

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Mention : Génie Biomédical

Spécialité : Instrumentation Biomédicale

présenté par :

Medjber Baya

&

Djellali Yousra

---

# *Développement d'une Application d'Évaluation d'Ulcère du Pied Diabétique*

---

Proposé par : Dr. A. Chentir

Année Universitaire : 2020/2021

## ***Remerciements***

*Nous tenons à remercier **ALLAH** le tout puissant de nous avoir donné le courage, la patience et la volonté pour réaliser ce travail.*

*Nous voudrions dans un premier temps remercier, notre directrice de mémoire **Dr. A. Chentir**, pour sa patience, sa disponibilité et surtout ses judicieux conseils et pour avoir contribué à alimenter notre réflexion.*

*Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'il ont porté à notre mémoire en acceptant d'examiner notre travail.*

*Nos familles, pour leurs soutiens constants et leurs encouragements.*

## ***Dédicaces***

*Je voudrais dédier ce modeste travail à :*

*Mes parents, merci pour votre soutien, vos prières, votre bienveillance et votre présence qui sont la source de ma force pour affronter les obstacles.*

*Mes chères sœurs **Manel** et **Fatma Zohra** et mon frère **Mounir** qui n'ont pas cessé de me conseiller, encourager et me soutenir tout au long de mes études.*

*Tous les mots restent faibles pour leur exprimer mes sentiments et qu'ils acceptent seulement ces lignes en guise de témoignage.*

***Baya.***

## ***Dédicaces***

*Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers.*

### ***A ma chère mère***

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que tu as consenti pour mon instruction et mon bien être. Je te remercie pour tout le soutien et l'amour que tu me porte. Que ce modeste travail soit l'exaucement de tes vœux, et le fruit de tes innombrables sacrifices.*

### ***A la mémoire de mon cher père***

*Ce travail est dédié à mon père, décédé trop tôt, qui m'a toujours poussé et motivé dans mes études. J'espère que du monde qui est sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance.*

*Puisse dieu le tout puissant, l'avoir en son vaste paradis.*

### ***A mon frère et mes amies***

*A mon frère **Oussama** et mes meilleures amies **Zouaoui Chahla** et **Alim Yousra** , je vous remercie pour votre présence et votre soutien.*

***Yousra.***

**ملخص:** تم تطوير عدة طرق وخوارزميات للتعرف على قرحة القدم السكرية وتصنيفها. من بين كل هذه التقنيات، اخترنا نهج الشبكات العصبية الالتفافية من خلال تطبيق التعليم الاشتراكي على 3 قواعد بيانات. استخدمنا العديد من النماذج المدربة مسبقاً (VGG16 و VGG19 و InceptionResNet V2). بعد اختبارهم جميعاً وجدنا أن VGG19 هو الأفضل أداءً في تصنيف قرحة القدم السكرية وفقاً لـ (UT) وبدقة 99.47% في الاختبار. التطبيق الذي نقوم بإنشائه يجعل تقييم DFU وفقاً لنوع وخطر البتر ممكناً.

**الكلمات المفتاحية:** قرحة القدم السكرية، الشبكات العصبية الالتفافية

---

**Résumé :** Plusieurs méthodes et algorithmes ont été développés pour la reconnaissance et la classification de l'ulcère du pied diabétique (UPD) chez les patients malades. Parmi toutes ces techniques, nous avons choisis l'approche CNN en appliquant l'apprentissage par transfert sur 3 bases de données. Nous avons utilisé plusieurs modèles pré entraînés (VGG16, VGG19 et InceptionResNet V2). Après avoir testé ces derniers, nous avons constaté que le VGG19 est le plus performant en classification d'ulcère du pied diabétique selon (UT) avec une précision de 99,47% en test. L'application réalisée permet d'évaluer l'UPD à partir de son type plus le risque d'amputation.

**Mots clés :** CNN, UPD, VGG16, VGG19, InceptionResNet V2, Apprentissage par Transfert.

---

**Abstract:** Several methods and algorithms have been developed for the recognition and classification of DFU in diabetic patients. Among all these techniques, we have chosen the CNN approach by applying transfer learning on 3 databases. We used several pre-trained models (VGG16, VGG19 and InceptionResNet V2). After testing all of them, we found that VGG19 is the best performer in the classification of diabetic foot ulcer according to (UT) with an accuracy of 99.47% in the test. The application that we create makes the evaluation of DFU according to the type and the amputation risk possible.

**Keywords :** CNN, DFU, VGG16, VGG19, InceptionResNet V2, Transfer Learning.

---

## Table des Matières

Remerciements	
Dédicaces	
Résumés	
Table des Matières.....	i
Liste des Figures.....	iv
Liste des Tableaux.....	vi
Liste des abréviations.....	vii
Introduction Générale.....	1
<b>I Chapitre I : Contexte Médical .....</b>	
I.1 Introduction .....	3
I.2 Le diabète .....	3
I.2.1 Qu'est-ce que le diabète ?.....	4
I.2.2 Le Diabète de Type 1 .....	4
I.2.3 Le Diabète de Type 2 .....	5
I.2.4 Les complications du diabète .....	7
I.3 Le Pied diabétique .....	8
I.3.1 Rappel anatomique du pied .....	8
I.3.2 Physiopathologie .....	12
I.3.3 L'ulcère du Pied Diabétique.....	17
I.4 Conclusion .....	22
<b>II Chapitre II : Etat de l'Art .....</b>	
II.1 Introduction .....	23
II.2 Méthodes de Détection et de Classification .....	23
II.3 Conclusion .....	32
<b>III Chapitre III : L'algorithme CNN .....</b>	

# Table des Matières

---

III.1	Introduction .....	33
III.2	Le Deep Learning .....	33
III.3	Réseaux de Neurones.....	33
III.4	Les Réseaux de Neurones Convolutifs.....	34
III.4.1	Architecture de Réseaux de Neurone Convolutifs.....	35
III.4.2	Les Différentes Couches de CNN.....	36
III.5	Apprentissage par transfert (Transfer learning).....	40
III.6	Paramètres d'évaluation du modèle CNN .....	45
III.7	Conclusion .....	47
<b>IV</b>	<b>Chapitre IV : Résultats d'implémentation .....</b>	
IV.1	Introduction .....	48
IV.2	Base de Données utilisée .....	48
IV.3	Environnement et Bibliothèques .....	49
IV.3.1	Google Colaboratory .....	50
IV.3.2	Python.....	50
IV.3.3	TensorFlow .....	51
IV.3.4	Keras.....	51
IV.3.5	Gradio .....	51
IV.4	Développement de l'Application.....	52
IV.4.1	Le premier Système de Classification d'UPD.....	53
IV.4.2	Le Deuxième Système de Classification d'UPD .....	65
IV.4.3	Création d'une interface utilisateur .....	73
IV.5	Conclusion.....	73
<b>V</b>	<b>Conclusion Générale.....</b>	<b>74</b>
<b>VI</b>	<b>Bibliographie .....</b>	<b>76</b>
<b>Annexes</b>	.....	

## Liste des Figures

<b>Figure I-1</b> : Illustration médicale précise du pancréas .....	03
<b>Figure I-2</b> : Différence entre un sujet non Diabétique et un sujet Diabétique de type 1 .....	05
<b>Figure I-3</b> : Différence entre un sujet non Diabétique et un sujet Diabétique de type 2 .....	06
<b>Figure I-4</b> : Les Complications du Diabète .....	07
<b>Figure I-5</b> : Illustration d’Os du pied vue dorsale ... ..	08
<b>Figure I-6</b> : Illustration d’Os du pied vue latérale ... ..	09
<b>Figure I-7</b> : Vues plantaires des muscles d’un pied droit. ....	09
<b>Figure I-8</b> : Vues latérale et médiale des muscles d’un pied droit.....	10
<b>Figure I-9</b> : Vue dorsale des muscles courts extenseurs des orteils et de l’hallux.....	10
<b>Figure I-10</b> : Nerfs tibial et fibulaire .....	11
<b>Figure I-11</b> : Artères plantaires .....	12
<b>Figure I-12</b> : Pied de Charcot .....	13
<b>Figure I-13</b> : Mal Perforant Plantaire Neuropathique .....	13
<b>Figure I-14</b> : 2ème Orteil en Griffes chez un Patient Diabétique .....	14
<b>Figure I-15</b> : ulcérations ischémiques .....	14
<b>Figure I-16</b> : gangrène sèche du gros orteil .....	14
<b>Figure I-17</b> : Nécrose avec Dermo-Hypodermite sur Pied Diabétique .....	15
<b>Figure I-18</b> : Fasciite Nécrosante de la Jambe et de la Cheville chez un patient Diabétique..	15
<b>Figure I-19</b> : gangrène extensive du gros orteil .....	16
<b>Figure I-20</b> : Pied Diabétique avec Atteinte Osseuse .....	16
<b>Figure I-21</b> : Physiopathologie du Pied Diabétique .....	17
<b>Figure II-1</b> : Résultat data augmentation.....	28
<b>Figure II-2</b> : Modèle CNN pour la classification d’UPD.....	29
<b>Figure II-3</b> : architecture YOLOv2.....	30
<b>Figure II-4</b> : l’architecture du réseau utilisé par ASURA.....	31
<b>Figure III 1</b> : la relation entre les 3 domaines.....	33
<b>Figure III-2</b> : Schéma représentant l’architecture d’un CNN .....	36
<b>Figure III-3</b> :Illustration des différentes couches de CNN .....	36



<b>Figure III-4</b> : Exemple d'une convolution 2D. ....	37
<b>Figure III-5</b> : Pooling avec un filtre 2x2 et un pas de 2.....	38
<b>Figure III-6</b> : la courbe de la couche ReLU .....	39
<b>Figure III-7</b> : architecture du VGG16 .....	42
<b>Figure III-8</b> : Présentation de l'architecture VGG19 en 3D .....	43
<b>Figure III-9</b> : Illustration de l'architectureVGG19.....	44
<b>Figure III-10</b> : Schéma de principe d'Inception-ResNet-v2 .....	44
<b>Figure III-11</b> : Matrice de confusion.....	45
<b>Figure IV-1</b> : Exemple des images d'ulcère du Pied Diabétique( PartA_DFU).....	48
<b>Figure IV-2</b> : Exemple des images du pied sain (PartA_DFU).....	48
<b>Figure IV-3</b> : Différents types d'ulcère du Pied Diabétique(DFUC2021). ....	49
<b>Figure IV-4</b> : Exemple d'annotation d'images sous fichier CSV (DFUC2021).. ....	49
<b>Figure IV-5</b> : Logo de Colaboratory .....	50
<b>Figure IV-6</b> : Logo Python ....	50
<b>Figure IV-7</b> : Logo de TensorFlow .....	51
<b>Figure IV-8</b> : Logo de Keras .....	51
<b>Figure IV-9</b> : Organigramme Général .....	52
<b>Figure IV-10</b> : Une image de notre base de données. ....	56
<b>Figure IV-11</b> : Configuration du Modèle(1). ....	57
<b>Figure IV-12</b> : Configuration du Modèle(1). ....	58
<b>Figure IV-13</b> : Résultat de la phase d'apprentissage (modèle 1). ....	59
<b>Figure IV-14</b> : L'évaluation du Modèle (1). ....	60
<b>Figure IV-15</b> : Exemple d'1 résultat de prédiction (Modèle 1).....	60
<b>Figure IV-16</b> : Paramètres d'évaluation (Modèle 1). ....	60
<b>Figure IV-17</b> : Matrice de confusion (Modèle 1). ....	61
<b>Figure IV-18</b> : Résultat d'apprentissage (Modèle 2). ....	62
<b>Figure IV-19</b> : L'évaluation du Modèle (Modèle 2).. ....	62
<b>Figure IV-20</b> : Exemple de résultat de prédiction (Modèle 2). ....	63
<b>Figure IV-21</b> : Paramètres d'évaluation (Modèle 2).. ....	63
<b>Figure IV-22</b> : Matrice de confusion (Modèle 2).....	63
<b>Figure IV-23</b> : Classification de Texas .....	65
<b>Figure IV-24</b> : Configuration du Modèle 1 (Système 2). ....	67

## Liste des Figures

---

<b>Figure IV-25</b> : Configuration du Modèle 2 (Système 2). .....	68
<b>Figure IV-26</b> : exemples d'images du classe « None » .....	69
<b>Figure IV-27</b> : exemples d'images du classe « Infection ».....	69
<b>Figure IV-28</b> : Résultat d'apprentissage (Système 2).....	70
<b>Figure IV-29</b> : Résultat de prédiction (Système 2).....	71
<b>Figure IV-30</b> : Matrice de Confusion (Système 2).....	71
<b>Figure IV-31</b> : Paramètres d'évaluation (Système 2).....	71
<b>Figure IV-32</b> : Interface Graphique Développée .....	73

## Liste des Tableaux

<b>Tableau I-1</b> : Symptômes cliniques des ulcères du pied neuropathiques et ischémiques .....	18
<b>Tableau I-2</b> : Système de classification de Wagner-Meggitt.....	19
<b>Tableau I-3</b> : Classification de SINBAD.....	19
<b>Tableau I-4</b> : Système de classification PEDIS.....	20
<b>Tableau I-5</b> : Classification de l'Université du Texas et pourcentages de risques d'amputation associés.....	21
<b>Tableau I-6</b> : Caractéristiques des Systèmes de Classification pour les Ulcères du Pied Diabétique .....	22
<b>Tableau II-1</b> : L'architecture de la Méthode proposée par M. Goyal et al.....	25
<b>Tableau IV-1</b> : Informations sur la Base de Données PartA_DFU.....	48
<b>Tableau IV-2</b> : Informations sur la Base de Données DFUC2021 .....	49
<b>Tableau IV-3</b> : Les Paramètres utilisés pour la Phase d'Apprentissage (Modèle 1).....	57
<b>Tableau IV-4</b> : L'optimisateur SGD.....	57
<b>Tableau IV-5</b> : Les Paramètres utilisées pour le Modèle 2.....	58
<b>Tableau IV-6</b> : L'optimisateur Adam.....	58
<b>Tableau IV-7</b> : Tableau comparatif des 2 Modèles réalisés.....	64
<b>Tableau IV-8</b> : Distribution des images dans Base de Données utilisée (Système 2).....	66
<b>Tableau IV-9</b> : Distribution des images dans Base de Données utilisée (Système 2).....	66
<b>Tableau IV-10</b> : Résultats obtenus des 3 Modèles utilisés .....	69

## Liste des Abréviations

**API:** Application Programming Interface.

**CNN:** Convolutional Neural Network.

**DFU:** Diabetic Foot Ulcer.

**DCNN:** Deep Convolutional Neural Network.

**FCN:** Fully Connected Layer.

**GPU :** Graphics Processing Unit.

**IA :** Intelligence Artificielle.

**ML:** Machine Learning.

**ReLU:** Rectified Linear Unit.

**SIFT:** Scale-Invariant Feature Transform.

**SVM:** Support Vector Machine.

**TPU:** Tensor Processing Unit.

**UT:** University of Texas.

**UPD :** Ulcère du Pied Diabétique.

### Introduction Générale

Le diabète représente une pathologie chronique très importante qui menace la santé des algériens. En effet, l'Algérie a compté plus de 4,5 millions de diabétiques en 2019, ce nombre augmente chaque année.

La mortalité est doublée chez les diabétiques atteints d'un ulcère du pied, le risque d'amputation est multiplié de 10 à 30 fois par rapport à la population non diabétique. Le diabète provoque une multitude de complications dont plusieurs touchent le pied, ils sont la conséquence des mécanismes physiopathologiques (la neuropathie, l'artériopathie et l'infection).

L'altération grave de la santé et de la qualité de vie des individus atteints et l'importance du fardeau sociétal requièrent une approche interprofessionnelle dans la dispensation des soins aux ulcères du pied diabétique afin d'en optimiser la prise en charge, Il est ainsi pertinent de s'intéresser à développer un outil d'aide au diagnostic par l'intelligence artificielle qui va amplifier le potentiel humain.

L'intelligence Artificielle pourrait s'avérer très utile dans le cadre de classifier l'ulcère du pied diabétique et donc de faciliter le diagnostic qui est un point essentiel pour une meilleure prise en charge.

Nous avons tout dans notre entourage au moins une personne touchée par le diabète, notre soutien est précieux pour permettre au patient d'accepter sa maladie et d'adhérer à sa prise en charge pour minimiser ces complications, Le pied diabétique est une des complications majeures de cette maladie. Le développement d'un système d'évaluation des plaies de pieds diabétiques est un thème traité dans la promotion 2019/2020. Cela nous a poussés d'approfondir dans ce thème et d'apporter des améliorations.

L'ulcère du pied diabétique est la principale complication du diabète qui, s'il n'est pas correctement traité, peut entraîner une amputation. L'une des approches du traitement DFU dépend de l'attention des cliniciens et des patients. Cette approche thérapeutique présente des inconvénients tels que le coût élevé du diagnostic ainsi que la durée du traitement. Bien que cette approche donne des résultats puissants, le besoin d'une approche de diagnostic DFU à distance, rentable et pratique est urgent. Dont la problématique qui sera exposée.

L'objectif de notre projet fin d'étude est de trouver un algorithme robuste afin de développer une application qui va nous faire une classification d'ulcère du pied diabétique et de graduer la sévérité de ce dernier pour aider le médecin à évaluer le pronostic et de permettre une prise en charge adaptée à chaque type d'ulcère dans le but d'éviter l'amputation.

La structure de ce mémoire est fondée sur 4 chapitres.

- Le premier chapitre sera consacré à la description anatomique du pied et à la description physiopathologique du diabète, en particulier la complication la plus grave de ce dernier, dans laquelle le problème a été évoqué.
- Concernant le deuxième chapitre, qui s'intitule « État de l'art », nous parlerons des différentes méthodes, architectures et techniques qui ont été proposées pour la détection et la classification de la plaie du pied diabétique.
- Le troisième chapitre sera consacré aux approches et aux méthodes que nous avons utilisées pour la classification en imagerie médicale. Avec des tests et des organigrammes bien détaillés de chaque méthode.
- Le dernier chapitre intitulé « Résultats d'implémentation », sera consacré à la présentation et à l'interprétation des résultats obtenus qui seront regroupés dans une interface graphique finale.
- Une conclusion générale viendra clôturer notre travail.

---

# **Chapitre I**

## **Contexte Médical.**

---

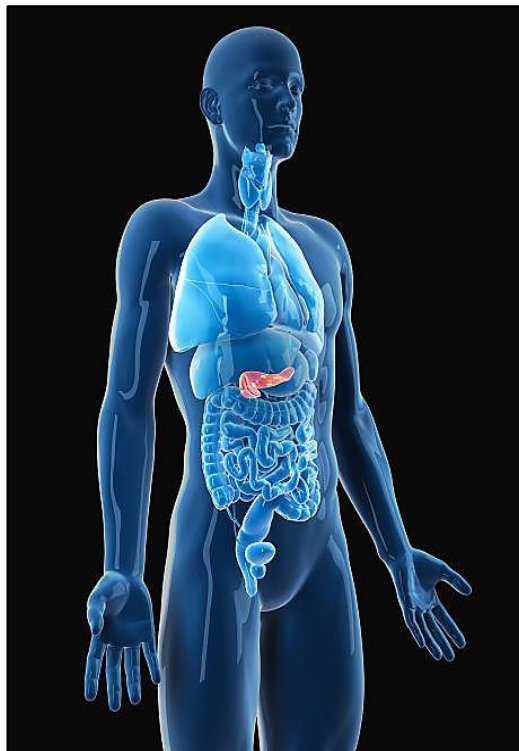
## I.1 Introduction

Les complications du diabète sont nombreuses et peuvent être sévères. Ces complications aggravent le diabète et tendent à faire baisser l'espérance de vie des personnes atteintes. La majorité des complications liées au diabète peuvent être évitées, réduites ou retardées si le diabète est dépisté et traité précocement et correctement [1].

Dans ce chapitre, nous commencerons par une petite description anatomique du pancréas par la suite nous parlerons sur le diabète et ses complications en général, puis sur le pied diabétique en particulier avec ses différents systèmes de classification.

## I.2 Le diabète

Le pancréas est situé dans l'abdomen derrière l'estomac (Figure I-1). Il secrète des enzymes très puissants qui se déchargent dans le tube digestif, mais aussi une hormone : l'insuline qui va dans le sang [2].



**Figure I-1 :** Illustration médicale précise du pancréas [3].



## I.2.1 Qu'est-ce que le diabète ?

Le diabète est une maladie chronique qui survient lorsque le pancréas ne produit pas suffisamment d'insuline ou lorsque l'organisme n'est pas capable d'utiliser efficacement l'insuline qu'il produit. L'insuline est une hormone régulatrice de la glycémie [4].

Le diabète est défini par une hyperglycémie chronique c'est-à-dire un excès de sucre dans le sang. Il traduit un trouble de l'assimilation, de l'utilisation et du stockage des sucres [5].

Il s'agit d'une pathologie chronique : le traitement permet de la contrôler mais pas de la guérir [5].

On distingue principalement deux types de diabète : le diabète de type 1 (moins fréquent) et le diabète de type 2 (plus de 90 % des cas) [5].

## I.2.2 Le Diabète de Type 1

Le diabète de type 1, appelé autrefois diabète insulino-dépendant (DID), est habituellement découvert chez les personnes jeunes : enfants, adolescents ou jeunes adultes [6].

### a. La cause

Il s'agit d'une pathologie auto-immune : le corps fabrique lui-même des anticorps qui vont détruire les cellules productrices d'insuline du pancréas. Lorsqu'un certain nombre de ces cellules ont été détruites, le corps manque d'insuline et le diabète apparaît [6].

### b. Les signes

Le manque d'insuline et l'hyperglycémie se traduisent par un amaigrissement, malgré un appétit conservé, une soif importante et des urines abondantes [6].

### c. Le traitement

Il s'agit de remplacer l'insuline que le pancréas ne peut plus produire, il s'agira donc d'injections d'insuline. Cette dernière peut être injectée avec des stylos à insuline (schéma à plusieurs injections par jour) ou avec un petit appareil (qui est fixé au niveau cutané par un petit cathéter que le patient ne change que tous les deux à trois jours et qui permet de délivrer de l'insuline en temps réel) que l'on appelle une pompe à insuline [6].

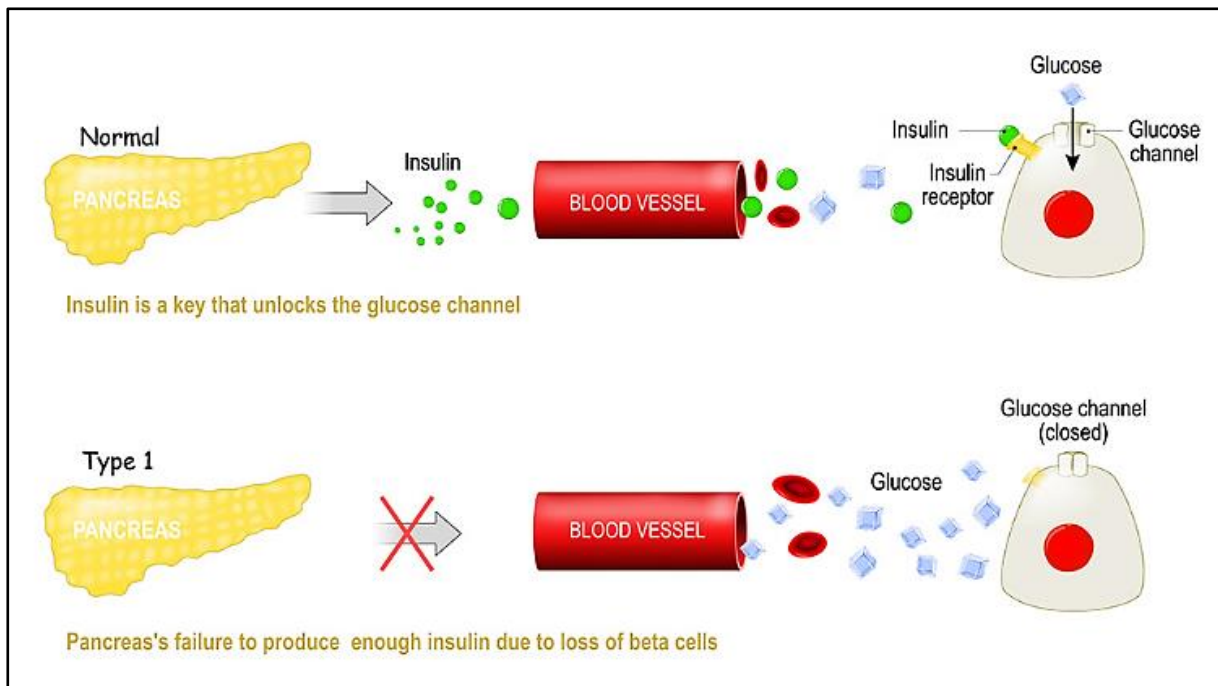


Figure I-2 : Différence entre un sujet non Diabétique et un sujet Diabétique de type 1 [7].

