22 – Prédiction avec une image de la classe good-almond

La figure 3.23 montre un exemple de prédiction d'une image appartenant à la classe "bad-almond" avec le modèle choisi.

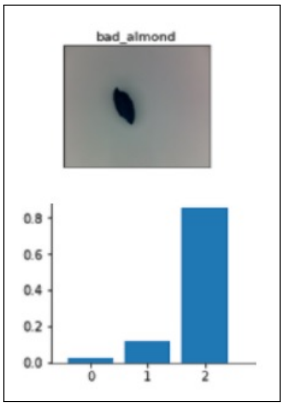

FIGURE 3.23 – Prédiction avec une image de la classe bad-almond

Conclusion

Ce dernier chapitre a présenté la conception de ce projet et les différentes étapes de sa réalisation en clôturant avec une étude comparative des résultats.

Conclusion générale

Le deep learning est une technique moderne qui apparaît à partir des années 2000. Il est le résultat de plusieurs années de recherches et de progrès jusqu'à devenir une nécessité principale dans la plupart des domaines.

Dans ce contexte se déroule le sujet de notre projet de stage ouvrier au sein de « Service Digital » qui consiste à la mise en œuvre d'une solution en faveur des industries spécialisées dans le décorticage et le tri des amandes. La solution avait pour objectif de permettre avec facilité la tâche de tri afin d'améliorer aussi bien la quantité et la qualité du produit final et optimiser le temps de production.

Ce rapport a permis d'englober les tâches effectuées durant ces deux mois. En effet, il y avait une présentation générale du cadre de projet par la présentation de l'organisme d'accueil, son client potentiel et le système utilisé dans le travail. Dans le deuxième chapitre, une étude préliminaire des notions utilisées qui sont le deep learning, les réseaux de neurones et le réseau de neurones convolutif en particulier.

Le dernier chapitre incarne les différents détails de la conception et la réalisation du projet commençant par les exigences logicielles, puis l'architecture de la solution, la recherche des données et enfin présentation des modèles trouvés avec une analyse et évaluation.

Ce stage était une vocation pour moi. J'ai réussi pendant ces deux mois d'avoir une nouvelle vision sur le monde de travail et savoir comment gérer les problèmes et le temps en cas d'être pressé par une livrable. Aussi, j'ai acquis autant de compétences managériales et techniques.

En termes de perspectives, le projet dans sa version actuelle n'est pas encore achevé. Les prochaines itérations apporteront à ce sujet en plus de l'évolution une amélioration de la précision des modèles. Aussi nous envisageons d'implémenter la solution dans l'industrie.