Dans la figure ci-dessus, le pancréas ne produit pas suffisamment d'insuline en raison de la perte de cellules bêta et vu que l'insuline est la clé qui déverrouille le canal du glucose, la cellule, à son tour, n'absorbera pas le glucose.

### I.2.3 Le Diabète de Type 2

Le Diabète de Type 2 apparaît généralement chez des personnes d'âge mur. Il est souvent associé au surpoids et l'obésité qui favorisent son apparition. Il existe en général des antécédents de Diabète dans la famille [8].

#### a. La cause

Il n'existe pas de cause précise mais on note deux anomalies qui sont responsables du Diabète. L'insulinorésistance, en partie génétique et en grande partie liée au surpoids (alimentation déséquilibrée et manque d'activité physique), traduit le fait que l'insuline produite par le pancréas n'arrive plus à être complètement efficace : le pancréas doit donc produire plus d'insuline pour faire le même travail. L'insulinopénie traduit l'épuisement du pancréas qui n'arrive plus à produire suffisamment d'insuline pour assumer les besoins de l'organisme [8].

## b. Les signes

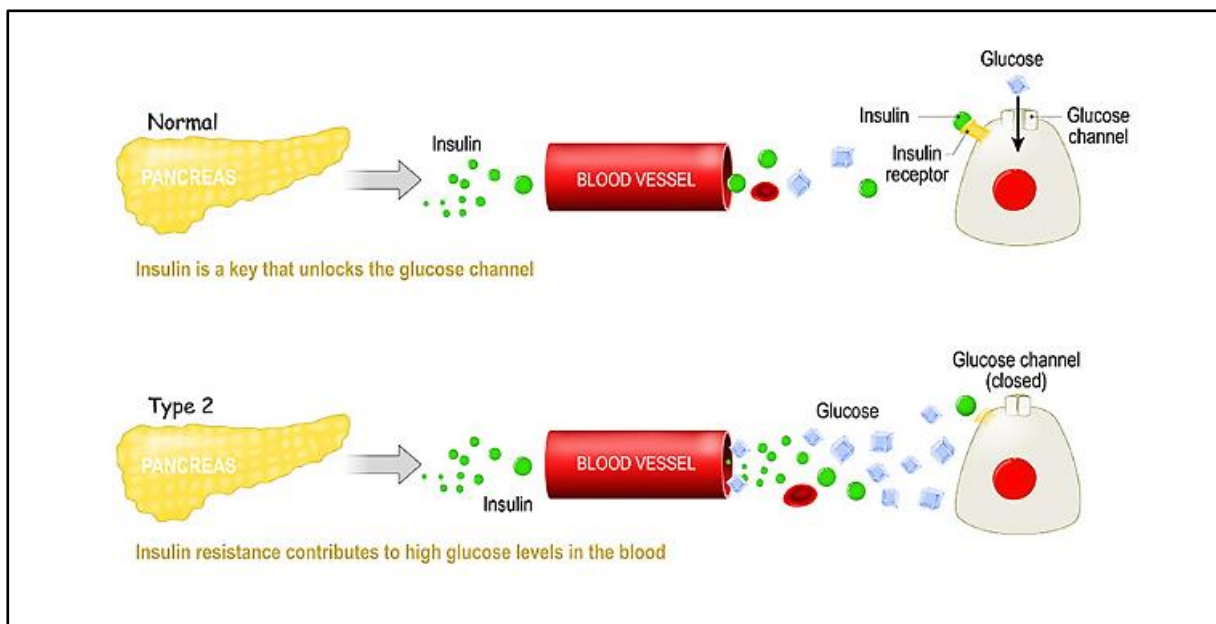
Ils sont souvent absents au début, le développement du diabète de type 2 peut donc passer longtemps inaperçu : c'est pourquoi, il est important de réaliser un dépistage régulièrement en cas de facteurs de risque (antécédents familiaux, diabète gestationnel, obésité...) [8].

## c. Le traitement

La perte de poids est primordiale, elle passe par l'augmentation de l'activité physique ainsi qu'une alimentation équilibrée. Si celle-ci est insuffisante on peut mettre en place des médicaments sous forme de comprimé ou d'injection selon les besoins. En cas de grand déséquilibre ou après avec l'évolution de la maladie un traitement par insuline peut devenir nécessaire [8].

Le diabète de type 2 est évitable par des mesures de prévention, telles que l'hygiène alimentaire et l'activité physique afin d'éviter la prise de poids en particulier quand on a des antécédents de diabète dans la famille [8].

Dans la Figure I.3, on montre que la résistance à l'insuline contribue à une glycémie élevée. L'insuline étant la clé qui déverrouille le canal du glucose.



**Figure I-3 :** Différence entre un sujet non Diabétique et un sujet Diabétique de type 2 [9].

## I.2.4 Les complications du diabète

### ➤ Pourquoi traiter le Diabète alors qu'il donne souvent peu de symptômes ?

Parce que l'hyperglycémie chronique endommage les petites et les grosses artères de l'organisme avec des conséquences sur de nombreux organes [10] comme illustrées dans la Figure I-4 :

- **L'œil** : c'est la rétinopathie diabétique. Le risque est la cécité. On surveille le fond d'œil au moins une fois par an.
- **Le rein** : c'est la néphropathie diabétique. Le risque est de développer une insuffisance rénale. On surveille la présence de protéine dans les urines au moins une fois par an, mais le plus souvent tous les six mois.
- **Les nerfs** : c'est la neuropathie diabétique. Le risque est de perdre la sensibilité surtout au niveau des pieds. On examine les pieds au moins une fois par an.
- **Le cœur** : le risque est de faire un infarctus du myocarde. On réalise un suivi régulier avec le cardiologue.
- **Le cerveau** : le risque est de faire un Accident Vasculaire Cérébral. On examine les vaisseaux du cou au moins une fois par an.
- **Les artères des jambes** : le risque est de développer de l'artérite. On surveille les pouls tous les deux ans, voire tous les ans si on a d'autres facteurs de risque cardio-vasculaires.
- **Les pieds** : l'association de neuropathie et artérite fragilise les pieds qui sont à risque de développer des plaies difficiles à cicatrifier. On surveille les pieds au moins une fois par an. En cas de plaie, on consulte son diabétologue très rapidement.

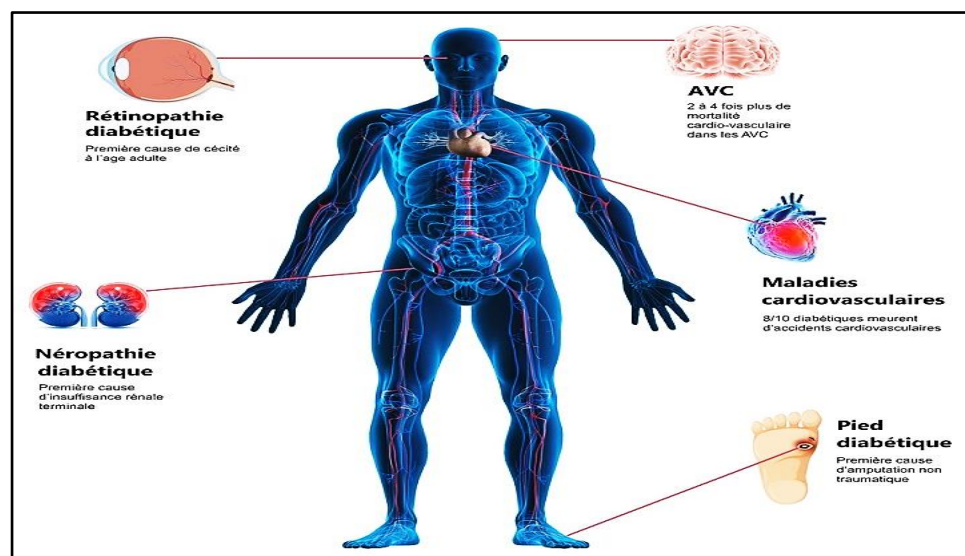


Figure I-4 : Les Complications du Diabète [11].

### I.3 Le Pied diabétique

Le pied diabétique regroupe toute infection, ulcération ou destruction des tissus profonds du pied associées à une neuropathie et/ou une artériopathie périphérique des membres inférieurs chez le diabétique. C'est une complication fréquente et grave du diabète avec un taux d'amputation de membres inférieurs très élevé et des conséquences souvent dramatiques sur le plan socio-économique et psychologique [12].

L'ulcère du pied diabétique (UPD) est l'une des complications les plus pénibles et il est défini comme le point final majeur le plus courant de la maladie du pied diabétique [13].

#### I.3.1 Rappel anatomique du pied

Le pied est la partie la plus distale du membre inférieur qui s'articule avec les os de la jambe par la cheville [14]. Il comprend en général :

- 28 os.
- 16 articulations.
- 107 ligaments qui tiennent ces dernières.
- 20 muscles intrinsèques.

##### a. Ostéologie :

Il existe trois groupes d'os au niveau du pied (Figures I-5 et I-6) [14] :

- les sept os du tarse : calcanéus, talus, cuboïde, os naviculaire et les trois cunéiformes
- les cinq métatarsiens.
- les phalanges.

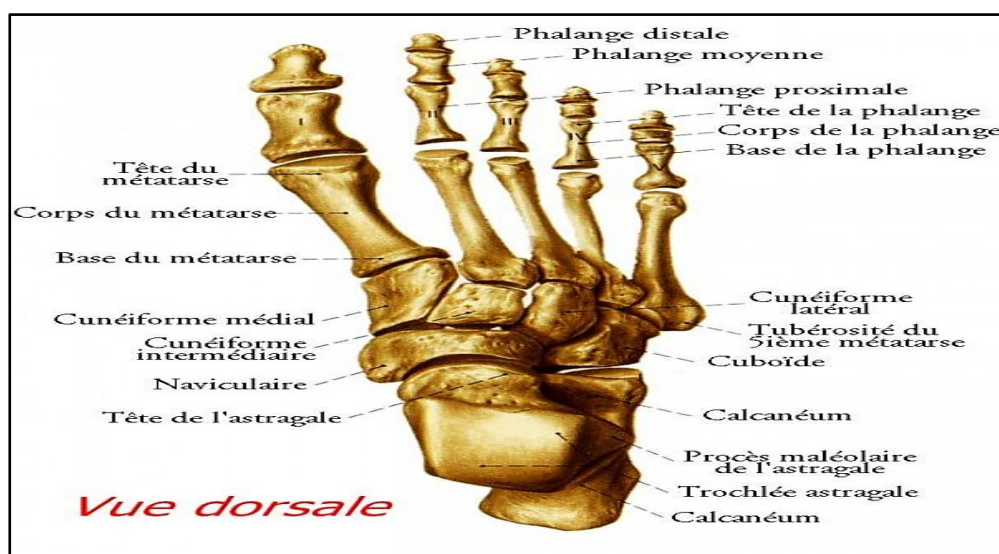


Figure I-5 : Illustration d'Os du pied- vue dorsale [15].



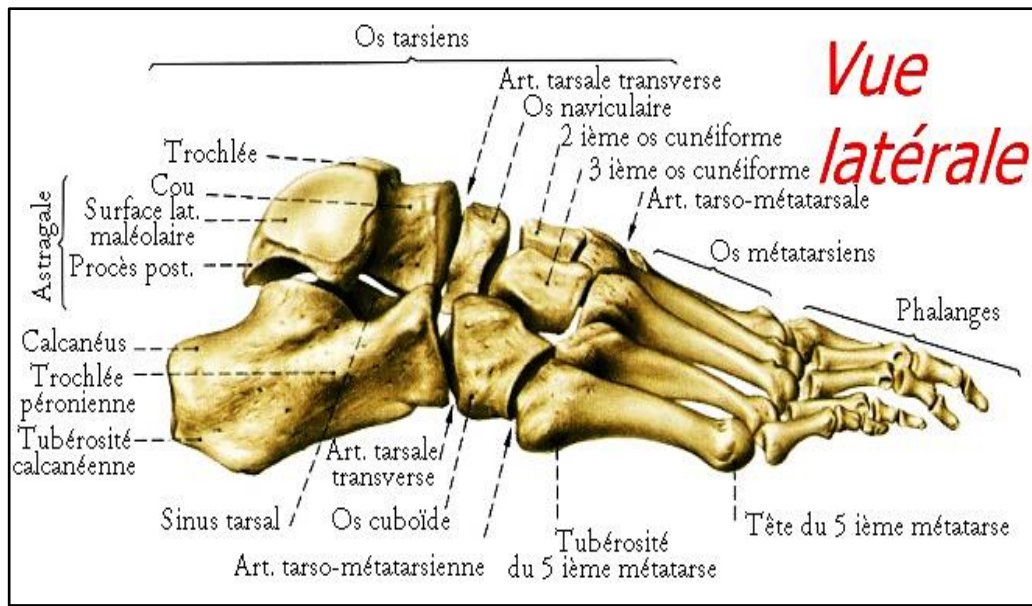


Figure I-6 : Illustration d’Os du pied - vue latérale [15].

**b. Muscles du pied**

Les muscles intrinsèques du pied sont répartis en plusieurs loges : les loges plantaires médiales, centrale et latérale, la loge interosseuse et la loge dorsale [14]. Les Figures I-7, I-8 et I-9 montrent plus de détails sur les muscles intrinsèques du pied.

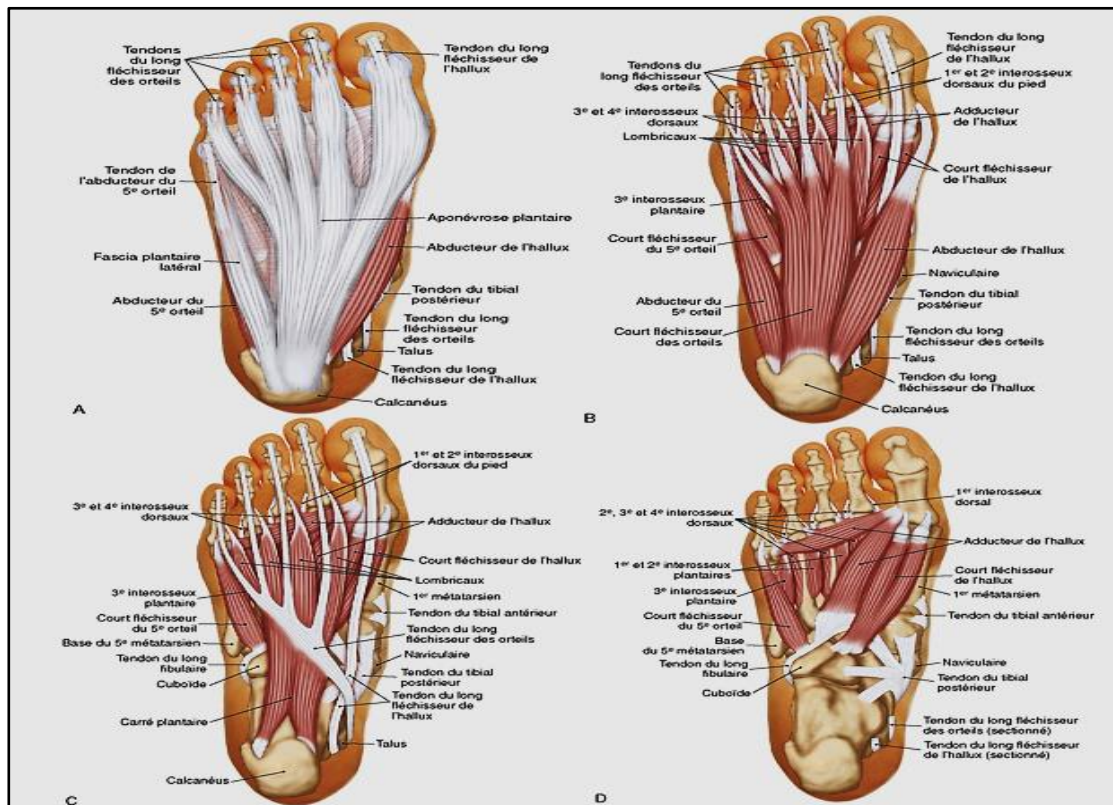


Figure I-7 : Vues plantaires des muscles d’un pied droit. A. Vue superficielle, montrant l’aponévrose plantaire. B. Vue superficielle montrant les muscles. C. Couche intermédiaire. D. Couche profonde [16].

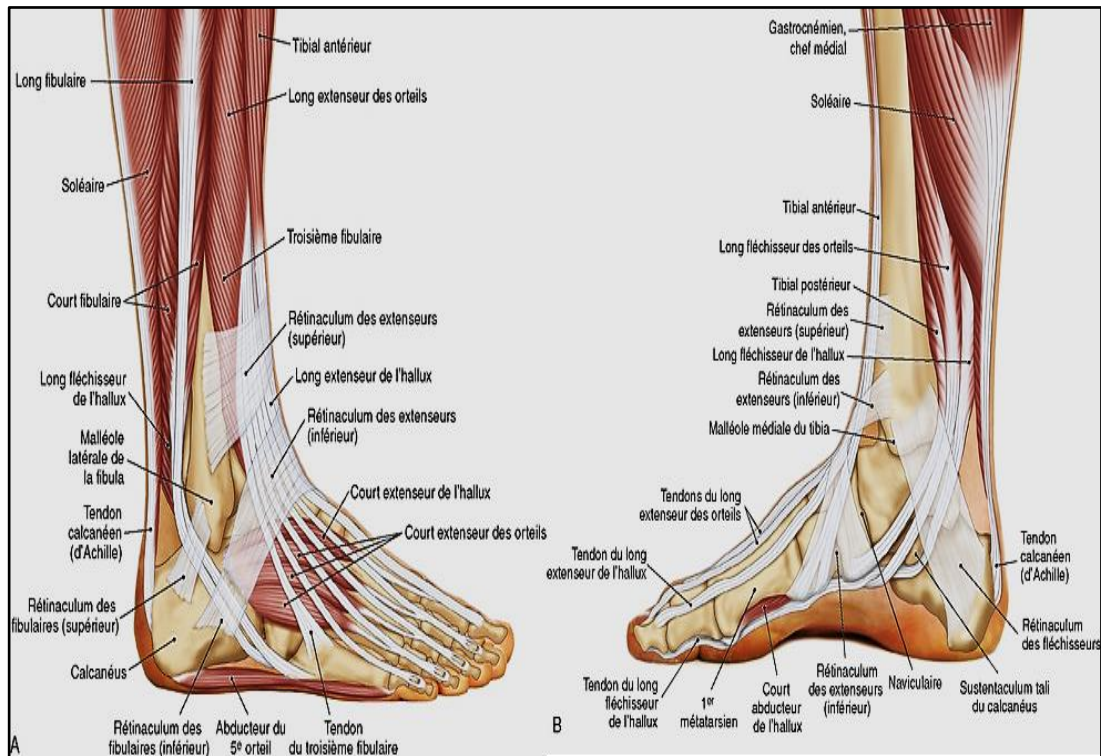


Figure I-8 : Vues latérale et médiale des muscles d'un pied droit. A. Vue latérale. B. Vue médiale [16].

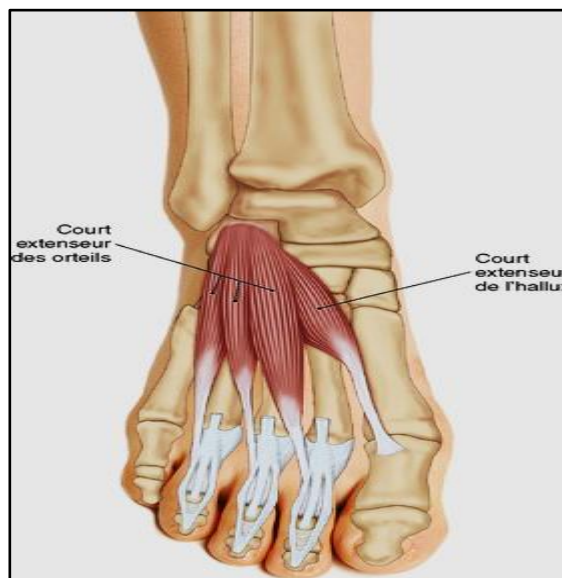


Figure I-9 : Vue dorsale des muscles courts extenseurs des orteils et de l'hallux [16].

**c. Nerfs du pied :**

Les nerfs du pied sont essentiellement issus du nerf sciatique dans la cuisse (Figure I.10). Ce sont des branches du nerf tibial d'une part, et d'autre part des nerfs fibulaires superficiels et fibulaires profonds, issus du nerf fibulaire commun dans la jambe. Une petite



partie de l'innervation est assurée par des branches du nerf saphène, issu du nerf fémoral dans la cuisse [17].

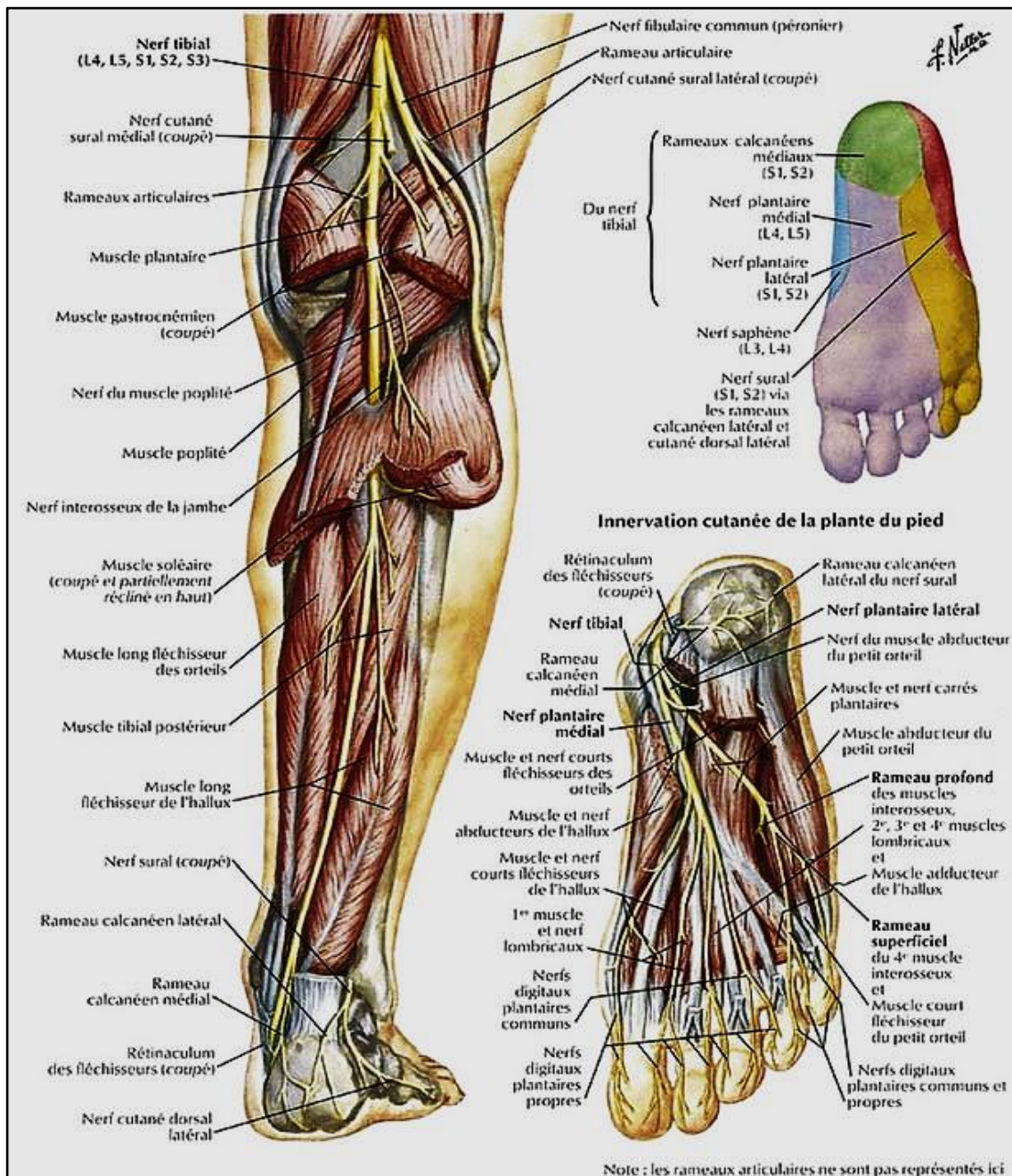


Figure I-10 : Nerfs tibial et fibulaire [18].

**d. Vascularisation du pied :**

Le dos du pied est vascularisé par l'artère dorsale du pied (Figure I.11). La plante du pied est irriguée par les artères plantaires latérale et médiale [14].



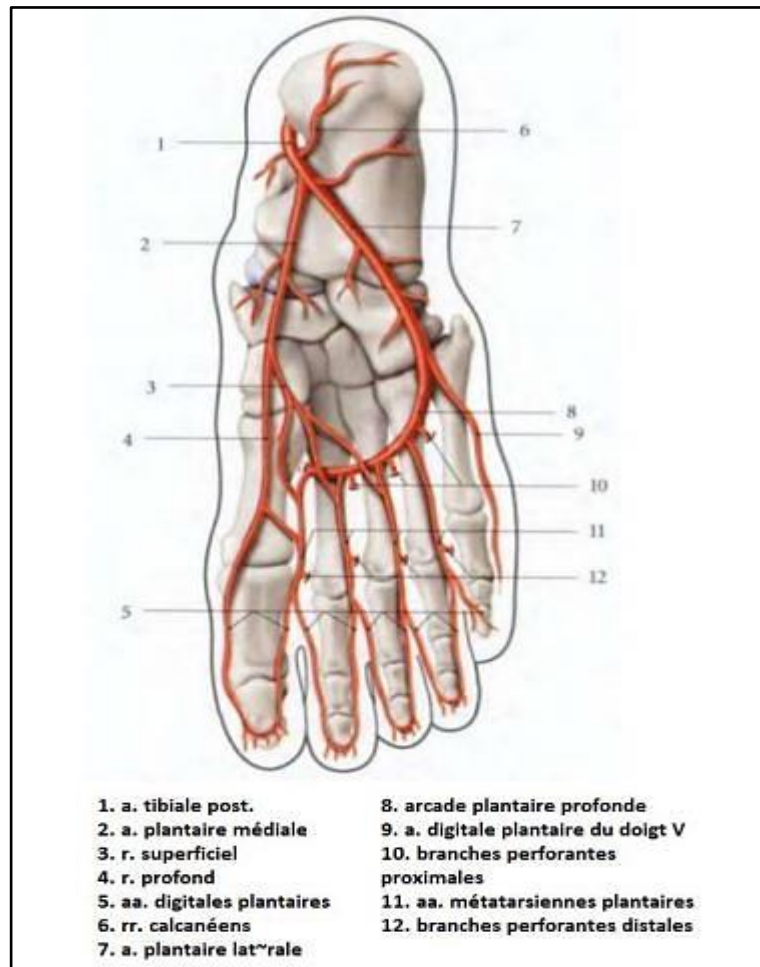


Figure I-11 : Artères plantaires [14].

### I.3.2 Physiopathologie

Les lésions du pied diabétiques sont la conséquence de plusieurs mécanismes physiopathologiques, qu'il est essentiel de connaître afin d'en prévenir l'apparition et d'en assurer une prise en charge précoce. La prévention et la prise en charge précoce et adaptée sont les seuls moyens d'éviter le risque d'amputation. Les principaux mécanismes à l'origine des lésions tiennent à deux complications du diabète, la neuropathie périphérique et l'artériopathie, qui peuvent être associées à des degrés divers, l'infection qui peut survenir sur ce terrain en présence d'une plaie, est un facteur de gravité [17].

#### a. Neuropathie

La neuropathie peut associer 3 types d'atteintes [19] :

- Les troubles de la sensibilité (tactile, thermique, algésique, profonde).

- Le déficit moteur responsable d'un déséquilibre entre les muscles extenseurs et les fléchisseurs du pied.
- L'atteinte végétative, autonome, source de sécheresse cutanée, d'œdèmes et de troubles vasomoteurs (ouvertures de shunts artério-veineux, perte de la vasomotion des capillaires).

Selon l'étude Rochester, la sévérité de la neuropathie est liée à la durée d'exposition à l'hyperglycémie. D'autres facteurs sont étudiés dans les recherches en cours portant sur le stress oxydatif, les produits finaux de glycation avancés, la protéine kinase C et la voie des polyols [13].

La neuropathie est très fréquemment à l'origine des lésions du pied car elle entraîne une perte de la sensibilité, favorisant les zones de frottements et d'hyperpression, et des troubles de la statique réalisant, à l'extrême, l'ostéoarthropathie nerveuse diabétique avec constitution d'un pied de Charcot [19]. Les figures suivantes (I.12 à I.14) montrent les conséquences d'une neuropathie.



**Figure I-12** : Pied de Charcot [14].



**Figure I-13** : Mal Perforant Plantaire Neuropathique [17]



**Figure I-14 :** 2<sup>ème</sup> Orteil en Griffé chez un Patient Diabétique [14].

### b. Artériopathie

Une artériopathie périphérique est retrouvée dans jusqu'à 50 % des cas. Elle affecte principalement les artères tibiales et péronières. L'hyperglycémie chronique entraîne une dysfonction endothéliale et des muscles lisses, conduisant à une vasoconstriction, encore potentiellement aggravée par les co-morbidités telles que l'hypertension artérielle, la dyslipidémie et le tabagisme via une occlusion de la lumière artérielle [20].

L'absence d'un pouls distal permet d'affirmer l'artériopathie mais la présence des pouls distaux ne permet pas de l'éliminer [17].

La présence d'un ulcère ischémique (Figure I.15) ou d'une gangrène (Figure I.16) signe l'artériopathie même si les pouls sont perçus [17].



**Figure I-15 :** ulcérations ischémiques [17].



**Figure I-16 :** gangrène sèche du gros orteil [17].

## c. Infection

La présence de plaie constitue une porte d'entrée infectieuse, et le risque de survenue d'infection augmente avec la durée d'évolution de la plaie. Cette infection peut toucher les tissus mous et/ou les os sous-jacents, et est favorisée par l'hyperglycémie. Elle met en danger le membre et la vie du patient, et nécessite un diagnostic et un traitement précoce [20].

Les plaies sont souvent colonisées, sans que la présence de ces bactéries soit significative d'une infection. Ainsi, les prélèvements superficiels ne sont pas suffisants au diagnostic positif et étiologique de l'infection de plaie du pied. Le diagnostic repose sur des critères cliniques : hyperthermie, sécrétions purulentes, et/ou présence d'au moins deux signes d'inflammation au niveau de la plaie.

Du fait de l'artériopathie et de la neuropathie, les signes d'inflammation peuvent être retardés, et une stagnation de diminution de la taille de l'ulcère doit faire suspecter une infection sous-jacente, de même que la présence d'un contact osseux, ou un déséquilibre brutal du diabète. L'imagerie, et en particulier l'IRM, peuvent aider au diagnostic, notamment d'ostéite.

L'infection se manifeste cliniquement selon différentes formes [17] :

- La dermo-hypodermite bactérienne (DHB) (Figure I.17).
- La dermo-hypodermite nécrosante (DHBN).
- La fasciite nécrosante (DHBNFN) (Figure I.18).
- La gangrène humide (Figure I.19).



**Figure I-17** : Nécrose avec Dermo-Hypodermite sur Pied Diabétique [17].



**Figure I-18** : Fasciite Nécrosante de la Jambe et de la Cheville chez un patient Diabétique [17].



**Figure I-19** : gangrène extensive du gros orteil [17].

La Figure ci-dessous, présente le cas d'un pied diabétique avec atteinte osseuse.



**Figure I-20** : Pied Diabétique avec Atteinte Osseuse [17].



La Figure I.21 résume les différents cas de physiopathologie rencontrés pour le pied diabétique.

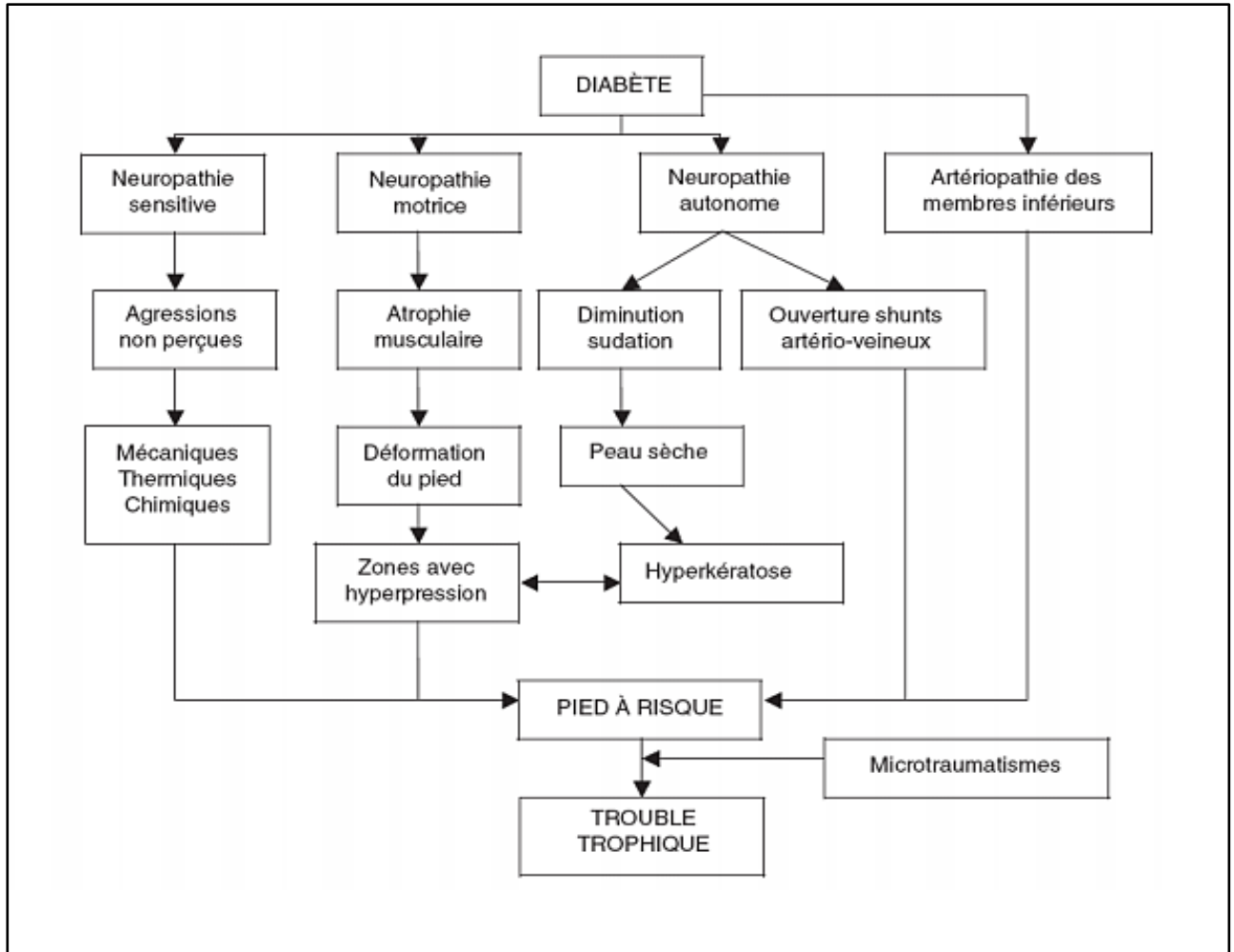


Figure I-21 : Physiopathologie du Pied Diabétique [14].

### I.3.3 L'ulcère du Pied Diabétique

L'ulcère du pied diabétique est une complication tardive et défigurante, qui entraîne un risque élevé d'amputation de n'importe quelle partie du pied ou de la jambe [21].

C'est une grotte d'infection, d'ischémie vasculaire sévère et de multiples traumatismes indolores. Les facteurs qui aggravent le problème sont l'âge avancé, la durée du diabète et le contrôle du diabète [21].

**a. Symptômes cliniques des ulcères du pied neuropathiques et ischémiques.**

Le Tableau suivant résume les différents types d'ulcère avec ses signes cliniques.

**Tableau I-1** : Symptômes cliniques des ulcères du pied neuropathiques et ischémiques [22].

Signes cliniques	Ulcère neuropathique 	Ulcère ischémique 
Déformation du pied	Orteils en griffe, possibilité d'une arche élevée, possibilité d'une déformation de Charcot	Pas de déformation particulière. Possibilité d'une amputation antérieure d'orteils ou de la partie avant du pied
Température du pied/Pouls du pied	Chaud, pouls palpable	Pieds froids ou température abaissée, le pouls peut être absent ou faible
Couleur de la peau	Normale ou rouge	Pâle/bleutée. Rougeur prononcée lorsqu'abaissé, blanchiment à l'élévation
État de la peau	Peau sèche en raison de la diminution de la transpiration	Mince, fragile et sèche
Emplacement de l'ulcère	Sur la face plantaire du pied et des orteils (80% sur l'avant-pied)	Bouts des orteils, talon, bords du pied
Présence de callosités	Fréquemment retrouvées sur les surfaces de mise en charge et généralement épaisses	Pas habituellement. si présent, croûte ou nécrose distale.
Caractéristiques de l'ulcère	Habituellement indolore, avec aspect enfoncé (granulation ou base plus profonde), entouré de callosités	Douloureux, particulièrement en présence de tissu nécrotique
Sensibilité	Sensation réduite ou absente au toucher, à la vibration, à la douleur et à la pression	Des sensations peuvent être présentes, mais diminuées, en présence d'une neuropathie associée
Réflexes achilléens	Généralement absents	Généralement présents
Pouls du pied	Présent et souvent bondissant. Veines saillantes et dilatées	Absent ou fortement affaibli

### b. Classification de l'ulcère du pied diabétique

**Système de classification de Wagner :** Comme le montre le Tableau ci-dessous, ce système s'est concentré sur les caractéristiques physiques de l'ulcère, la profondeur et la présence d'ostéomyélite ou de gangrène (0–5) [21].

**Tableau I-2 :** Système de classification de Wagner-Meggitt [23].

Stade	Lésion
0	Pied a risque élevé.
1	Ulcère superficiel et non infecté.
2	Ulcère profond sans atteinte osseuse.
3	Ulcère profond avec infection et atteinte des tendons/fascia et/ou des os.
4	Gangrène limitée (orteils, talons).
5	Gangrène étendue (humide ou sèche).

- **SINBAD :** Évalue le site, l'ischémie, la neuropathie, l'infection bactérienne et la profondeur et utilise un système de notation de 0 à 6 (**Tableau I-3**). Il s'est concentré sur les modifications pathologiques cliniques et macroscopiques de l'ulcère [21].

**Tableau I-3 :** Classification de SINBAD [14].

Catégorie	Définition	Score
Localisation	Avant-pied	0
	Au milieu du pied ou arrière-pied	1
Ischémie	Flux sanguin de la pédale intact: au moins une impulsion palpable	0
	Preuve clinique d'une réduction débit de la pédale	1
Neuropathie	Sensationprotectrice de la douleur intacte	0
	Sensation protectrice de la douleur perdue	1
Infection Bactérienne	Absente	0
	Présente	1
Surface	Ulcère <1cm <sup>2</sup>	0
	Ulcère ≥ 1cm <sup>2</sup>	1
Profondeur	Ulcère limité à la peau et tissu sous-cutané	0
	Ulcère atteignant le muscle, le tendon ou plus profond	1
Score total possible		6



- PEDIS** : Ce système a été conçu par le Groupe de travail international sur le pied diabétique et utilise les cinq composants : perfusion, étendue, profondeur, infection et sensation (voir **Tableau I-4**). Il n'inclut pas la localisation de l'ulcère [21].

**Tableau I-4** : Système de classification PEDIS [24].

<b>PERFUSION (VASCULARISATION)</b>	
Grade 1	pas de symptômes, pas de signes d'artériopathie périphérique (P1)
Grade 2	symptômes ou signes d'artériopathie périphérique, mais pas d'ischémie critique du membre (P2)
Grade 3	ischémie critique du membre (P3)
<b>EXTENT (ÉTENDUE)</b>	
	surface de la plaie mesurée en cm <sup>2</sup> après débridement
<b>DEPTH (PROFONDEUR)</b>	
Grade 1	ulcère superficiel limité au derme (D1)
Grade 2	ulcère profond, pénétrant sous le derme jusqu'aux structures sous-cutanées, impliquant les fascias, les muscles ou les tendons (D2)
Grade 3	toutes les structures du pied, y compris l'os et/ou l'articulation (contact osseux ou ulcère pénétrant jusqu'à l'os) (D3)
<b>INFECTION (INFECTION)</b>	
Grade 1	aucun symptôme ni signe d'infection (I1)
Grade 2	infection de la peau et des tissus sous-cutanés (I2)
Grade 3	érythème >2cm et infection plus profonde (I3)
Grade 4	infection avec signes systémiques (I4)
<b>SENSATION (SENSIBILITÉ)</b>	
Grade 1	aucune perte de la sensibilité de protection (S1)
Grade 2	perte de la sensibilité de protection (S2)

- Université du Texas :** La classification de l'Université du Texas, appelée aussi classification UT, a une meilleure valeur prédictive d'amputation chez les personnes diabétiques [24]. Elle est alphanumérique et tient compte de deux entrées en combinant un grade et un stade. Les quatre grades numériques vont de 0 à 3 en fonction de la profondeur de l'affection tissulaire. Les stades A à D renvoient à la présence d'infection ou d'ischémie de la plaie. Les risques d'amputation augmentent vers la droite de l'axe X et vers le bas de l'axe Y (voir **Tableau I-5**) [25].

**Tableau I-5 :** Classification de l'Université du Texas et pourcentages de risques d'amputation associés [25].

	<b>Grade 0</b> Phase pré ou postulcération avec épithélialisation	<b>Grade 1</b> Ulcère superficiel qui pénètre dans le derme	<b>Grade 2</b> Ulcère qui pénètre dans le derme, le tissu sous-cutané, le muscle, le tendon ou la capsule articulaire	<b>Grade 3</b> Ulcère qui pénètre jusqu'à l'os ou l'articulation
<b>Stade A</b> Pas d'infection Pas d'ischémie	<b>0A</b> 0 %	<b>1A</b> 0 %	<b>2A</b> 0 %	<b>3A</b> S/O
<b>Stade B</b> Avec infection Pas d'ischémie	<b>0B</b> 12,5 %	<b>1B</b> 8,5 %	<b>2B</b> 28,6 %	<b>3B</b> 92 %
<b>Stade C</b> Pas d'infection Avec ischémie	<b>0C</b> 25 %	<b>1C</b> 20 %	<b>2C</b> 25 %	<b>3C</b> 100 %
<b>Stade D</b> Avec infection Avec ischémie	<b>0D</b> 50 %	<b>1D</b> 50 %	<b>2D</b> 100 %	<b>3D</b> 100 %

Le Tableau I.6 présente les avantages et les inconvénients des systèmes de classifications citées ci-dessus.

Tableau I-6 : Caractéristiques des Systèmes de Classification pour les Ulcères du Pied Diabétique [14]

Système de classification	Points essentiels	Avantages/inconvénients
Wagner <sup>11</sup>	Evalue la profondeur de l'ulcère ainsi que la présence de la gangrène et la perte de perfusion à l'aide de six grades (0 à 5)	Bien établi Ne traite pas complètement de l'infection et de l'ischémie
Université du Texas <sup>12</sup>	Evalue la profondeur de l'ulcère, la présence de signes d'une ischémie des membres inférieurs à l'aide d'une matrice de quatre grades combinés à quatre stades	Bien établi Décrit mieux que Wagner la présence d'une infection et d'une ischémie et peut aider à prévoir l'issue de l'ulcère du pied diabétique
PEDIS <sup>13</sup>	Evalue la perfusion, l'étendue (la taille), la profondeur (perte tissulaire), l'infection et la sensation (neuropathie) au moyen de quatre grades (1 à 4)	Elaboré par l'IWGDF Facile d'utilisation (définitions claires, peu de catégories) pour les praticiens qui possèdent un faible niveau d'expérience dans la prise en charge du pied diabétique
SINBAD <sup>14</sup>	Evalue l'emplacement, l'ischémie, la neuropathie, l'infection bactérienne, la surface et la profondeur (par une cotation binaire 0/1)	Version simplifiée de classification Inclut notamment l'emplacement de l'ulcère et les données conduisant à penser qu'il pourrait s'agir d'un déterminant important de l'issue

## I.4 Conclusion

Le diabète et ses complications sont un problème majeur de santé publique. Il faut comprendre que le syndrome du pied diabétique ne s'agit pas d'une simple plaie, mais d'une complication sérieuse, pouvant mettre en danger la survie du pied ou de la jambe. Cette complication pose des problèmes de diagnostic précoce et de prise en charge.

L'évaluation et la prise en charge de l'ulcère du pied nécessitent une équipe médicale mais également paramédicale pluridisciplinaire et inter-métiers.

---

# **Chapitre II**

## **État de l'Art**

---

## II.1 Introduction

Le pied diabétique à risque est un véritable problème de santé publique dominé par un taux d'amputation des membres inférieurs très élevé. Ce qui peut causer des difficultés au patient pour le déplacement pour un rendez-vous chez son médecin traitant, donc pour lui faciliter la tâche, plusieurs méthodes ont été développées pour réaliser un diagnostic sur place en utilisant différents algorithmes et systèmes.

Dans ce chapitre, nous présentons un état de l'art classifié par ordre chronologique, qui contient les différentes méthodes utilisées pour la détection, l'extraction et la classification d'ulcère pour le pied diabétique.

## II.2 Méthodes de Détection et de Classification

- **Liu et al, 2015 [26]** : ont développé un système de télémédecine intelligent pour la détection des complications du pied diabétique avec l'imagerie spectrale, images thermiques infrarouges et reconstruction de surface 3D. Cependant, pour mettre en œuvre ce système, il y a une exigence de plusieurs appareils coûteux et formation spécialisée à l'utilisation de ces appareils ce qui présente un grand inconvénient.
- **Wang et al, 2016 [27]** : ont utilisé une boîte de capture d'image pour capturer des données d'image et déterminer la zone de DFU en utilisant une classification basée sur la machine à vecteur de support en cascade. Ils ont proposé l'utilisation de la méthode de clustering itératif linéaire simple (SLIC) pour une segmentation efficace des super-pixels et pour l'extraction des caractéristiques afin d'effectuer une classification en deux étapes. Parmi les inconvénients de cette méthode :
  - Elle n'a pas été validée sur un grand ensemble de données.
  - La boîte de capture d'image est très peu pratique pour la collecte de données car il est nécessaire d'assurer un contact entre les pieds du patient et la surface de la boîte, ce qui ne serait pas autorisé dans un établissement de soins de santé en raison de préoccupations concernant le contrôle des infections.

- **Brown et coll, 2017 [28]** : ont créé une application mobile appelée MyFootCare, qui tente de promouvoir les auto-soins des patients à l'aide d'objectifs personnels. L'application conserve un enregistrement photographique en série des pieds du patient. La segmentation DFU est complétée à l'aide d'un processus semi-automatisé, où l'utilisateur délimite manuellement l'emplacement DFU et le tissu cutané environnant. MyFootCare permet aux patients de prendre des photos de leurs pieds en plaçant le téléphone au sol. Le patient place son pied au-dessus de l'écran du téléphone, et la photo est automatiquement prise lorsque le pied est correctement positionné. Le retour vocal est utilisé pour guider l'utilisateur lors du positionnement de son pied.
  
- **Patel et al ,2017 [29]** : Leur travail est basé sur l'analyse des techniques de classification utilisées en imagerie médicale. En passant par le prétraitement, la segmentation, l'extraction de caractéristiques, la détection de texture puis la classification. Les techniques de classification telles que les réseaux bayésiens, les réseaux neuronaux, le classificateur K-nearest le plus proche, les techniques de logique floue et la machine à vecteur de support sont utilisées pour la classification de la région de la plaie. Cette étude fournit un système de détection des ulcères du pied pour la reconnaissance et la classification des ulcères du pied chez les personnes diabétiques. Le système de détection et de classification du tissu de la plaie de l'ulcère du pied diabétique est mis en œuvre à l'aide de MATLAB. Dans ce travail, les étapes nécessaires à l'identification de l'ulcère du pied à l'aide du traitement d'images sont examinées.
  
- ✓ **Avantages :**
  - Le filtre de Gabor est utilisé pour la segmentation, car il permet une localisation optimale de l'articulation dans les domaines fréquentiel et spatial et pour la classification ils ont utilisé le k-mean.
  - Avec le résultat de la classification, il est montré que trois types de tissus sont identifiés et étiquetés.
  - Sur la base de cette classification des tissus un traitement approprié est donné au patient souffrant d'un ulcère du pied pour une guérison efficace et rapide de la plaie.

- **M. Goyal et al, 2017, [30]** : Dans ce travail, ils ont développé des approches d'apprentissage en profondeur pour former divers modèles de réseaux entièrement convolutifs FCNs (Fully Convolutional Network) capables de détecter et de segmenter automatiquement le DFU et la zone cutanée environnante avec un degré élevé de précision. Ils ont proposé des modèles pré entraînés sur des ensembles de données importants pour former les réseaux entièrement convolutifs.

Les différentes architectures utilisées sont :

- FCN-AlexNet.
- FCN-32s.
- FCN-16s.
- FCN-8s.

Ce travail sera utile pour faire la classification multi-classes et la segmentation

- **M. Goyal et al, 2018, [31]** : ont proposé l'utilisation des fonctionnalités traditionnelles de vision par ordinateur pour détecter les ulcères du pied chez les patients diabétiques. Dans ce problème de classification DFU, ils ont évalué les deux classes en tant que peau normale (peau saine) et peau anormale (DFU).

Par ailleurs, ils ont utilisé les réseaux de neurones convolutifs pour la première fois dans la classification DFU, ils ont proposé une nouvelle architecture de réseau de neurones convolutifs, DFUNet, avec une meilleure extraction des caractéristiques pour identifier les différences de caractéristiques entre une peau saine et le DFU. Le Tableau suivant montre les différentes couches de l'architecture utilisée.

**Tableau II-1** : L'architecture de la Méthode proposée par M. Goyal et al. [31]

Layer no.	Layer type	Filter size	Stride	No. of filters	FC units	Input	Output
Layer 1	Conv.	7×7	2×2	64	-	3×224×224	64×112×112
Layer 2	Max-pool.	3×3	2×2	-	-	64×112×112	64×56×56
Layer 3	Conv.	1×1	1×1	64	-	64×56×56	64×56×56
Layer 4	Conv.	3×3	1×1	192	-	64×56×56	192×56×56
Layer 5	Max-pool.	3×3	2×2	-	-	192×56×56	192×28×28
Layer 6	Parallel conv.	1×1,3×3,5×5	1×1	32⊕64⊕128	-	192×28×28	224×28×28
Layer 7	Max-pool.	3×3	2×2	-	-	224×28×28	224×14×14
Layer 8	Parallel conv.	1×1,3×3,5×5	1×1	32⊕64⊕128	-	224×14×14	224×14×14
Layer 9	Parallel conv.	1×1,3×3,5×5	1×1	32⊕64⊕128	-	224×14×14	224×14×14
Layer 10	Max-pool.	3×3	2×2	-	-	224×14×14	224×7×7
Layer 11	Parallel conv.	1×1,3×3,5×5	1×1	32⊕64⊕128	-	224×7×7	224×7×7
Layer 12	Max-pool.	7×7	1×1	-	-	224×7×7	224×1×1
Layer 13	Fully conn.	-	-	-	1000	-	-
Layer 14	Fully conn.	-	-	-	No. of Classes	-	-



---

✓ **Inconvénients**

- DFUN a généralement du mal à classer la peau pré-ulcéreuse et la détecte généralement comme normale. De plus, les DFU qui sont de très petite taille sont mal classées comme normales.
  - Pour la peau normale, les plaques avec orteil, peau très ridée et peau très rouge sont mal classées par la méthode proposée.
- **Babu et al, 2018 [32]** : Ont travaillé avec la méthode de PSO (particle swarm optimization) qui est une méthode de calcul qui optimise un problème qui a pour but de segmenter la plaie du pied diabétique. Le mouvement de chaque de chaque particule est affecté par son meilleur local, une position connue. C'est un processus stochastique basé sur la population qui résout les problèmes d'optimisation continus et discrets. Ils ont ensuite classifié la région d'intérêt en passant par l'extraction des caractéristiques puis une classification avec Naive Bayes.
- **Daniel Alejandro et al, 2019 [33]** : Dans ce travail, la conception d'une base de données de thermogrammes plantaires est présentée. La base de données est composée de 334 Thermogrammes individuels provenant de 122 sujets diabétiques et de 45 sujets non diabétiques. Chaque thermogramme comprend quatre images supplémentaires correspondant aux angiosomes plantaires, et chaque image est accompagnée de sa température. L'extraction des angiosomes permet d'analyser la température de ces régions et peut être utilisée pour identifier les changements possibles dans l'angiosome dont l'approvisionnement en sang qui peuvent indiquer un risque d'ulcération. En outre, le document présente certaines techniques de segmentation, d'enregistrement et de correction de la posture des pieds présentées dans des travaux antérieurs, en soulignant les différences et les problèmes qui n'ont pas été résolus.
- **M. Goyal et al, 2019 [34]** : Ont formé et validé un modèle d'apprentissage profond supervisé capable de localiser les DFU. Le backbone utilisé était Faster R-CNN avec Inception V2. Ce modèle de classificateur unique a été formé à l'aide d'un apprentissage par transfert à deux niveaux (partiel et complet) avec l'ensemble de données MS COCO.



---

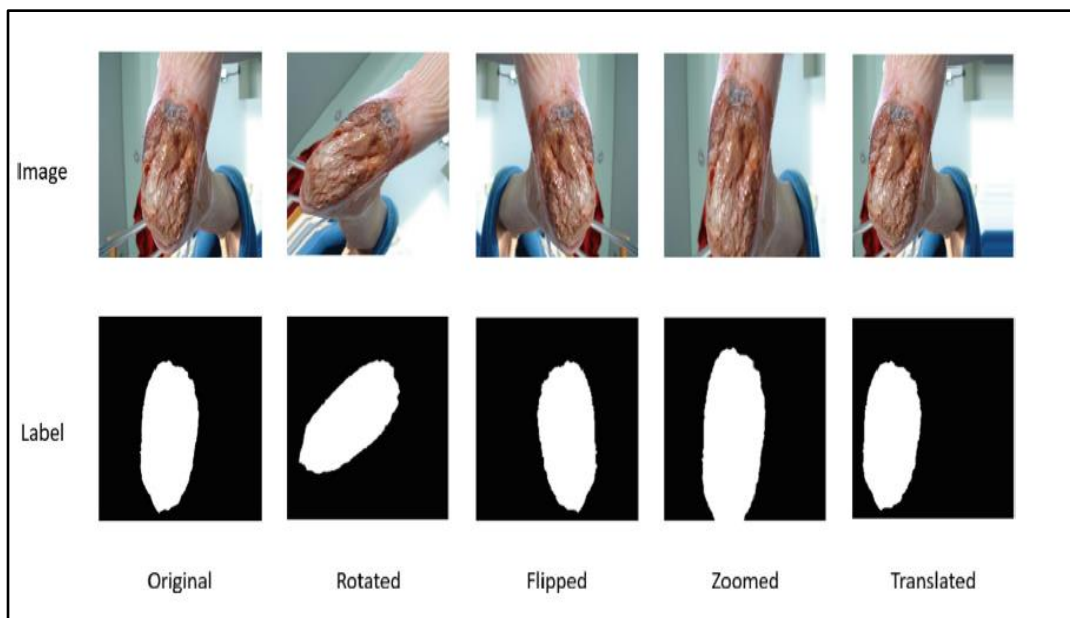
**✓ Avantages**

- Cette méthode est capable de plusieurs détections par image, et a démontré une MAP élevée pour des paramètres expérimentaux.
- **L.Alzubaid et al, 2019 [35]** : Dans cet article, ils ont rassemblé un grand ensemble de données de 754 pieds de patients atteints d'UDP et de peau saine. Un expert a divisé le jeu de données en deux catégories : normale et aberrante. Leur jeu de données a été utilisé pour entraîner le modèle QUTN et le proposé pour les UDP. Les classificateurs SVM et KNN ont été formés en utilisant les caractéristiques récupérées par leur modèle proposé. Ils ont comparé les performances de leur réseau à celles de réseaux CNN de pointe (GoogleNet, AlexNet et VGG16). Ces réseaux CNN ont été formés sur le jeu de données ImageNet, qui comporte un grand nombre de données. Ils les ont réentraînés sur leur jeu de données après les avoir réglés finement, en transférant l'apprentissage précédent de ces réseaux d'apprentissage profond. Le cadre proposé, DFU QUTNet avec SVM, a un score F1 plus élevé de 94,5 % que les approches de pointe, selon la comparaison.
- **M. Goyal et al, 2020[36]** : proposent un nouvel ensemble de données et des techniques de vision par ordinateur pour identifier la présence d'infection et d'ischémie dans DFU, Pour l'approche d'apprentissage automatique artisanal, ils proposent un nouveau descripteur de fonctionnalités, à savoir le Descripteur de couleur Superpixel. Ensuite, nous utilisons le modèle CNN (Ensemble Convolutional Neural Network) pour une reconnaissance plus efficace de l'ischémie et de l'infection. Nous proposons d'utiliser une méthode naturelle d'augmentation des données, qui identifie la région d'intérêt sur des images de pied et se concentre sur la recherche des traits saillants existant dans cette région. Enfin, nous évaluons les performances de nos techniques proposées sur la classification binaire, c'est-à-dire ischémie versus non-ischémie et infection versus non-infection.

**✓ Inconvénients**

- une performance modérée des classificateurs dans la classification de l'infection.

- **Niri Rania et al, 2020 [37]** : Dans ce travail, ils ont effectué une segmentation de l'UFC à l'aide de Deep Learning pour la segmentation sémantique. Leur objectif était de trouver un réseau neuronal entièrement convolutif précis et adapté à une petite base de données. Trois différents réseaux entièrement convolutifs ont été testés pour effectuer la segmentation de la zone de l'ulcère. Le réseau U-Net a obtenu un coefficient de similarité de 97,25 % et un indice d'intersection sur union de 94,86 %. Ces résultats préliminaires démontrent la puissance des réseaux neuronaux entièrement convolutifs dans la segmentation des ulcères du pied diabétique à l'aide d'un réseau de type U-Net et en utilisant un nombre limité d'échantillons d'entraînement. Le nombre d'échantillons de la base de données est limité et ne suffit pas à alimenter un réseau neuronal d'apprentissage profond. En raison des limitations des ressources de calcul et de mémoire, le modèle U-net a été entraîné pendant seulement 5 époques. Ils ont utilisé une data augmentation pour obtenir une base de donnée plus grande et pour augmenter la taille de l'ensemble d'apprentissage et éviter le sur-ajustement. Chaque image a été retournée horizontalement, pivotée, translatée et zoomée. Après l'augmentation des données, la taille de la base de données a été multipliée par quatre et contient maintenant 368 images au lieu de 92. Le résultat du data augmentation est montré dans la figure suivante.



**Figure II-1** : Résultat data augmentation [37].

- **Javaria amin et al ,2020 [38]** : Dans cet article, deux expériences ont été réalisées pour évaluer la méthode proposée. D'abord pour la classification de l'infection et d'ischémie, et la seconde pour la localisation des régions infectées qui sont obtenues après la classification. Dans ce travail, des approches d'apprentissage profond sont présentées pour la localisation et la classification des UPD. Le modèle CNN à 16 couches est plus performant car les résultats dépendent de la combinaison des couches de convolution sélectionnées. Pour la localisation, le modèle YOLOv2-DFU est conçu en utilisant un réseau shuffle comme colonne vertébrale du modèle YOLOv2. Les résultats de la localisation ont prouvé qu'une combinaison du shuffle-net et du modèle YOLOv2 est plus efficace pour localiser la région infectée dans les images de pieds. La Figure suivante montre le modèle CNN utilisée dans la classification d'UPD.

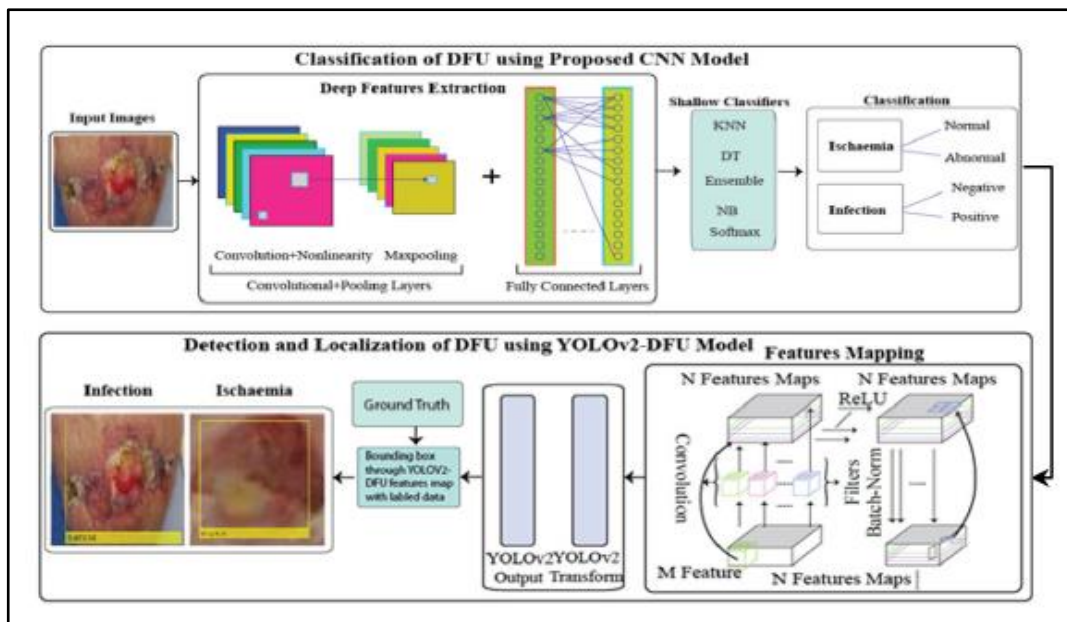


Figure II-2 : Modèle CNN pour la classification d'UPD [38].

- **Le modèle CNN :** Le modèle de réseau neuronal convolutif proposé, classe les images de pieds en positif/négatif. Et fournit également une aide pour trouver le modèle d'image pour reconnaître les symptômes de la lésion. Le modèle CNN proposé est entraîné sur les images de pieds diabétiques à partir de zéro. Dans ce modèle, des couches convolutionnelles sont utilisées avec 5 tailles de filtre pour activer les caractéristiques.

- **YOLOv2-DFU** : Le modèle YOLOv2-DFU prédit les étiquettes de classe à l'aide d'une boîte d'ancrage (Figure II.3).

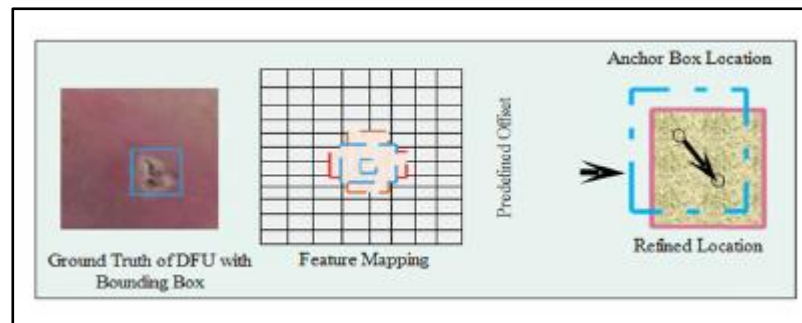


Figure II-3 : architecture YOLOv2 [38].

- **YT Chino et al, 2020 [39]** : Dans cet article, ils ont proposé le cadre ASURA pour segmenter et estimer la zone de la plaie d'une image d'ulcère cutané. ASURA utilise un réseau neuronal profond encodeur / décodeur pour segmenter la plaie à partir de l'image. La comparaison d'ASURA avec des méthodes basées sur des algorithmes de traitement d'image traditionnels, tels que CL-Measure et ICARUS, a révélé que les approches d'apprentissage en profondeur présentaient de bien meilleurs résultats. ASURA détecte également la règle / bande de mesure présente dans l'image et estime automatiquement la densité de pixels de l'image. Cela permet une mesure précise de la taille des plaies. En conséquence, les principales contributions de cet article sont :

- **Segmentation** : ASURA a pu segmenter correctement les régions de la plaie avec un score Dice supérieur à 90%, soit jusqu'à 16% meilleur que les méthodes de pointe (QTDU et DeepLabv3+) dans les deux ensembles de données (ULCER-BOTH).
- **Mesure de surface** : ASURA a été en mesure d'estimer automatiquement la densité de pixels des images avec une erreur relative de 5,6% et d'estimer semi-automatiquement la zone de la plaie en  $cm^2$  avec une erreur relative de 12,1%. À notre connaissance, il s'agit d'une nouvelle ressource, très recherchée par les médecins et les spécialistes de la santé.

La Figure suivante présente l'architecture de cette approche.

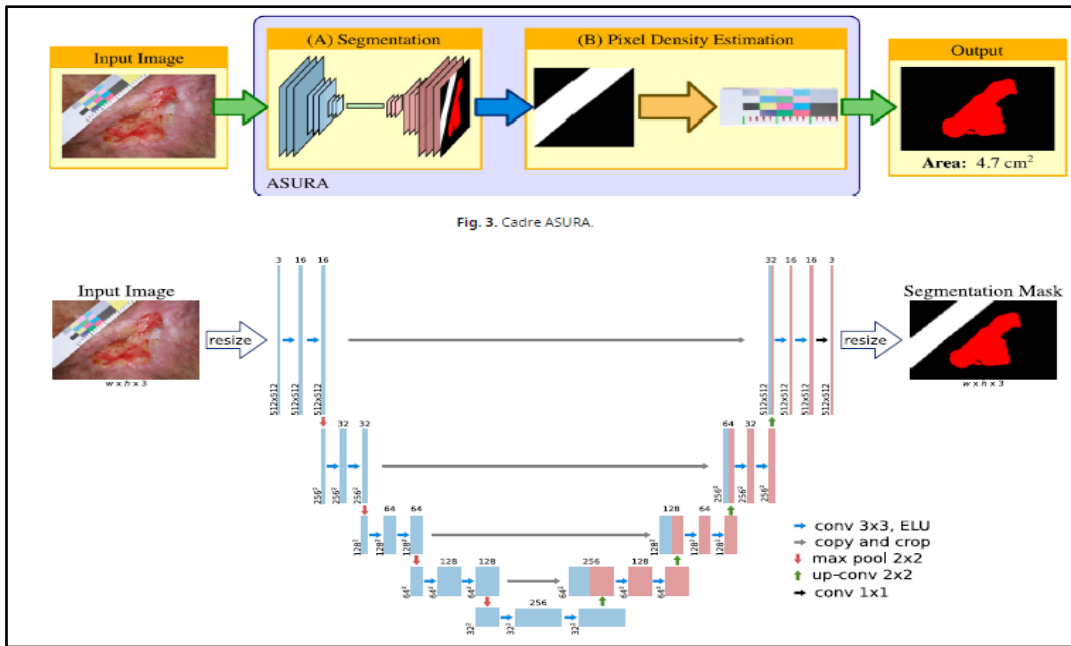


Figure II-4 : L'architecture du réseau utilisé par ASURA [39].

- **L. Alzubaidi et al ,2021 [40]** : L'approche est basée sur l'entraînement des modèles d'apprentissage profond sur un grand nombre d'images non étiquetées d'une tâche spécifique. Puis en affinant le modèle pour l'entraîner sur un petit nombre d'images étiquetées de la même tâche. Ils ont conçu un modèle DCNN hybride reposant sur plusieurs idées, dont des couches convolutives parallèles et des connexions résiduelles, ainsi qu'une mise en commun de la moyenne globale. Ils ont prouvé empiriquement l'efficacité de l'approche et du modèle proposés en les appliquant à deux tâches difficiles, le cancer de la peau et le cancer du sein. Ils ont obtenu d'excellents résultats dans les deux tâches :

- Dans la tâche de classification du cancer de la peau, le modèle proposé a obtenu une valeur de score F1 de 89,09 % lorsqu'il est entraîné à partir de zéro et 98,53 % avec l'approche proposée.
- Le modèle proposé a obtenu un score F1 de 85,29 % lorsqu'il a été entraîné à partir de zéro et de 97,51 % avec l'approche proposée.

En outre, Ils ont introduit une autre technique d'apprentissage par transfert appelée double-transfert. Ils l'ont utilisée pour améliorer la performance de la tâche de classification DFU, et ont obtenu un score F1 de 99,25%.

- **Moi Hoon Yap et al, 2021 [41] :** Ils ont examiné les performances du réseau avec et sans le modèle pré-entraîné et ils ont constaté que le modèle pré-entraîné est plus performant que la plupart des résultats. DenseNet121 a obtenu la meilleure performance globale, avec un score F1 macro-moyen de 0,55 et une AUC de 0,88. EfficientNetB0 est la deuxième meilleure performance, avec un score F1 de 0,53 et une AUC de 0,84 (précision et rappel plus élevés qu'InceptionV3). Ils ont choisi deux des meilleurs modèles, DenseNet121 et EfficientNetB0, et ils ont effectué d'autres tests d'augmentation des données sur la base de ces résultats. Les meilleures valeurs pour l'ensemble d'entraînement.
  
- ✓ **Inconvénients :** La détection de l'infection et la détection de la cooccurrence de l'ischémie restent un plus grand défi. Cela est dû aux difficultés à détecter l'infection dans les "deux" catégories, car elle est souvent classée comme l'ischémie, avec ou sans augmentation des données.

### II.3 Conclusion

Dans ce chapitre nous avons regroupé les différentes méthodes qui ont été utilisées pour détecter et classifier la plaie du pied diabétique. Afin de choisir une méthode pertinente qui nous aidera dans notre travail.

D'après les articles que nous avons lus et analysés, nous avons constaté que les réseaux de neurones convolutifs se sont avérés très efficaces pour le traitement de données et dans les domaines tels que la reconnaissance et la classification.

Dans le chapitre suivant nous allons approfondir et donner plus de détails sur l'algorithme CNN choisi et sur chaque technique en expliquant chaque étape.

---

# **Chapitre III**

## **L'algorithme CNN**

---



### III.1 Introduction

Après avoir fait la synthèse des différentes méthodes utilisées pour traiter notre problème, nous avons opté pour une méthode basée sur le « deep learning », plus précisément, faire un apprentissage profond par le CNN (Convolutional Neural Network) en appliquant un apprentissage par transfert. Les différents algorithmes seraient expliqués dans la suite de ce chapitre.

### III.2 Le Deep Learning

Le Deep Learning est un nouveau domaine de recherche du Machine Learning, qui a été introduit dans le but de rapprocher le ML de son objectif principal : L'intelligence artificielle.

Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données [42].

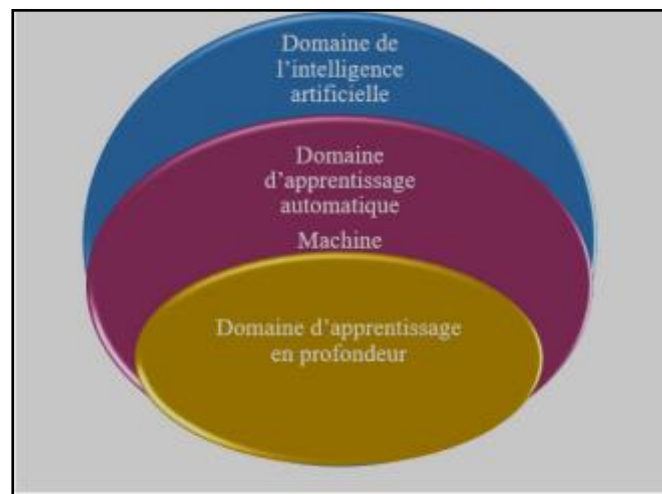


Figure III-1 : Relation entre les 3 Domaines [42].

### III.3 Réseaux de Neurones

Pour bien comprendre les réseaux de neurones convolutifs, il est important de connaître les bases des réseaux de neurones. Les principaux éléments à retenir sont les suivants [43] :

- Un réseau de neurones est un système composé de neurones, généralement répartis en plusieurs couches connectées entre elles.

- Un tel système s'utilise pour résoudre divers problèmes statistiques, mais nous nous intéressons ici qu'au problème de classification. Dans ce cas, le réseau calcule à partir de l'entrée un score (ou probabilité) pour chaque classe. La classe attribuée à l'objet en entrée correspond à celle de score le plus élevé.
- Chaque couche reçoit en entrée des données et les renvoie transformées. Pour cela, elle calcule une combinaison linéaire puis applique éventuellement une fonction non-linéaire, appelée fonction d'activation. Les coefficients de la combinaison linéaire définissent les paramètres (ou poids) de la couche.
- Un réseau de neurones est construit en empilant les couches : la sortie d'une couche correspond à l'entrée de la suivante.
- Cet empilement de couches définit la sortie finale du réseau comme le résultat d'une fonction différentiable de l'entrée.
- La dernière couche calcule les probabilités finales en utilisant pour fonction d'activation la fonction logistique (classification binaire) ou la fonction *softmax* (classification multi-classes).
- Une fonction de perte (*loss function*) est associée à la couche finale pour calculer l'erreur de classification. Il s'agit en général de l'entropie croisée.
- Les valeurs des poids des couches sont apprises par rétropropagation du gradient : on calcule progressivement (pour chaque couche, en partant de la fin du réseau) les paramètres qui minimisent la fonction de perte régularisée. L'optimisation se fait avec une descente du gradient stochastique.

### III.4 Les Réseaux de Neurones Convolutifs

Les réseaux de neurones convolutifs désignent une sous-catégorie de réseaux de neurones : ils présentent donc toutes les caractéristiques listées ci-dessus. Cependant, les CNN sont spécialement conçus pour traiter des images en entrée [43].

### III.4.1 Architecture de Réseaux de Neurone Convolutifs

Elle est composée de deux blocs principaux :

➤ **Le Premier Bloc**

Il fait la particularité de ce type de réseaux de neurones, puisqu'il fonctionne comme un **extracteur de caractéristiques**. Pour cela, il effectue du *template matching* en appliquant des opérations de filtrage par convolution. La première couche filtre l'image avec plusieurs noyaux de convolution, et renvoie des "*feature maps*", qui sont ensuite normalisées (avec une fonction d'activation) et/ou redimensionnées [43].

Ce procédé peut être réitéré plusieurs fois : nous filtrons les *features maps* obtenues avec de nouveaux noyaux, ce qui nous donne de nouvelles *features maps* à normaliser et redimensionner, et que nous pouvons filtrer à nouveau, et ainsi de suite. Finalement, les valeurs des dernières *feature maps* sont concaténées dans un vecteur. Ce vecteur définit la sortie du premier bloc, et l'entrée du second [43].

➤ **Le Second Bloc**

Il n'est pas une caractéristique d'un CNN, il se retrouve en fait à la fin de tous les réseaux de neurones utilisés pour la classification. Les valeurs du vecteur en entrée sont transformées (avec plusieurs combinaisons linéaires et fonctions d'activation) pour renvoyer un nouveau vecteur en sortie [43].

Ce dernier vecteur contient autant d'éléments qu'il y a de classes : l'élément représente la probabilité que l'image appartienne à la classe. Chaque élément est donc compris entre 0 et 1, et la somme de tous vaut 1 [43].

Ces probabilités sont calculées par la dernière couche de ce bloc (et donc du réseau), qui utilise une fonction logistique (classification binaire) ou une fonction *softmax* (classification multi-classes) comme fonction d'activation [43].

Comme pour les réseaux de neurones ordinaires, les paramètres des couches sont déterminés par rétropropagation du gradient : l'entropie croisée est minimisée lors de la phase d'entraînement. Mais dans le cas des CNN, ces paramètres désignent en particulier les *features* des images [43].

La Figure suivante montre les 2 blocs du CNN.

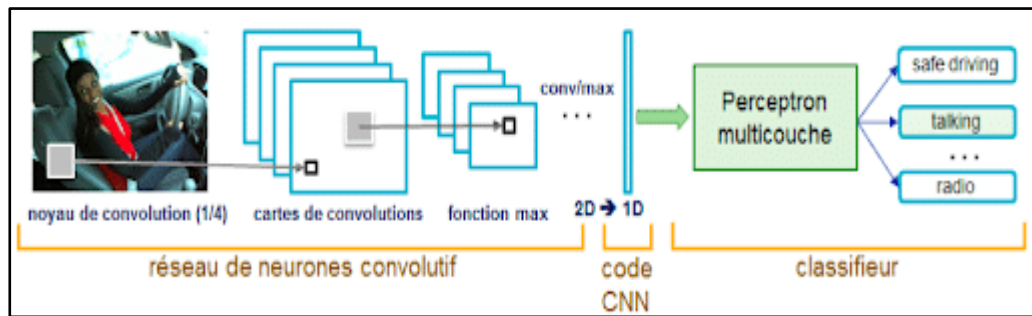


Figure III-2 : Schéma représentant l'architecture d'un CNN [44].

### III.4.2 Les Différentes Couches de CNN

Il existe quatre types de couches pour un réseau de neurones convolutif : la couche de **convolution**, la couche de **pooling**, la couche de **correction ReLU** et la couche **fully-connected** [45].

La Figure III-3 montre ces différentes couches.

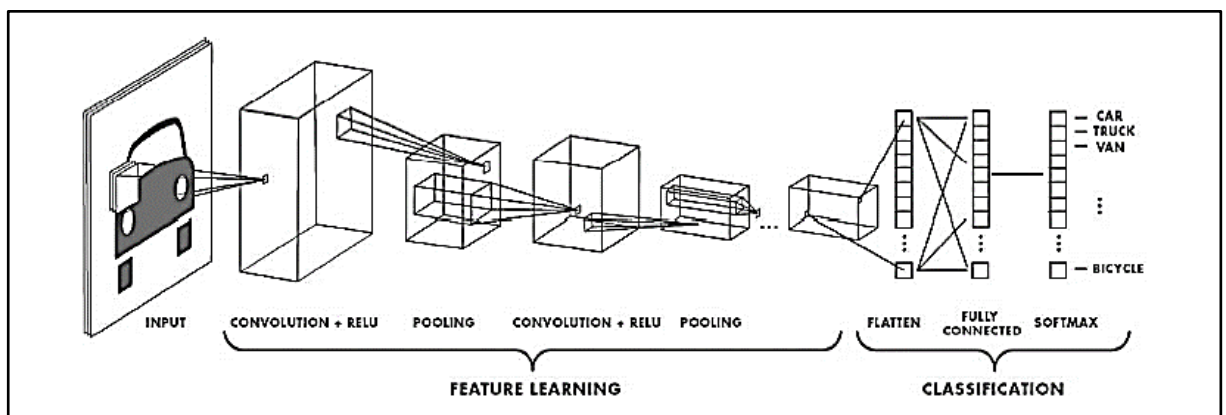


Figure III-3 : Illustration des différentes couches de CNN [42].

#### ➤ La Couche de Convolution

Elle est la composante clé des réseaux de neurones convolutifs, et constitue toujours au moins leur première couche [45].

Son but est de repérer la présence d'un ensemble de *features* dans les images reçues en entrée. Pour cela, nous réalisons un filtrage par convolution : le principe est de faire "glisser" une fenêtre représentant la *feature* sur l'image, et de calculer le produit de convolution entre la feature et chaque portion de l'image balayée. Une *feature* est alors vue comme un filtre : les deux termes sont équivalents dans ce contexte [45].

La couche de convolution reçoit donc en entrée plusieurs images, et calcule la convolution de chacune d'entre elles avec chaque filtre. Les filtres correspondent exactement aux features que l'on souhaite retrouver dans les images [45].

Nous obtenons pour chaque paire (image, filtre) une carte d'activation, ou feature map, qui nous indique où se situent les features dans l'image : plus la valeur est élevée, plus l'endroit correspondant dans l'image ressemble à la feature [45].

Contrairement aux méthodes traditionnelles, les features ne sont pas prédéfinies selon un formalisme particulier, mais apprises par le réseau lors la phase d'entraînement. Les noyaux des filtres désignent les poids de la couche de convolution et sont initialisés puis mis à jour par rétropropagation du gradient [45].

C'est là toute la force des réseaux de neurones convolutifs : ceux-ci sont capables de déterminer tout seul les éléments discriminants d'une image, en s'adaptant au problème posé. Par exemple, si la question est de distinguer les chats des chiens, les features automatiquement définies peuvent décrire la forme des oreilles ou des pattes [45].

La Figure III.4 montre un exemple d'une couche 2D.

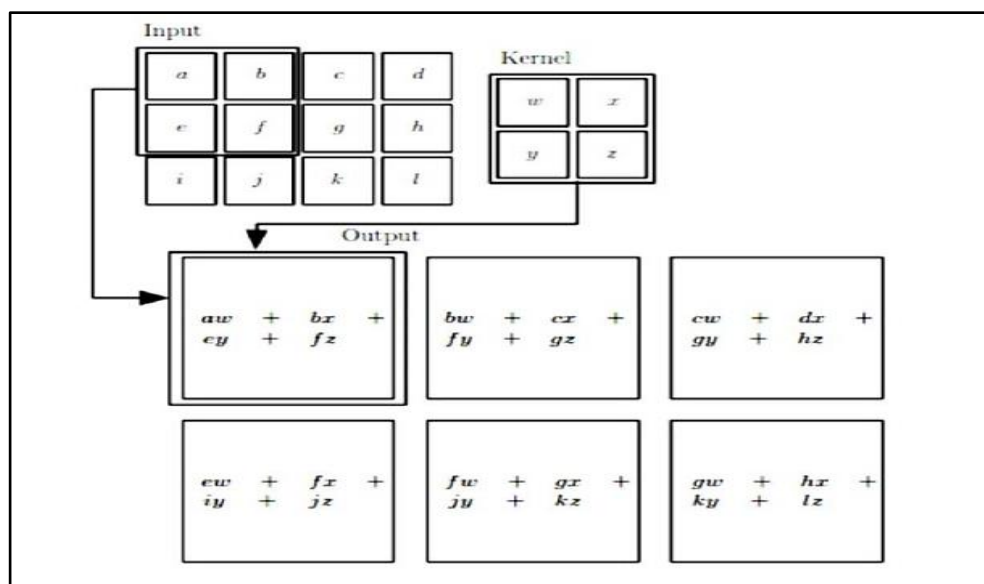


Figure III-4 : Exemple d'une convolution 2D [42].

➤ **La Couche de Pooling**

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs *feature maps*, et applique à chacune d'entre elles l'opération de *pooling* [45]. L'opération de *pooling* consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes [45].

Pour cela, nous découpons l'image en cellules régulières, puis nous gardons au sein de chaque cellule la valeur maximale. En pratique, nous utilisons souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille  $2 \times 2$  pixels qui ne se chevauchent pas, ou des cellules de taille  $3 \times 3$  pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc) [45]. Nous obtenons en sortie le même nombre de *feature maps* qu'en entrée, mais celles-ci sont bien plus petites [45].

La couche de *pooling* permet de réduire le nombre de paramètres et de calculs dans le réseau. Nous améliorons ainsi l'efficacité du réseau et nous évitons le sur-apprentissage (Overfitting) [45]. Les valeurs maximales sont repérées de manière moins exacte dans les *feature maps* obtenues après *pooling* que dans celles reçues en entrée [45].

La Figure III-5 présente un exemple d'une couche Pooling.

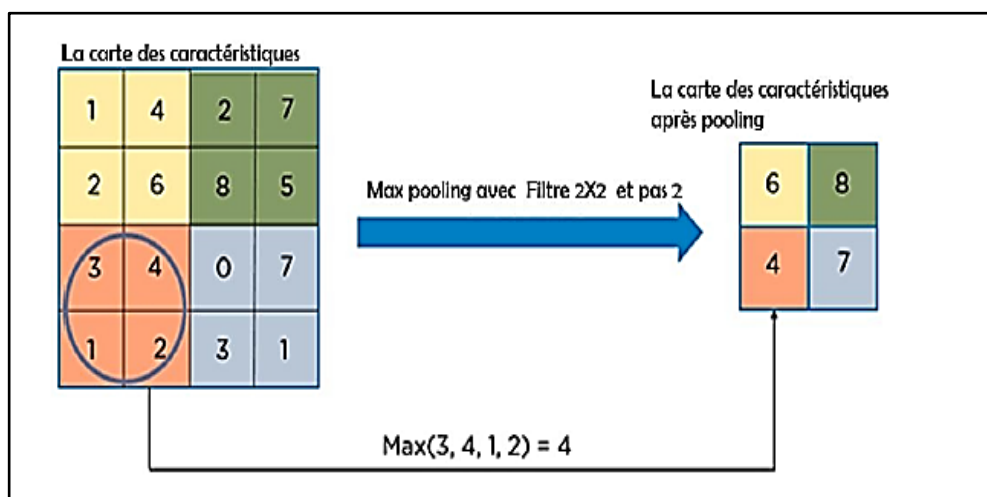


Figure III-5 : Max Pooling avec un filtre 2x2 et un pas de 2 [42].

➤ **La Couche de Correction ReLU**

ReLU (*Rectified Linear Units*) désigne la fonction réelle non-linéaire définie par la relation :  $\text{ReLU}(x) = \max(0, x)$  [45]. La Figure suivante montre la courbe de la couche ReLU

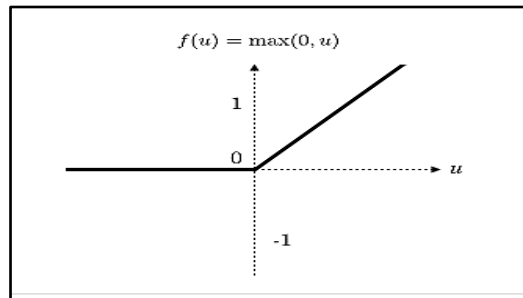


Figure III-6 : La courbe de la couche ReLU [45]

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation [45].

➤ **La Couche fully-connected**

La couche fully-connected constitue toujours la dernière couche d'un réseau de neurones, convolutif ou non – elle n'est donc pas une caractéristique d'un CNN [45].

Ce type de couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie. Pour cela, elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée [45].

La dernière couche fully-connected permet de classifier l'image en entrée du réseau : elle renvoie un vecteur de taille NN, où NN est le nombre de classes dans notre problème de classification d'images. Chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe [45].

Chaque valeur du tableau en entrée "vote" en faveur d'une classe. Les votes n'ont pas tous la même importance : la couche leur accorde des poids qui dépendent de l'élément du tableau et de la classe [45].

Pour calculer les probabilités, la couche fully-connected multiplie donc chaque élément en entrée par un poids, fait la somme, puis applique une fonction d'activation (logistique si  $N=2$ , softmax si  $N>2$ ) [45].



### III.5 Apprentissage par transfert (Transfer learning)

Entraîner un réseau de neurones convolutif est très coûteux : plus les couches s'empilent, plus le nombre de convolutions et de paramètres à optimiser est élevé.

L'ordinateur doit être en mesure de stocker plusieurs Giga octets de données et de faire efficacement les calculs. C'est pourquoi les fabricants de matériel informatique multiplient les efforts pour fournir des processeurs graphiques (GPU) performants, capables d'entraîner rapidement un réseau de neurones profond en parallélisant les calculs [46].

Le Transfer Learning (ou apprentissage par transfert) permet de faire du Deep Learning sans avoir besoin d'y passer un mois de calculs. Le principe est d'utiliser les connaissances acquises par un réseau de neurones lors de la résolution d'un problème afin d'en résoudre un autre plus ou moins similaire. Nous réalisons ainsi un transfert de connaissances, d'où le nom [46].

En plus d'accélérer l'entraînement du réseau, le Transfer Learning permet d'éviter le sur-apprentissage (overfitting). En effet, lorsque la collection d'images en entrée est petite, il est vivement déconseillé d'entraîner le réseau de neurones en partant de zéro (c'est-à-dire avec une initialisation aléatoire) : le nombre de paramètres à apprendre étant largement supérieur au nombre d'images, le risque d'overfitting est énorme [46].

Le Transfer Learning est une technique très utilisée en pratique et simple à mettre en œuvre. Elle nécessite d'avoir un réseau de neurones déjà entraîné, de préférence sur un problème proche de celui que nous voulons résoudre. De nos jours, nous pouvons facilement en récupérant un sur Internet, et notamment dans les bibliothèques de Deep Learning, comme Keras que nous allons utiliser dans notre implémentation [46].

Nous pouvons exploiter le réseau de neurones pré-entraîné de plusieurs façons, en fonction de la taille du jeu de données en entrée et de sa similarité avec celui utilisé lors du pré-entraînement [46].

#### ➤ **Stratégie 1 : Fine-tuning total**

Nous remplaçons la dernière couche fully-connected du réseau pré-entraîné par un classifieur adapté au nouveau problème (SVM, régression logistique...) et initialisé de manière aléatoire. Toutes les couches sont ensuite entraînées sur les nouvelles images [46].

La stratégie 1 doit être utilisée lorsque la nouvelle collection d'images est grande : dans ce cas, nous pouvons se permettre d'entraîner tout le réseau sans courir le risque d'overfitting. De plus, comme les paramètres de toutes les couches (sauf de la dernière) sont initialement ceux du réseau pré-entraîné, la phase d'apprentissage sera faite plus rapidement que si l'initialisation avait été aléatoire [46].

### ➤ **Stratégie 2 : Extraction des features**

Cette stratégie consiste à se servir des features du réseau pré-entraîné pour représenter les images du nouveau problème. Pour cela, nous retirons la dernière couche fully-connected et nous fixons tous les autres paramètres. Ce réseau tronqué va ainsi calculer la représentation de chaque image en entrée à partir des features déjà apprises lors du pré-entraînement. Nous entraînons alors un classifieur, initialisé aléatoirement, sur ces représentations pour résoudre le nouveau problème [46].

La stratégie 2 doit être utilisée lorsque la nouvelle collection d'images est petite et similaire aux images de pré-entraînement. En effet, entraîner le réseau sur aussi peu d'images est dangereux puisque le risque d'overfitting est important. De plus, si les nouvelles images ressemblent aux anciennes, elles peuvent alors être représentées par les mêmes features [46].

### ➤ **Stratégie 3 : Fine-tuning partiel**

Il s'agit d'un mélange des stratégies 1 et 2 : nous remplaçons à nouveau la dernière couche fully-connected par le nouveau classifieur initialisé aléatoirement, et on fixe les paramètres de certaines couches du réseau pré-entraîné. Ainsi, en plus du classifieur, on entraîne sur les nouvelles images les couches non-fixées, qui correspondent en général aux plus hautes du réseau [46].

Nous utilisons cette stratégie lorsque la nouvelle collection d'images est petite mais très différente des images du pré-entraînement. D'une part, comme il y a peu d'images d'entraînement, la stratégie 1 qui consiste à entraîner tout le réseau n'est pas envisageable à cause du risque d'overfitting [46].

D'autre part, nous éliminons également la stratégie 2 puisque les nouvelles images ont très peu de points communs avec les anciennes : utiliser les features du réseau pré-entraîné pour les représenter n'est pas une bonne idée, mais les features des couches basses sont simples et génériques (donc peuvent se retrouver dans deux images très différentes), tandis que celles des couches hautes sont complexes et spécifiques au problème. Ainsi, la stratégie de fixer les couches basses et d'entraîner le classifieur et les couches hautes constituent un bon compromis [46].

Dans cette partie nous allons voir les différents modèles pré-entraînés que nous allons utiliser dans notre travail.

### ➤ VGG16

Une version du réseau de neurones convolutif très connu appelé VGG-Net, VGG-16 est constitué de plusieurs couches, dont 13 couches de convolution et 3 *fully-connected*. Il doit donc apprendre les poids de 16 couches [47].

Il prend en entrée une image en couleurs de taille 224 \* 224 et la classifie dans une des 1000 classes. Il renvoie donc un vecteur de taille 1000, qui contient les probabilités d'appartenance à chacune des classes.

L'architecture de VGG-16 est illustrée par la Figure III-7 et III-8 :

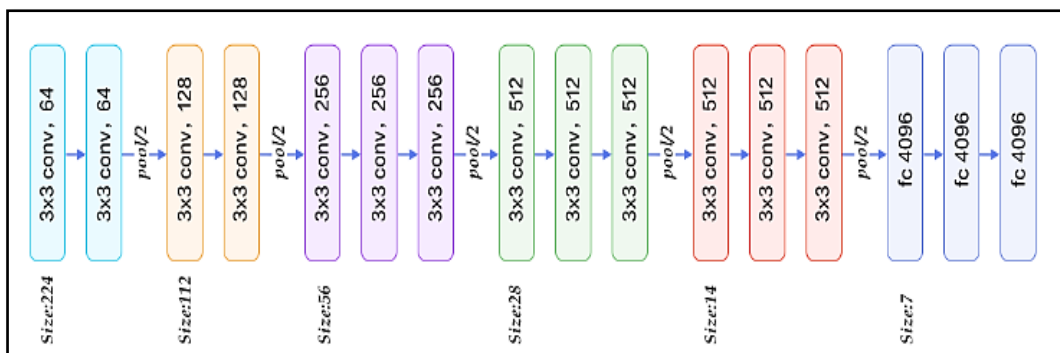


Figure III-7 : Architecture du VGG16 [47].

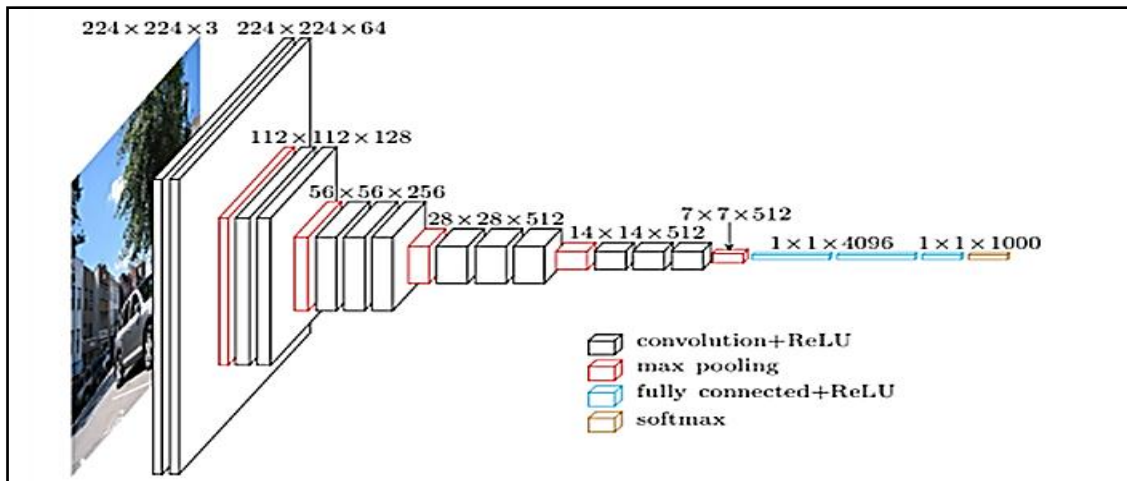


Figure III-8 : Présentation de l'architecture VGG16 en 3D [47].

Chaque couche de convolution utilise des filtres en couleurs de taille  $3 \times 3$ , déplacés avec un pas de 1 pixel. Le zero-padding vaut 1 pixel afin que les volumes en entrée aient les mêmes dimensions en sortie. Le nombre de filtres varie selon le "bloc" dans lequel la couche se trouve. De plus, un paramètre de biais est introduit dans le produit de convolution pour chaque filtre.

Chaque couche de convolution a pour fonction d'activation une ReLU. Autrement dit, il y a toujours une couche de correction ReLU après une couche de convolution. [47]

#### ➤ VGG19

VGG19 est une variante du modèle VGG qui se compose en bref de 19 couches (16 couches de convolution, 3 couches entièrement connectées, 5 couches MaxPool et 1 couche SoftMax) [48]. L'architecture du VGG19 est la suivante :

Une image RVB de taille fixe ( $224 * 224$ ) a été donnée en entrée à ce réseau, ce qui signifie que la matrice était de forme (224, 224, 3) [48].

Le seul prétraitement effectué est de soustraire la valeur RVB moyenne de chaque pixel, calculée sur l'ensemble de l'ensemble d'apprentissage [48]. Des noyaux utilisés de taille ( $3 * 3$ ) avec une taille d'un pas de 1 pixel, cela leur a permis de couvrir toute la notion d'image [48]. Un remplissage spatial a été utilisé pour préserver la résolution spatiale de l'image [48].

Un max pooling a été effectué sur une fenêtre de  $2 * 2$  pixels avec stride 2. Cela a été suivi par l'unité linéaire rectifiée (ReLU) pour introduire la non-linéarité afin de mieux classer le modèle et d'améliorer le temps de calcul, car les modèles précédents utilisaient des fonctions tanh ou sigmoïde [48].

La mise en œuvre de trois couches entièrement connectées dont les deux premières étaient de taille 4096 et après cela une couche avec 1000 canaux pour la classification ILSVRC à 1000 voies et la couche finale est une fonction softmax [48].

La Figure suivante montre l'architecture du VGG19

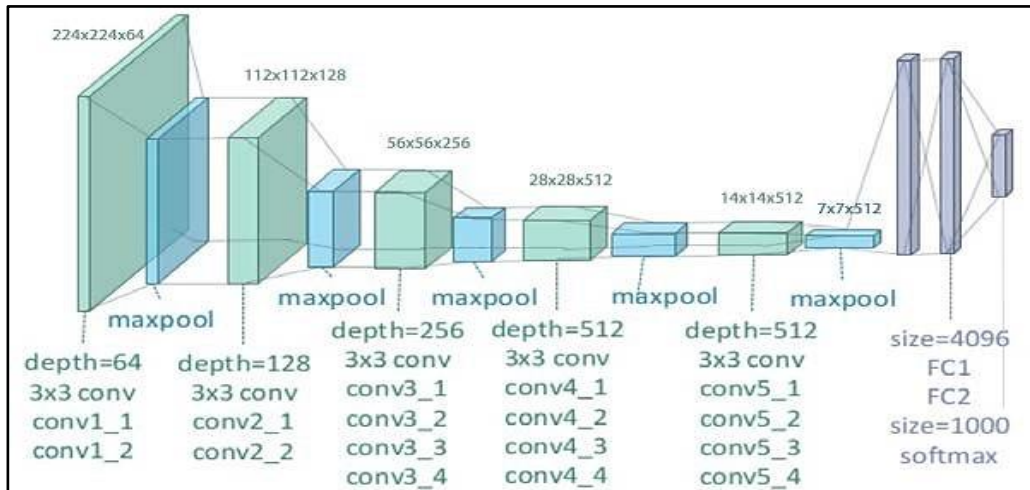


Figure III-9 : Illustration de l'Architecture VGG19 [48]

➤ Inception-ResNet-V2

Inception -Resnet V2 est un réseau neuronal convolutif entraîné sur plus d'un million d'images de la base de donnée Imagenet. Le réseau a une profondeur de 164 couches peut classer les images dans 1000 catégories d'objets, comme le clavier, la souris...etc.

Le réseau appris des présentations de caractéristiques riches et avec une taille d'entrée d'image de 299\* 299[49].

L'architecture d'Inception-ResNet-V2 est présentée dans la figure au-dessous.

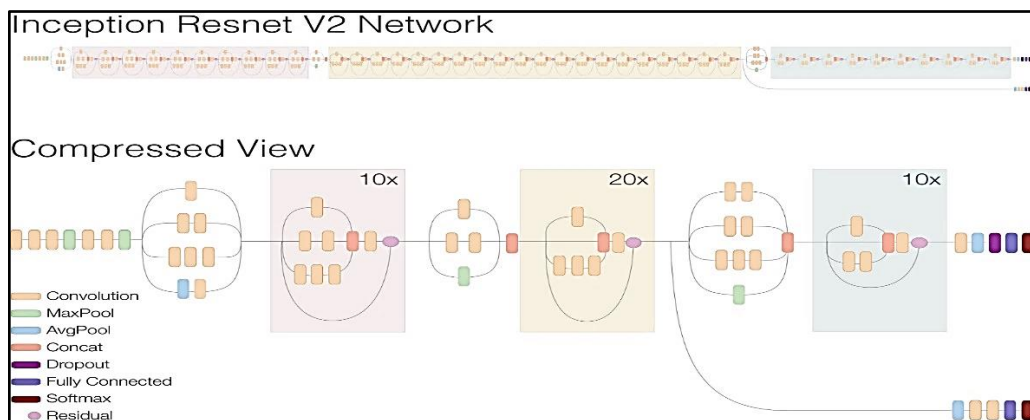


Figure III-10 : Schéma de principe d'Inception-ResNet-v2 [50].

### III.6 Paramètres d'évaluation du modèle CNN

Cette partie présente les métriques de performance couramment utilisées dans le Deep Learning et le machine Learning. L'utilisation de mesures de performance appropriées serait importante pour comparer et identifier le meilleur modèle d'apprentissage automatique pour un problème donné. Pour comprendre les métriques populaires - exactitude, précision, rappel, f1, etc., examinons d'abord la matrice de confusion [51].

#### ➤ Matrice de Confusion

Une matrice de confusion est une table (Figure III-11) qui est utilisée souvent pour décrire la performance d'un modèle de classification (ou « classificateur ») sur un ensemble de données de test. Il y a quatre entrées dans une matrice de confusion : vrai positif, faux positif, faux négatif et vrai négatif.

- Un vrai positif est un résultat où le modèle prédit correctement la classe positive.
- De même, un vrai négatif est un résultat où le modèle prédit correctement la classe négative.
- Un faux positif est un résultat où le modèle prédit de manière incorrecte la classe positive.
- Un faux négatif est un résultat où le modèle prédit de manière incorrecte la classe négative [51].

La figure suivante montre une matrice de confusion.

		True value	
		Positive	Negative
Predicted value	Positive	true positive (tp)	false positive (fp)
	Negative	false negative (fn)	true negative (tn)

Figure III-11 : Matrice de Confusion [51].

➤ **Precision, Accuracy**

- Accuracy mesure le nombre de données correctement prédites par le modèle (vrai positif + vrai négatif) sur le nombre total de données [51].

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

- La précision mesure le nombre de vrais positifs sur la somme des classes que le modèle a prédites comme positives (vrai positif + faux positif). Cela peut être considéré comme la probabilité que le modèle correctement prédit comme positif soit en fait positif [52].

$$\text{Précision} = \frac{t_p}{t_p + f_p}$$

➤ **Recall**

Le recall mesure le nombre de vrais positifs sur la somme des classes positives. Il peut être considéré comme la fraction que le modèle a correctement prédite parmi les classes positives [52].

$$\text{Recall} = \frac{t_p}{t_p + f_n}$$

La précision et le rappel ne considèrent pas le vrai négatif. Pour obtenir une haute précision, le modèle doit réduire les faux positifs (c'est-à-dire lorsque le modèle prédit à tort comme positif ce qui était en fait une classe négative). Atteindre un rappel élevé serait important dans les applications où la valeur faussement négative devrait être faible, comme le diagnostic de maladie [51].

➤ **Score F1**

Le score F1 combine précision et rappel et il est défini par la moyenne harmonique de ceux-ci. Le score F1 ne prend pas non plus en compte les faux négatifs et il est couramment utilisé dans le traitement du langage naturel, les systèmes de recherches d'informations, etc [51].

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$



➤ **ROC**

La courbe ROC (Receiver Operating Characteristic) représente la relation entre le taux de vrais positifs (axe des y) et le taux de faux positifs (axe des x), où :

$$\text{taux de vrais positifs} = \text{vrais positifs} / (\text{vrai positifs} + \text{faux négatifs})$$

$$\text{taux de faux positifs} = \text{faux positif} / (\text{faux positif} + \text{vrai négatif}).$$

Une bonne courbe ROC aurait un taux de vrais positifs relativement élevé et présenterait une courbe raide où la valeur de l'axe des x est petite [51].

### **III.7 Conclusion**

Le réseau de neurone convolutionnel est l'un des techniques de deep learning les plus utilisés, puisqu'il intègre les deux étapes de traitement d'image.

Les réseaux de neurones (CNN) ont réussi à se faire une place de choix dans des domaines tels que le traitement de l'image.

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images.

---

# **Chapitre IV**

## **Résultats d'Implémentation.**

---

## IV.1 Introduction

Dans ce chapitre nous allons exposer les différents outils de notre implémentation. D'abord, nous décrivons les bases de données utilisées puis nous donnerons un aperçu sur les environnements et les bibliothèques utilisés pour la programmation. Ensuite nous allons expliquer les différentes étapes de notre implémentation. Nous allons également parler sur les résultats obtenus et enfin nous allons vous montrer l'interface finale de notre application.

## IV.2 Base de Données utilisée

Nous avons travaillé avec 2 bases de données :

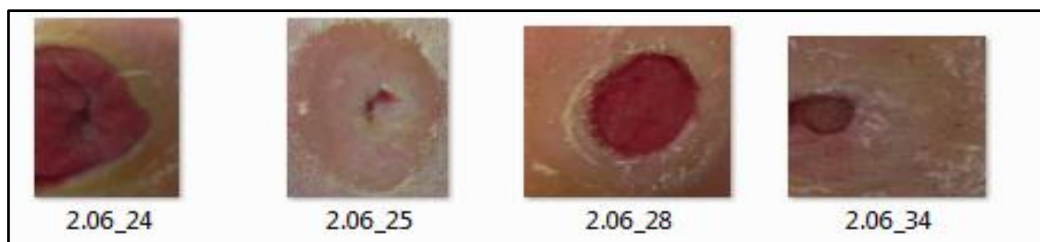
### ➤ PartA\_DFU\_dataset

Contenant 1681 images en 2D, en couleur et avec des tailles différentes. Ces dernières sont annotées par des experts. Le Tableau suivant nous montre plus d'informations sur cette base de données [30] [34] [36] [41] [51].

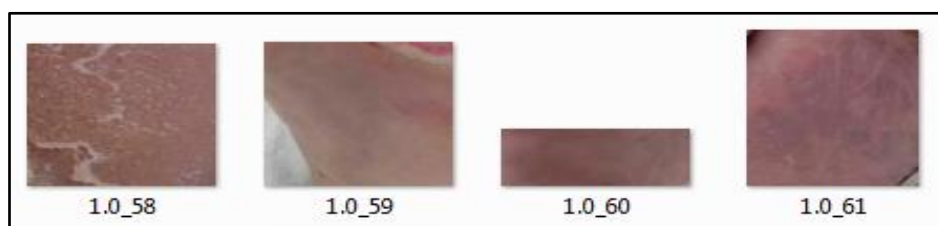
**Tableau IV-1 :** Informations sur la Base de Données PartA\_DFU.

PartA_DFU_dataset	Nombre d'Images	Format	Le contenu
Abnormal	1039	JPEG	Patches d'UPD
Normal	642	JPEG	Patches des pieds sains

Les figures suivantes représentent quelques exemples de notre base de données.



**Figure IV-1 :** Exemple des images d'ulcère du Pied Diabétique (PartA\_DFU).



**Figure IV-2 :** Exemple des images du pied sain (PartA\_DFU).

➤ **DFUC2021\_train**

Contenant un dossier d'images (Figure IV-3) en 2D, en couleur, et un fichier CSV qui contient l'annotation de chaque image comme (Figure IV-4). Le Tableau suivant nous montre plus d'informations sur cette base de données [30] [34] [36] [41] [52].

Tableau IV-2 : Informations sur la Base de Données DFUC2021

DFUC2021_train	Nombre d'images	Format	Taille	Le contenu
Images	9949	JPEG	224*224	Patches d'UPD + d'autres lésions



Figure IV-3 : Différents types d'ulcère du Pied Diabétique (DFUC2021).

	A	B	C	
1	image,none,infection,ischaemia,both			
2	301000.jpg,1,0,0,0			

Figure IV-4: Exemple d'annotation d'images sous fichier CSV (DFUC2021).

### IV.3 Environnement et Bibliothèques

Nous avons implémenté notre programme sur un PC, avec un processeur Intel® Core™ i7 CPU @ 2.40 GHz, et une mémoire vive de 8.00 Go, sous système d'exploitation Windows 10, 64 bits.

### IV.3.1 Google Colaboratory

Colaboratory, souvent raccourci en "Colab", nous permet d'écrire et d'exécuter du code Python dans notre navigateur et offre les avantages suivants [53] :

- Aucune configuration requise.
- Accès gratuit aux GPU.
- Partage facile.

Colab (Figure IV-5) permet aussi d'importer un ensemble de données d'images, d'entraîner un classificateur d'images sur cet ensemble et d'évaluer le modèle. Les notebooks Colab exécutent ce code sur les serveurs cloud de Google. Nous avons donc à notre disposition toute la puissance du matériel Google, y compris les GPU et TPU, quelle que soit la puissance de notre ordinateur.

La configuration du Notebook est montrée dans l'Annexe 1.



Figure IV-5 : Logo de Colaboratory [53].

### IV.3.2 Python

Python (Figure IV-6) est le langage de programmation open source développé par Guido Van Rossum au début des années 1980, il est caractérisé par sa polyvalence et est utilisé pour le développement web, l'IA, le Machine Learning, les systèmes d'exploitation, le développement d'applications mobiles, les jeux vidéo et bien d'autres [54].

Il joue un rôle essentiel dans notre cas « Machine Learning », en fournissant des bibliothèques et des frameworks, les plus populaires étant Scikit-Learn et TensorFlow [54].



Figure IV-6 : Logo Python [55].

### IV.3.3 TensorFlow

TensorFlow (Figure IV-7) est une plate-forme Open Source dédiée au Machine Learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du Machine Learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie [56].



Figure IV-7 : Logo de TensorFlow [56].

### IV.3.4 Keras

Keras (Figure IV- 8 ) est une API conçue pour les êtres humains, pas pour les machines. Keras suit les meilleures pratiques pour réduire la charge cognitive : il propose des API cohérentes et simples, minimise le nombre d'actions utilisateur requises pour les cas d'utilisation courants et fournit des messages d'erreur clairs et exploitables. Il contient également une documentation complète et des guides de développement [57].

Keras est le framework d'apprentissage en profondeur le plus utilisé, c'est un élément central de l'écosystème TensorFlow 2 étroitement connecté, couvrant chaque étape du flux de travail d'apprentissage automatique, de la gestion des données à la formation sur les hyper paramètres en passant par les solutions de déploiement [57].



Figure IV-8: Logo de Keras [58].

### IV.3.5 Gradio

Gradio est sorti en février 2019 en tant que bibliothèque open source en python pour créer des interfaces utilisateur faciles à utiliser, personnalisables pour tout modèle d'apprentissage automatique ou d'apprentissage profond, ou toute fonction arbitraire [57].

Elle se compose de trois paramètres :

- fn : une fonction qui exécute l'opération principale de l'interface utilisateur.
- Inputs : le type de composant d'entrée.
- Outputs : le type de composant de sortie.

## IV.4 Développement de l'Application

Afin de réaliser notre application nous avons suivi l'organigramme illustré dans la figure suivante.

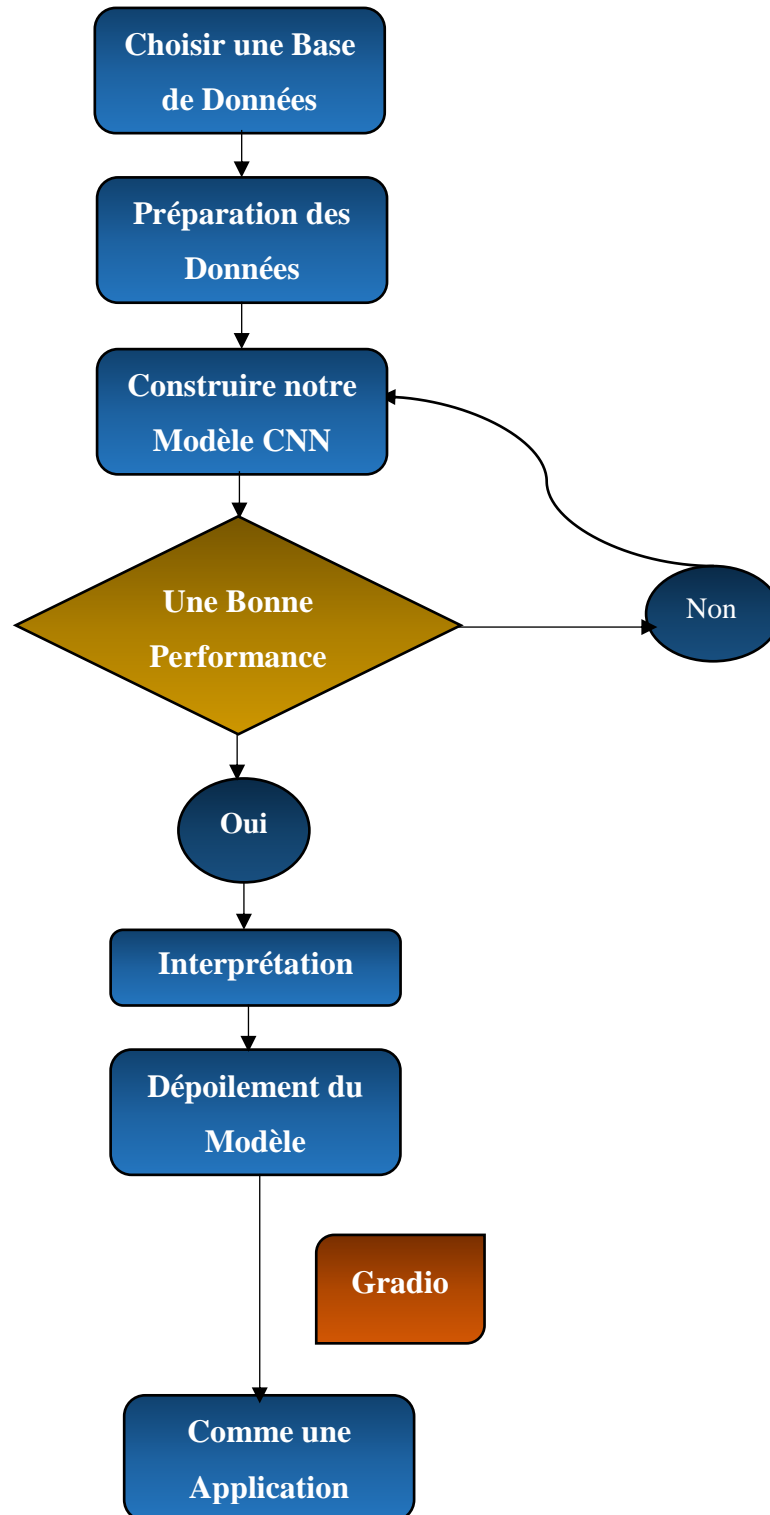


Figure IV-9 : Organigramme Général.



Nous avons donc développé deux systèmes de classification :

#### IV.4.1 Le premier Système de Classification d'UPD

Dans cette partie nous allons travailler sur une classification binaire d'UPD, ce programme va nous classer un pied sain d'un pied ulcéreux.

##### a. Programmation

Nous avons réalisé le programme en suivant les différents processus suivants :

##### Étape 1 : Importation des Bibliothèques et des couches nécessaires

Nous commençons par préparer notre environnement en important les bibliothèques et les couches nécessaires. Pour notre cas, comme nous utilisons le framework TensorFlow, nous allons importer la bibliothèque Keras ainsi que d'autres bibliothèques importantes telles que :

- Numpy et matplotlib : Outils de visualisation et de manipulation de données
- sklearn.model\_selection: Pour train\_test\_split.
- sklearn.metrics : Pour évaluer notre modèle en voyant la matrice de confusion, le F1 score, la précision, Accuracy et d'autres.
- ImageDataGenerator pour le pré-traitement des images.
- Couches spécifiques à CNN telles que Dropout, Flatten, Activation, Conv2D, MaxPooling2D, BatchNormalisation.

Le code source de cette partie est donné en annexe 2.

##### Étape 2 : Base de Données utilisée

Nous avons choisi pour cette tâche la base de données « PartA\_DFU\_Dataset », elle est composée de deux classes :

- Normal : peau saine
- Abnormal : Ulcères des pieds diabétiques.

##### Étape 3 : Préparation des Données

Maintenant que nous avons la base de données, nous allons convertir cet ensemble de données en données d'entraînement.

➤ **Définition des paramètres**

Nous avons défini 3 variables :

- DATADIR c'est le dossier où se trouvent nos images :

```
DATADIR = "/content/drive/MyDrive/pfe /PartA_DFU_Dataset"
```

- CATEGORIES qui correspond au sous dossiers (les classes) :

```
CATEGORIES = ["Abnormal", "Normal"]
```

**Remarque :** les catégories vont être numériques : 0 pour « Abnormal » et 1 pour « Normal ». L'ordre est important, nous avons commencé par « Abnormal » donc nous allons lui attribuer la valeur 0.

- IMG\_SIZE qui correspond à la taille d'image :

```
IMG_SIZE = 299
```

**Remarque :** une image de faible résolution (taille d'image\*taille d'image) ne permet pas à un réseau de convolution de voir des différences. Par contre une image de haute résolution va lui permettre de voir des différences très facilement avec une puissance de calculs énorme.

- Nous avons créé une variable training\_data dans laquelle nous avons stocké le résultat de ce que nous allons faire avec nos images :

```
training_data=[]
```

➤ **Création du Dataset**

- Nous avons indexé nos classes : 0 pour « Abnormal » et 1 pour « Normal »

```
class_num = CATEGORIES.index(category)
```

- Il est à noter que toutes les images sans exception sont parcourues dans notre Dataset.
- Ces dernières sont transformées en tableaux et stockés sous forme matricielle.
- Toutes les images sont redimensionnées en même temps.
- Une fois que nous avons fait ça, On rajoute le tout dans la variable `training_data`.
- Une fois que nous avons lancé la fonction `training-data`, l'ensemble des informations de nos images sera affiché :

```
100% |██████████| 642/642 [02:47<00:00, 3.83it/s]
100% |██████████| 1039/1039 [04:27<00:00, 3.89it/s]
```

#### ➤ Finalisation du Dataset

- Pour simplifier nous avons séparé ce que nous avons obtenu auparavant en X et Y :
  - ✓ X pour les images.
  - ✓ Y pour les labels.
- Puis le tout regroupé dans un grand tableau (les images de 0 à 255) et (labels 0 et 1), ce qui va produire une liste.
- Cette liste sera transformée en Numpy array.
- Et enfin, on stocke X et Y dans des fichiers.

#### ➤ Chargement des Données

- Pour réutiliser le Dataset que nous avons créé il suffit juste de l'importer (load).

```
(1679, 227, 227, 3)
```

- Le chargement des données a été fait avec succès (1679 images en couleur de taille (227, 227)).

#### ➤ Vérification des Données importées

La figure suivante nous montre une image de notre base de données avec l'étiquette « Normal » et de taille (227,227).

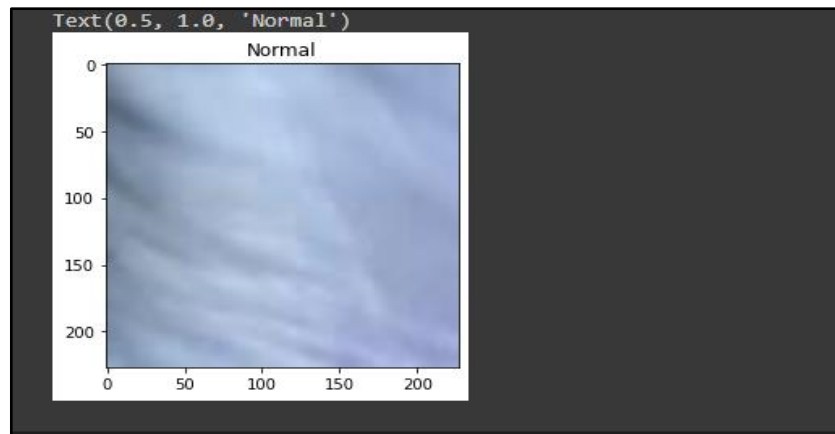


Figure IV-10 : Une image de notre base de données.

➤ **Séparation des données**

Nous avons divisé nos données en 3 parties :

- ✓ Test 30%
- ✓ Train 60 %
- ✓ Validation 10%

➤ **Augmentation des Données**

Pour une meilleure performance nous avons augmenté nos données en utilisant des fonctions telles que la rotation, la translation, le zoom et autres.

**Étape 4 : L'architecture de notre modèle CNN**

Au cours de nos expérimentations, nous avons créé deux modèles (modèle 1 et modèle 2) avec différentes architectures. Dans ce qui suit nous présentons les différents modèles.

➤ **Modèle 1 :**

Pour la classification binaire de notre base de données, nous avons utilisé l'apprentissage par transfert en utilisant un modèle pré entraîné VGG16 (l'architecture est expliquée dans le chapitre précédent).

Après les nombreux tests que nous avons faits sur notre base de données pour avoir un meilleur résultat, nous avons constaté que notre base de donnée est limitée et très différente des images du pré-entraînement et cela nous a causé un problème dans la phase d'apprentissage. Nous avons utilisé le fine-tuning partiel pour régler ce problème. Alors nous avons entraîné le

nouveau classifieur (classification binaire) et les couches haute du modèle VGG16 sauf les 5 premières couches(les plus basses).

Le Tableau IV-3 présente les différents paramètres utilisés pour entrainer notre modèle.

**Tableau IV-3** : Les Paramètres utilisés pour la Phase d'Apprentissage (Modèle 1).

Paramètres	Propriétés
Batch Size	32
Input Shape	224*224
Epoch	100
Loss	binary_crossentropy
Early stoping	20

Le Tableau IV-4 présente l'optimisateur que nous avons utilisé.

**Tableau IV-4** : L'optimisateur SGD.

Hyper Paramètres Optimisateur	Taux d'apprentissage	Momentum
SGD	0.0001	0.9

La Figure IV-11 montre les couches utilisées, leur taille de sortie ainsi que le nombre totale des paramètres entraînés.

```

Model: "sequential_5"
-----
Layer (type)                Output Shape                 Param #
-----
vgg16 (Functional)          (None, 7, 7, 512)           14714688
-----
flatten_5 (Flatten)         (None, 25088)                0
-----
dropout_5 (Dropout)         (None, 25088)                0
-----
dense_5 (Dense)              (None, 1)                    25089
-----
Total params: 14,739,777
Trainable params: 14,627,201
Non-trainable params: 112,576
    
```

**Figure IV-11** : Configuration du Modèle 1.

➤ **Architecture du Modèle 2**

Nous avons appliqué L'Apprentissage par Transfert en utilisant le modèle pré entraîné Inception-Resnet-V2. En retirant la dernière couche fully-connected et en fixant tous les paramètres et en ajoutant 2 couches fully-connected.

La première couche est composé de 256 neurones où la fonction d'activation utilisée est le ReLU. La deuxième couche utilise la fonction sigmoid qui permet de calculer la distribution de probabilité des 2 classes (nombre de classes dans la base de données PartA\_DFU\_Dataset). Le Tableau IV-5 présente les différents paramètres utilisés entrainer notre modèle.

**Tableau IV-5 :** Les Paramètres utilisées pour le Modèle 2.

Paramètres	Propriétés
Batch Size	32
Input Shape	299*299
Epoch	100
Loss	binary_crossentropy

Le Tableau IV-6 présente l'optimisateur que nous avons utilisé.

**Tableau IV-6 :** L'optimisateur Adam.

Hyper Paramètres	Taux d'apprentissage
Optimisateur	
Adam	0.001

La Figure IV-12 montre les couches utilisée, leur taille de sortie ainsi que le nombre totale des paramètres entrainés.

```

Model: "sequential_1"
-----
Layer (type)                Output Shape         Param #
-----
inception_resnet_v2 (Functio (None, 8, 8, 1536)   54336736
-----
flatten_1 (Flatten)         (None, 98304)        0
-----
dense_2 (Dense)              (None, 256)          25166080
-----
dropout_1 (Dropout)         (None, 256)          0
-----
dense_3 (Dense)              (None, 1)            257
-----
Total params: 79,503,073
Trainable params: 25,166,337
Non-trainable params: 54,336,736
    
```

**Figure IV-12 :** Configuration du Modèle 2.

**b. Résultats et interprétation**

Après avoir terminé les étapes nécessaires pour construire un système de classification, passons maintenant à l'évaluation des 2 modèles précédents :

**➤ Model 1**

Les deux graphes présentés sur la Figure IV-13 indiquent la variation des données d'apprentissage par rapports aux données de validation en fonction des époques pour le processus d'apprentissage. La précision d'entraînement commence par une valeur minimale proche de 0 et celle de validation à une valeur de 0,82. Cette différence est due à l'utilisation de la méthode d'apprentissage par transfert. Les valeurs de précision pour les données d'apprentissage et celles de validation se rapprochent. Puis ces dernières augmentent progressivement jusqu'à l'époque 34, où chacune d'elles atteint sa meilleure valeur, ceci reflète qu'à chaque époque le modèle apprend plus d'informations.

- ✓ Précision d'entraînement : 0,98 à l'époque 28.
- ✓ Précision de validation : 0,97 à l'époque 20.

L'erreur d'entraînement commence par une valeur maximale de 0,64 et celle de la validation à une valeur de 0,48. Les valeurs de l'erreur pour les données d'apprentissage et celles de validation se rapprochent. Puis ces dernières diminuent progressivement jusqu'à l'époque 34, où chacune d'elles atteint sa meilleure valeur :

- ✓ Erreur d'entraînement : 0,09 à l'époque 28.
- ✓ Erreur de validation : 0,13 à l'époque 26.

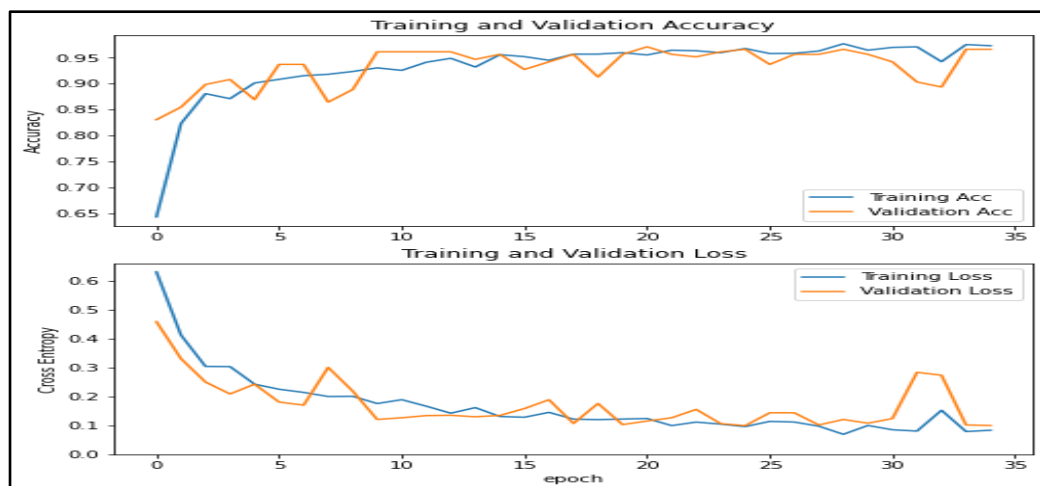


Figure IV-13 : Résultat de la phase d'apprentissage (Modèle 1).



La Figure IV-14 montre la précision et l'erreur obtenue dans chaque phase : apprentissage, test et validation.

```
[15] loss, accuracy = model.evaluate(X_test, y_test)
      loss, accuracy = model.evaluate(X_val, y_val)
      loss, accuracy = model.evaluate(X_train,y_train)

20/20 [=====] - 8s 407ms/step - loss: 0.0606 - acc: 0.9806
 7/7  [=====] - 2s 214ms/step - loss: 0.0992 - acc: 0.9660
58/58 [=====] - 14s 233ms/step - loss: 0.0525 - acc: 0.9849
```

Figure IV-14 : L'évaluation du Modèle 1.

Nous avons testé la performance de notre modèle par une image qui a été choisi aléatoirement par le système. La Figure IV-15 montre la prédiction obtenue.

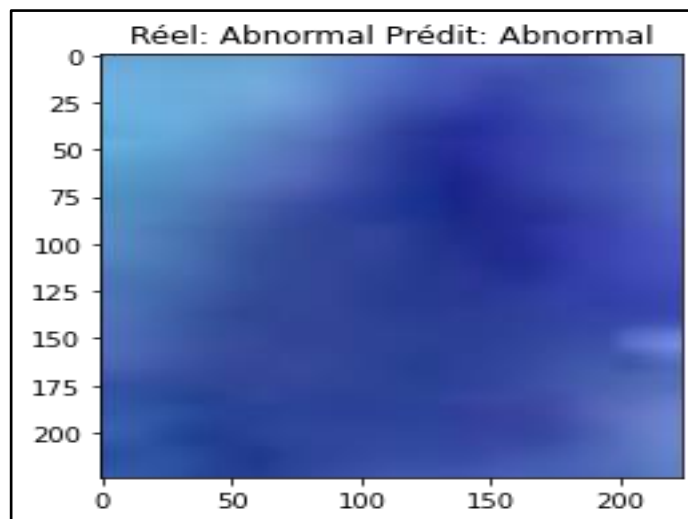


Figure IV-15 : Exemple d'un résultat de prédiction (Modèle 1).

Les Figure IV-16 et IV-17 montrent plus de paramètres pour l'évaluation (définis dans le chapitre précédent). Nous remarquons que la performance du modèle en classe « Normal » est meilleure que la classe « Abnormal »

	precision	recall	f1-score	support
Normal	0.99	0.94	0.97	331
Abnormal	0.94	0.99	0.96	286
accuracy			0.96	617
macro avg	0.96	0.97	0.96	617
weighted avg	0.97	0.96	0.96	617

Figure IV-16 : Paramètres d'évaluation (Modèle 1).

```
[[312 19]
 [ 3 283]]
```

Figure IV-17 : Matrice de confusion (Modèle 1).

Nous avons testé notre ensemble de données sur 617 images. D'après la matrice de confusion ci- dessous, nous avons :

- ✓ **Dans la classe Abnormal** : Le modèle à prédit 19 fois un résultat faux et 312 fois que la classe est correcte.
- ✓ **Dans la classe Normal** : Le modèle à prédit 3 fois un résultat faux et 283 fois que la classe est correcte.

	Abnormal	Normal
Abnormal	312	19
Normal	3	283

#### ➤ **Modèle 2**

Les deux graphes présents sur la Figure IV-18 présentent la variation des données d'apprentissage par rapports aux données de validation en fonction des époques pour le processus d'apprentissage du modèle 2.

La précision d'entraînement commence par une valeur minimale proche de 0 et celle de validation à une valeur de 0,82. Cette différence est due à l'utilisation de la méthode d'apprentissage par transfert. Les valeurs de précision pour les données d'apprentissage et celles de validation se rapprochent puis augmentent progressivement jusqu'à l'époque 34, où chacune d'eux atteint sa meilleure valeur.

- ✓ Précision d'entraînement : 0,98 à l'époque 10.
- ✓ Précision de validation : 0,97 à l'époque 12.

L'erreur d'entraînement commence par une valeur maximale de 0,64 et celle de validation à une valeur de 0,48. Les valeurs de l'erreur pour les données d'apprentissage et celles de validation se rapprochent. Puis ces dernières diminuent progressivement jusqu'à l'époque 34, où chacune d'eux atteint sa meilleure valeur :

- ✓ Erreur d'entraînement : 0,09 à l'époque 2.
- ✓ Erreur de validation : 0,13 à l'époque 12.

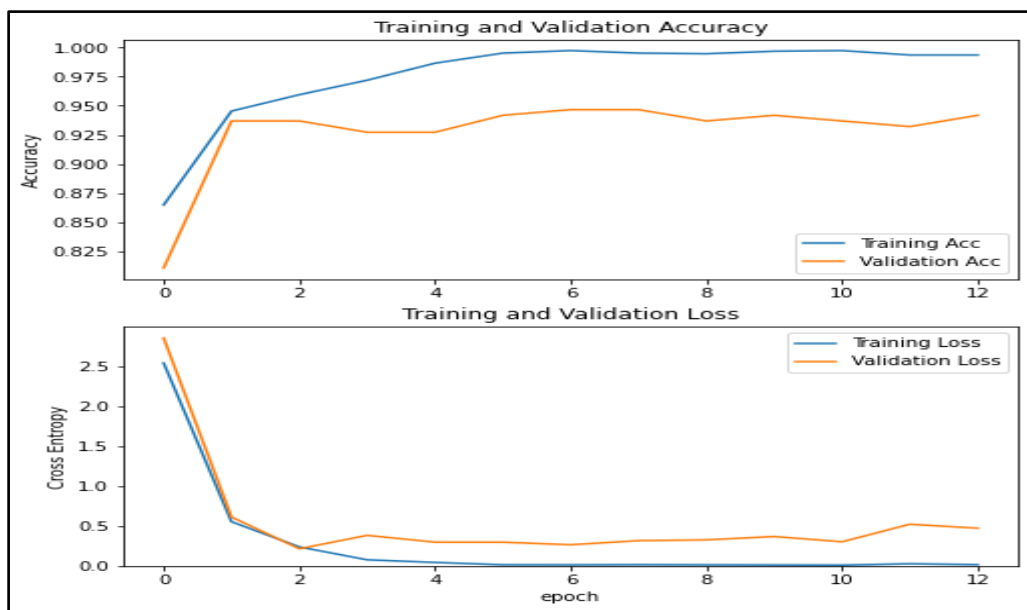


Figure IV-18 : Résultat d'apprentissage (Modèle 2).

La Figure IV-19 montre la précision et l'erreur dans chaque phase : apprentissage, test et validation.

```

1 loss, accuracy = model.evaluate(X_test, y_test)
2 loss, accuracy = model.evaluate(X_val, y_val)
3 loss, accuracy = model.evaluate(X_train,y_train)

20/20 [=====] - 16s 791ms/step - loss: 0.0438 - acc: 0.9951
7/7 [=====] - 5s 658ms/step - loss: 0.4712 - acc: 0.9417
58/58 [=====] - 42s 715ms/step - loss: 0.0013 - acc: 1.0000
    
```

Figure IV-19 : L'évaluation du Modèle 2.

Nous avons testé la performance de notre modèle par une image qui a été choisie aléatoirement par le système. La Figure IV-20 montre la prédiction obtenue.

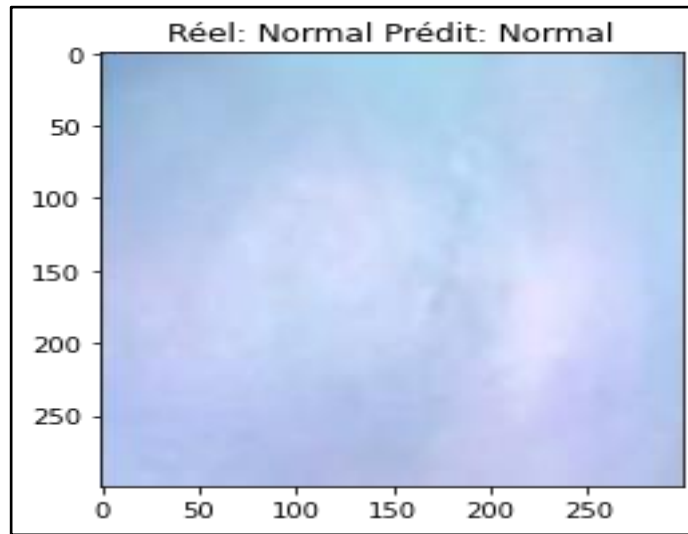


Figure IV-20 : Exemple de résultat de prédiction (Modèle 2).

Les Figures IV-21 et IV-22 montrent plus de paramètres d'évaluation. Nous remarquons que la performance du modèle en classe « Normal » est meilleure que la classe « Abnormal »

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	306
Abnormal	1.00	1.00	1.00	311
accuracy			1.00	617
macro avg	1.00	1.00	1.00	617
weighted avg	1.00	1.00	1.00	617

Figure IV-21 : Paramètres d'évaluation (Modèle 2).

```

ROC AUC: 0.998392
[[306  0]
 [ 1 310]]
    
```

Figure IV-22 : Matrice de confusion (Modèle 2).

Nous avons testé notre ensemble de données sur 617 images. D'après la matrice de confusion obtenue, nous avons :

- ✓ **Dans la classe Abnormal :** Le modèle a prédit 0 résultat faux et 306 fois que la classe est correcte.
- ✓ **Dans la classe Normal :** Le modèle à prédit 1 fois un résultat faux et 310 fois que la classe est correcte.

	Normal	Abnormal
Normal	306	0
Abnormal	1	310

**Tableau IV-7 :** Tableau comparatif des 2 Modèles réalisés.

	Base de données utilisée	Architecture utilisée	Nombre d'époch	Validation Accuracy	Test Accuracy	Loss	Temps d'exécution
<b>Modèle 1</b>	PartA_DFU_Dataset	VGG-16	34	96 %	98,06%	09,92%	16 min
<b>Modèle 2</b>	PartA_DFU_Dataset	Inception-Resnet-V2	13	94,17 %	99,51 %	21,02%	11 min

D'après le Tableau ci-dessus nous avons constaté que le modèle 2 est meilleur que le 1<sup>er</sup> modèle en termes du temps d'exécution, nombre d'époques et la précision du test.

La performance du premier système était satisfaisante, mais vu que cette application est conçue pour les médecins, ce que nous avons fait auparavant n'est pas suffisant pour deux raisons :

- Il ne définit pas les types d'UPD
- Il est impossible d'évaluer la gravité d'UPD car elle est relative à la profondeur de ce dernier et nous ne pouvons pas réaliser ça juste par une classification des images.

#### IV.4.2 Le Deuxième Système de Classification d'UPD

Nous avons pensé de développer un autre système de classification d'UPD selon la classification médicale, la plus utilisée à l'échelle internationale dans les hôpitaux, pour déterminer le risque d'amputation. C'est la Classification UT (Université du Texas).

On passe par deux étapes :

- Classifier les lignes du tableau grâce à un réseau de neurones convolutifs (pour définir le Type).
- Ajouter les grades comme des données qui seront être coché par un médecin (pour définir la Gravité).

Le système va obtenir les résultats de ces deux étapes et va nous donner le risque d'amputation comme c'est illustré sur Tableau suivant :

	<b>Grade 0</b> Phase pré ou postulcération avec épithélialisation	<b>Grade 1</b> Ulcère superficiel qui pénètre dans le derme	<b>Grade 2</b> Ulcère qui pénètre dans le derme, le tissu sous-cutané, le muscle, le tendon ou la capsule articulaire	<b>Grade 3</b> Ulcère qui pénètre jusqu'à l'os ou l'articulation
<b>Stade A</b> Pas d'infection Pas d'ischémie	<b>0A</b> 0 %	<b>1A</b> 0 %	<b>2A</b> 0 %	<b>3A</b> S/O
<b>Stade B</b> Avec infection Pas d'ischémie	<b>0B</b> 12,5 %	<b>1B</b> 8,5 %	<b>2B</b> 28,6 %	<b>3B</b> 92 %
<b>Stade C</b> Pas d'infection Avec ischémie	<b>0C</b> 25 %	<b>1C</b> 20 %	<b>2C</b> 25 %	<b>3C</b> 100 %
<b>Stade D</b> Avec infection Avec ischémie	<b>0D</b> 50 %	<b>1D</b> 50 %	<b>2D</b> 100 %	<b>3D</b> 100 %

Figure IV-23 : Classification de Texas [25].

**Partie A :**

Pour réaliser la première partie, nous avons suivi les mêmes étapes que nous avons faites dans le premier système. Le tableau suivant présente les 3 programmes que nous avons réalisés.

➤ **Bases de données utilisées**

- ✓ **base de données (DFUC2021\_train) :** nous avons séparé les images dans des sous dossiers selon les types, et vu que le nombre d'images est grand (9949 images), nous avons eu un problème de RAM donc nous avons réduit le nombre d'images (Tableau IV-8).

**Tableau IV-8 :** Distribution des images dans Base de Données utilisée (Système 2).

Les classes	Nombre d'images
None	512
Infection	520
Ischaemia	227
Both	493

- ✓ **BD3 :** Nous avons gardé les 3 classes de la base de données (PartA\_DFU\_dataset) et nous avons remplacé la classe « None » par la classe « Normal » de la base de données (DFUC2021\_train) comme le montre le Tableau suivant :

**Tableau IV-9 :** Distribution des images dans Base de Données utilisée (Système 2).

Les classes	Nombre d'images
Normal	642
Infection	520
Ischaemia	227
Both	493

➤ **Architecture utilisée**

- **Modèle 1**

Dans ce modèle nous avons utilisé la même architecture que le 1<sup>er</sup> modèle du 1<sup>er</sup> système réalisé en gardant les mêmes paramètres ainsi que l'optimisateur. Nous n'avons changé que la dernière couche *fully-connected* qui utilise la fonction softmax au lieu de sigmoid qui permet de calculer la distribution de probabilité des 4 classes (nombre de classes dans la base de données DFUC2021\_train).

La figure suivante montre les couches utilisées, leur taille de sortie ainsi que le nombre total des paramètres entraînés.

```
Model: "sequential_3"
-----
Layer (type)                Output Shape              Param #
-----
vgg16 (Functional)          (None, 7, 7, 512)        14714688
-----
flatten_3 (Flatten)         (None, 25088)            0
-----
dense_4 (Dense)             (None, 4)                100356
-----
Total params: 14,815,044
Trainable params: 14,702,468
Non-trainable params: 112,576
```

Figure IV-24 : Configuration du Modèle 1 (Système 2).

- **Modèle 2**

Pour la classification multi-classes de notre base de données, nous avons utilisé l'apprentissage par transfert en utilisant un modèle pré entraîné VGG19.

Après de nombreux tests que nous avons faits sur notre base de données pour avoir un meilleur résultat, nous avons constaté que notre base de donnée est limitée et très différente des images du pré-entraînement et cela nous a causé un problème dans la phase d'apprentissage. Nous avons utilisé le fine-tuning partiel pour régler ce problème.

Alors nous avons entraîné le nouveau classifieur (une couche fully-connected qui utilise la fonction softmax qui permet de calculer la distribution de probabilité des 4 classes (nombre de classes dans la base de données DFUC2021\_train) et les couches hautes du modèle VGG19 sauf les 7 premières couches (les plus bas).



La figure suivante montre les couches utilisées, leur taille de sortie ainsi que le nombre total des paramètres entraînés.

```

Model: "sequential_3"
-----
Layer (type)                Output Shape                Param #
-----
vgg16 (Functional)          (None, 7, 7, 512)         14714688
-----
flatten_3 (Flatten)         (None, 25088)              0
-----
dense_4 (Dense)              (None, 4)                  100356
-----
Total params: 14,815,044
Trainable params: 14,702,468
Non-trainable params: 112,576

```

Figure IV-25 : Configuration du Modèle 2 (Système 2).

- **Modèle 3**

Nous avons utilisé la même architecture que le 2<sup>ème</sup> modèle avec une autre base de données (BD3).

### c. Résultats et interprétation

Le tableau ci-dessous nous montre les différents résultats en fonction des différentes architectures utilisées et bases de données ainsi que le nombre d'époque. Les résultats obtenus sont exprimés en termes de précision de validation, de test et erreur et enfin de temps d'exécution.

Le temps d'exécution n'est pas couteux. Ceci revient à la dimension de la base de données utilisée plus l'utilisation d'apprentissage par transfert. Nous avons remarqué qu'avec les deux premières architectures utilisées la précision de la validation est plus faible que celle du troisième modèle qui est à 95,23 % et nous avons obtenu un bon résultat en test où la précision était à 99,47 %.

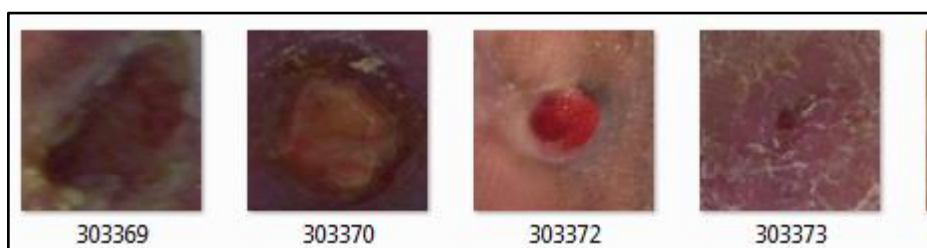
D'après ces résultats, nous avons constaté que le modèle 3 est le plus performant car nous avons utilisé une base de données différente, par contre avec les deux premiers leur

performance était un peu faible à cause de la ressemblance de leurs caractéristiques d'où pourquoi le CNN n'as pas pu faire la différence entre eux (Infection et None) et cela est dû au problème de contraste.

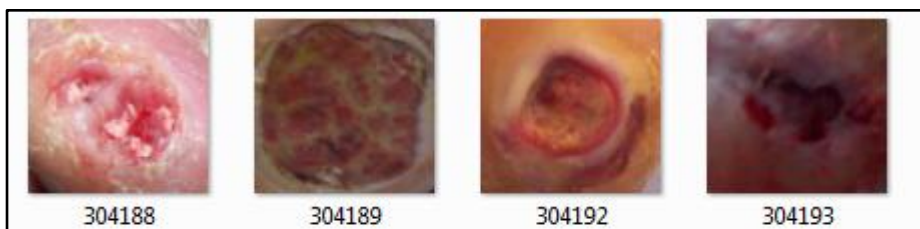
Nous allons utiliser ce modèle pour notre application. Les Figures IV-26 et IV-27 montrent quelques exemples des deux classes.

**Tableau IV-10** : Résultats obtenus des 3 Modèles utilisés.

	Base de données utilisée	Architecture utilisée	Nombre d'epoch	Validation Accuracy	Test Accuracy	Loss	Temps d'exécution
<b>Modèle 1</b>	DFUC2021_train)	VGG-16	100	83,52 %	97,91 %	57,8 %	20 min
<b>Modèle 2</b>	DFUC2021_train)	VGG-19	100	78,40 %	97,72 %	55,79%	21 min
<b>Modèle 3</b>	BD3	VGG-19	100	95,23 %	99,47 %	17,62%	23 min



**Figure IV-26** : Exemples d'images du classe « None ».



**Figure IV-27** : Exemples d'images du classe « Infection ».

#### d. Évaluation du modèle choisi

Les deux graphes de la Figure IV-28 présentent la variation des données d'apprentissage par rapports aux données de validation en fonction des époques pour le processus d'apprentissage.

La précision d'entraînement commence par une valeur minimale proche de 0 et celle de validation à une valeur de 0,65. Cette différence est due à l'utilisation de la méthode d'apprentissage par transfert. Les valeurs de précision pour les données d'apprentissage et celles de validation se rapprochent. Puis ces dernières augmentent progressivement jusqu'à l'époque 35, où chacune d'elles atteint sa meilleure valeur, ceci reflète qu'à chaque époque le modèle apprend plus d'informations.

- ✓ Précision d'entraînement : 1,00 à partir de l'époque 18.
- ✓ Précision de validation : 0,94 à l'époque 16.

L'erreur d'entraînement commence par une valeur maximale de 1,02 et celle de validation à une valeur de 0,85. Les valeurs de l'erreur pour les données d'apprentissage et celles de validation se rapprochent. Puis ces dernières diminuent progressivement jusqu'à l'époque 35, où chacune d'elles atteint sa meilleure valeur :

- ✓ Erreur d'entraînement : proche de 0 à partir de l'époque 18.
- ✓ Erreur de validation : 0,25 à l'époque 15.

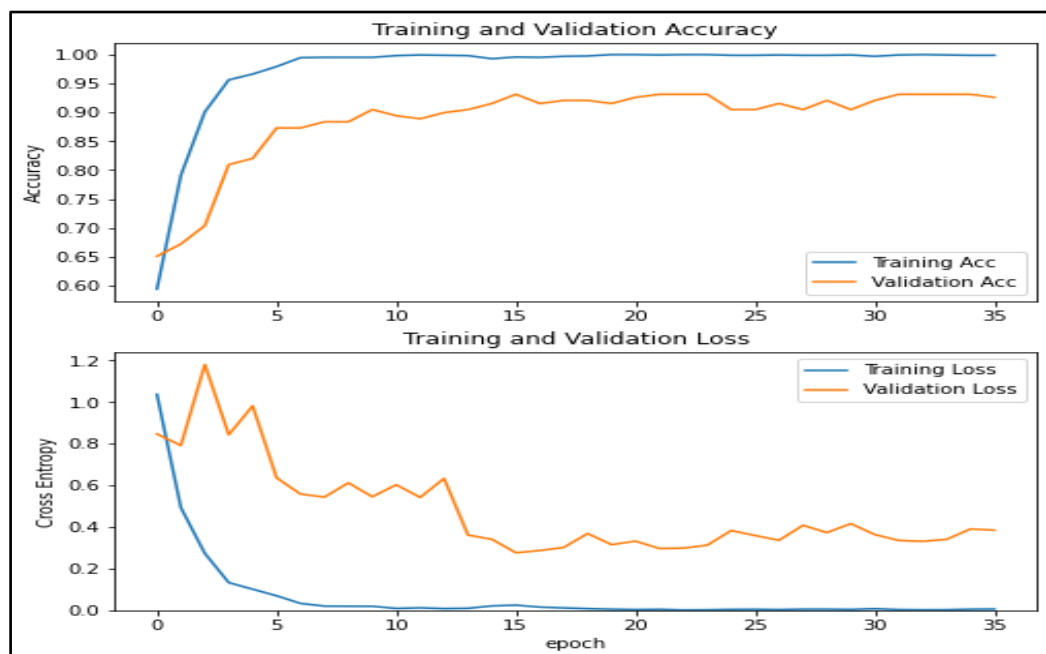


Figure IV-28 : Résultat d'apprentissage (Système 2).

Nous avons testé la performance de notre modèle par une image qui a été choisie aléatoirement par le système. La Figure IV-29 montre la prédiction obtenue.

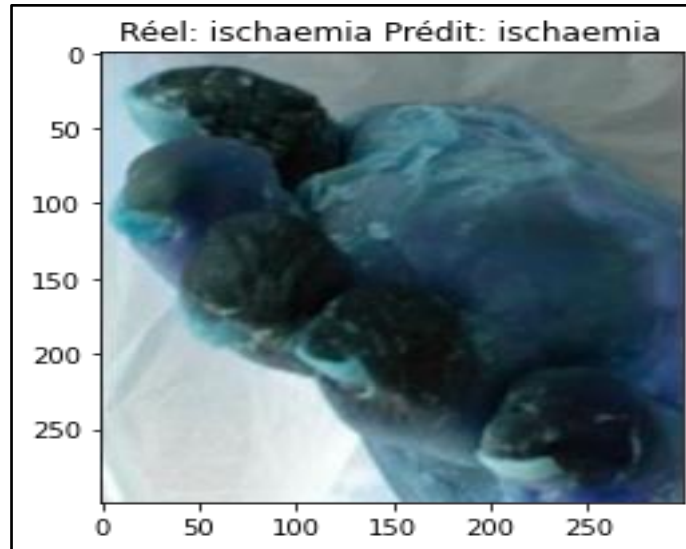


Figure IV-29 : Résultat de prédiction (Système 2).

Les Figures IV-30 et IV-31 montrent plus de paramètres pour l'évaluation. Nous remarquons que la performance du modèle en classes « Normal, Ischaemia » est meilleure que les classe « Infection, Bothl ».

```

[[214  0  0  0]
 [  0 156  0  1]
 [  0  0  51  0]
 [  0  2  0 141]]
    
```

Figure IV-30 : Matrice de Confusion (Système 2).

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	214
Infection	0.99	0.99	0.99	157
ischaemia	1.00	1.00	1.00	51
Both	0.99	0.99	0.99	143
accuracy			0.99	565
macro avg	1.00	0.99	0.99	565
weighted avg	0.99	0.99	0.99	565

Figure IV-31 : Paramètres d'évaluation (Système 2).

Nous avons testé notre ensemble de données sur 565 images. D'après la matrice de confusion obtenue, on a :

- ✓ Dans la classe « Normal » : Toutes les prédictions sont correctes.
- ✓ Dans la classe « Infection » : Une prédiction fausse.
- ✓ Dans la classe « Ischaemia » : Toutes les prédictions sont correctes.
- ✓ Dans la classe « Both » : deux prédictions fausses.

	Normal	Infection	Ischaemia	Both
Normal	214	0	0	0
Infection	0	156	0	1
Ischaemia	0	0	51	0
Both	0	2	0	141

D'après l'évaluation de notre modèle, nous constatons que ce dernier fonctionne parfaitement dans les 2 classes (Normal, Ischaemia).

**Partie B :**

Après avoir testé notre système de classification d'UPD et obtenu le résultat désiré, passons maintenant à la 2<sup>ème</sup> étape.

Les grades qui vont être cochés par le médecin seront ajoutés comme des conditions dans le programme de l'interface qui va être créée par Gradio.

### IV.4.3 Création d'une interface utilisateur

Nous allons maintenant convertir le code Deep Learning développé auparavant en une application Web grâce à Gradio pour avoir une interface souple et simple à utiliser.

L'installation de Gradio et les bibliothèques utilisées sont présentées dans l'annexe 3.

La Figure IV-32 montre l'interface de notre application Web.

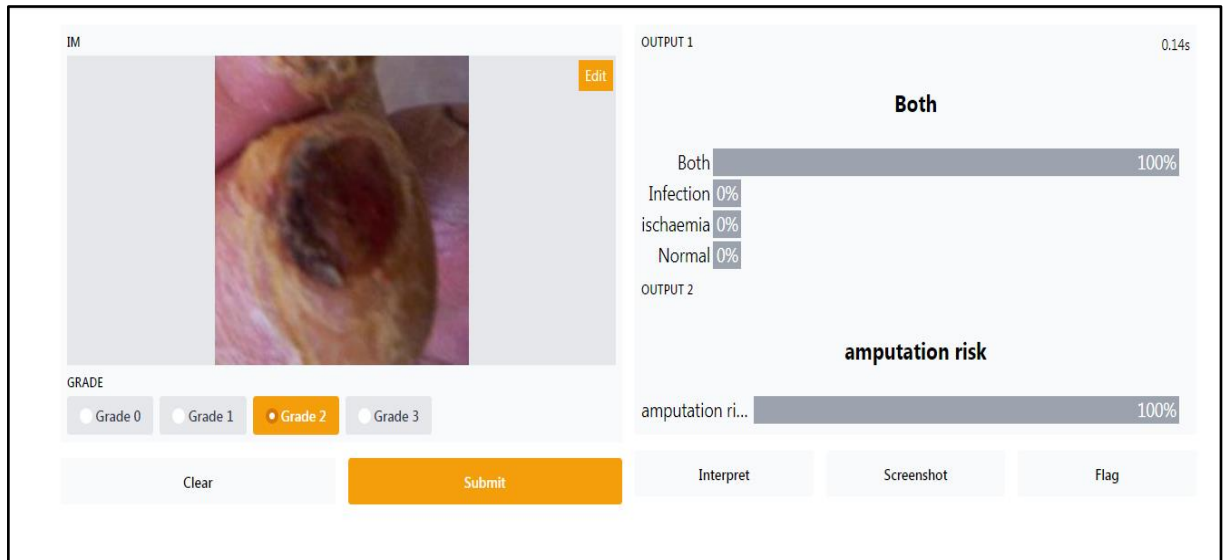


Figure IV-32 : Interface Graphique développée.

## IV.5 Conclusion

Nous avons présenté dans ce chapitre une approche de classification basée sur les réseaux de neurones convolutionnels.

Pour cela, nous avons réalisé 2 systèmes de classification d'UPD avec différentes architectures et de bases de données. Nous avons comparé ces derniers afin de choisir le plus approprié et le plus performant pour notre objectif.

La comparaison des résultats trouvés a montré que le nombre d'époque, la taille de la base et la profondeur de réseaux, sont des facteurs importants pour l'obtention de meilleurs résultats.

---

## **Conclusion Générale**

---

### Conclusion Générale

Dans le cadre de notre travail, nous avons abordé un des algorithmes de Deep Learning qui est le réseau de neurones convolutifs CNN en basant sur Le Transfert Learning (ou apprentissage par transfert) pour résoudre notre problématique.

Ce mémoire consiste donc à développer une application qui va nous permettre de faire une classification d'ulcère du pied diabétique et de graduer la sévérité de ce dernier pour aider le médecin d'évaluer le pronostic et de permettre une prise en charge adaptée à chaque type d'ulcère dans le but d'éviter l'amputation.

Nous avons fait un aperçu sur la description anatomique du pied et ses complications chez un sujet diabétique, ainsi que les différents systèmes de classifications médicales. Passant par une recherche bibliographique qui expose les différentes méthodes, architectures et techniques qui ont été proposées pour la détection et la classification de la plaie du pied diabétique, on s'est intéressé surtout aux réseaux de neurones convolutifs CNN et l'apprentissage par transfert.

Nous avons réalisé 2 systèmes de classification d'ulcère du pied diabétique, dans le 1<sup>er</sup> nous avons appliqué 2 architectures qui sont le VGG16 et InceptionResNet V2 pour classifier un pied sain d'un pied touché par l'ulcère.

Puis, nous avons amélioré notre travail en réalisant le 2<sup>ème</sup> système selon la classification médicale UT (Université du Texas) pour classifier l'ulcère, graduer la sévérité (en ajoutant des données à cocher par le médecin) et donner le risque d'amputation.

Nous avons fait une étude comparative des différentes architectures utilisées ainsi que les bases de données utilisées pour choisir le meilleur modèle afin de réaliser notre application Web. Nous avons obtenu une bonne précision de 99,47 %.



En comparant notre travail avec les précédents, nous avons conclu que nous avons obtenu des résultats très satisfaisants avec 95% de précision en validation, sachant que nous avons 4 classes (normal, infection, ischémie, both) avec un risque d'amputation au lieu de deux classes comme les travaux mentionnés précédemment où ils ont obtenu une précision de 82 %.

Comme perspectives futures, nous pouvons citer :

Le développement de notre application qui permettra de graduer la sévérité d'UPD, en changeant les données qui doivent être cochées par le médecin, par des images scanner, et d'écho Doppler.

---

# **Bibliographie**

---

### Bibliographie

- [1] A.C. Della Valle, « *Diabète : symptômes, causes, diagnostic, traitements.* », Le Journal des Femmes, 2020. Disponible sur : <https://sante.journaldesfemmes.fr/fiches-maladies/2499922-diabete-type-1-2-definition-causes-symptomes-traitement-regime-complications/> , consulté le 30/07/2021.
- [2] P. Blot, « *Maladies Endocriniennes, Diabète* », Formation Ambulancier. Disponible sur : <https://www.formationambulancier.fr/01-cours/m2/2560-mal-diabete.html>, consulté le 30/07/2021.
- [3] I. Stock, « *Medically accurate illustration of the Pancreas* », 2015. Disponible sur : <https://www.istockphoto.com/photo/medical-illustration-gm490481988-75226355>, consulté le 30/07/2021.
- [4] Organisation Mondiale de la Santé, « *Diabète* »2021.Disponible sur : <https://www.who.int/fr/news-room/fact-sheets/detail/diabetes>, consulté le 30/07/2021
- [5] « *Le Diabète* », 2018. Disponible sur : <https://www.diabete-guyane-obesite.com/le-diabete/>, consulté le 30/07/2021.
- [6] « *Diabète de type 1* », 2018. Disponible sur : <https://www.diabete-guyane-obesite.com/diabete-de-type-1/>, consulté le 30/07/2021.
- [7] « *Types 1 Diabetes Mellitus vector image* ». Disponible sur : <https://www.vectorstock.com/royalty-free-vector/types-1-diabetes-mellitus-vector-31601876>, consulté le 30/07/2021
- [8] « *Diabète de type 2* », 2018. Disponible sur : <https://www.diabete-guyane-obesite.com/diabete-de-type-2/>, consulté le 30/07/2021
- [9] « *Types 2 Diabetes Mellitus comparison Cell vector image* ». Disponible sur : <https://www.vectorstock.com/royalty-free-vector/types-2-diabetes-mellitus-comparison-cell-vector-31601870>, consulté le 30/07/2021.
- [10] « *Les Complications du Diabète* », 2018. Disponible sur : <https://www.diabete-guyane-obesite.com/les-complications-du-diabete/>, consulté le 30/07/2021.
- [11] J.Gamon, D.Rebouilla [et al.], « *Innovation Santé - Le Diabète qu'est-ce que c'est ?* », Ed. Les Webzines ECO, 2019. Disponible sur : <https://webzine.etudes-economiques.credit-agricole.com/Innovation-Sante/Diabete-nouvelles-technologies/Decryptage-1-Le-diabete-qu-est-ce-que-c-est>, consulté le 30/07/2021.
- [12] A.Djibril , E. Mossi [et al.], « *Pied diabétique : aspects épidémiologique, diagnostique, thérapeutique et évolutif à la Clinique Médico-chirurgicale du CHU Sylvanus Olympio à Lomé* », Pan African Medical Journal, 2018, vol. 30, n° 4, consulté le 19/06/2021.

- [13] M. Braiki, M.A. Khalifa [et al.], « *Introductory Chapter: Diabetic Foot* », The Eye and Foot in Diabetes, 2020, IntechOpen, DOI: 10.5772/intechopen.94331. Disponible sur : <https://www.intechopen.com/chapters/73713>, consulté le 23/04/2021.
- [14] F. Simou , « *Pied diabétique et pratiques traditionnelles* », Thèse de Doctorat, Université Cadi Ayyad, Faculté de Médecine et de Pharmacie, Marrakech, 2020. Disponible sur : <http://wd.fmpm.uca.ma/biblio/theses/annee-htm/FT/2020/these210-20.pdf>, consulté le 15/08/2021.
- [15] P. Boulier, « *Le Squelette et les Muscles du Pied* ». Disponible sur : <https://www.cabinetpodologie.fr/details-le+squelette+et+les+muscles+du+pied-91.html>, consulté le 15/08/2021.
- [16] « *Chapitre 20 : Région n° 11—Palpation des muscles intrinsèques du pied* », 2017. Disponible sur <https://clemedicine.com/20-region-n-11-palpation-des-muscles-intrinseques-du-pied/>, consulté le 15/08/2021.
- [17] B. El Allali, « *Prise en Charge Chirurgicale du Pied Diabétique* », Thèse de Doctorat, Université Mohammed V, Faculté de Médecine et de Pharmacie, Rabat, 2015. Disponible sur : <http://ao.um5s.ac.ma/xmlui/bitstream/handle/123456789/14802/M-55-2015.pdf?sequence=1&isAllowed=y>, consulté le 15/08/2021.
- [18] « *Chapitre 9 : Pied et cheville* », 2017. Disponible sur : <https://clemedicine.com/9-pied-et-cheville/>, consulté le 15/08/2021.
- [19] M. Leutenegger, D. Malgrange [et al.], « *Le Pied Diabétique* ». Disponible sur : [https://www.fmctourcoing.org/PROGRAMME\\_FMC/DIABETE\\_10\\_00/DIABETEFMI/alfediam-pied.html](https://www.fmctourcoing.org/PROGRAMME_FMC/DIABETE_10_00/DIABETEFMI/alfediam-pied.html), consulté le 23/04/2021.
- [20] C. Kembellec, « *Plaie du Pied Diabétique : étude qualitative des obstacles à l'observance de la décharge* », Thèse de Doctorat en Médecine, Université de Bordeaux, France, 2015. Disponible sur : <https://dumas.ccsd.cnrs.fr/dumas-01445924/document>, consulté le 14/05/2021.
- [21] I.A. Shaikh, N.M. Sddiqui, « *Ulcère du Pied Diabétique : une approche simple et complète* », The Eye and Foot in Diabetes, 2020, IntechOpen, DOI : 10.5772/intechopen.92585. Disponible sur : <https://www.intechopen.com/chapters/72330>, consulté le 24/04/2021.
- [22] C. Münter, P. Price [et al.], « *Ulcères du pied diabétique – prévention et traitement* », guide pratique, Coloplast, 2012. Disponible sur : [https://www.coloplast.ca/Global/Canada/WSC/PDF's%20%20FR/QuickGuide\\_Diabetic%20Foot%20Ulcer\\_FR\\_BKLET.pdf](https://www.coloplast.ca/Global/Canada/WSC/PDF's%20%20FR/QuickGuide_Diabetic%20Foot%20Ulcer_FR_BKLET.pdf), consulté le 21/04/2021.
- [23] E. Larose, « *Classification des Ulcères du Pied Diabétique.* », 2021. Disponible sur : <https://slideplayer.fr/slide/12098787/>, consulté le 02/03/2021.
- [24] L. Orioli, « *La Prise en Charge du Pied Diabétique : de la nécessité d'une équipe pluridisciplinaire.* », Louvain Médical. Disponible sur :

<https://www.louvainmedical.be/sites/default/files/content/article/pdf/lmed-mars-2017-08-orioli.pdf>, consulté le 16/03/2021.

[25] D. St-Cyr, L. Martineau, « *L'ulcère du Pied Diabétique Partie 1* », Formation continue, Ordre des Infirmeries et Infirmiers du Québec, 2017, vol. 14 / n° 1. Disponible sur : <https://www.oiiq.org/sites/default/files/uploads/periodiques/Perspective/vol14no01/09-pratique-clinique-acfa.pdf>, consulté le 14/03/2021.

[26] Liu W, Anguelov D, Erhan D, [et all.], « *Single shot multibox detector* », 2015, consulté le 30/03/2021.

[27] L. Wang [et all.], « *Area Determination of Diabetic Foot Ulcer Images Using a Cascaded Two-Stage SVM Based Classification* », Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, USA, 2016, consulté le 30/03/2021.

[28] R. Brown [et all.], « *MyFootCare: a mobile self-tracking tool to promote self-care amongst people with diabetic foot ulcers* », *Proceedings of the 29th Australian Conference on Computer-Human Interaction*, <https://dl.acm.org/doi/10.1145/3152771.3156158>, 2017, consulté le 30/03/2021.

[29] S. Patel [et all.], « *Diabetic Foot Ulcer Wound Tissue Detection and Classification* », International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS), 2017, consulté le 05/04/2021.

[30] M. Goyal [et al.], « *Fully Convolutional Networks for Diabetic Foot Ulcer Segmentation* », International Conference on Systems, Man and Cybernetics (SMC), UK, 2017, consulté le 05/04/2021.

[31] M. Goyal [et all.], « *DFUNet : Convolutional Neural Networks for Diabetic Foot Ulcer Classification* », Centre for Advanced Computational Science, School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, UK, 2018, consulté le 05/04/2021.

[32] K.S. Babu, [et all.], « *Efficient Detection and Classification of Diabetic Foot Ulcer Tissue using PSO Technique* », International Journal of Engineering & Technology, vol. 7, n°3, Inde, 2018, consulté le 05/04/2021.

[33] D. Alejandro [et all.], « *Plantar Thermogram Database for the Study of Diabetic Foot Complications* », Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, Mexico, 2019, consulté le 05/04/2021.

[34] M. Goyal [et all.], « *Robust Methods for Real-Time Diabetic Foot Ulcer Detection and Localization on Mobile Devices* », IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, VOL. 23, n° 4, 2019. Disponible sur : [https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8456504&fbclid=IwAR1zCmcgvCKewXvTWCrm8s7t8ddQLHhBAkWdpJDubTFuWr1A4wHO8X\\_H71U](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8456504&fbclid=IwAR1zCmcgvCKewXvTWCrm8s7t8ddQLHhBAkWdpJDubTFuWr1A4wHO8X_H71U), consulté le 24/04/2021.

- [35] L. Alzubaidi [et all.], « *DFU\_QUTNet: diabetic foot ulcer classification using novel deep convolutional neural network* », Multimedia Tools and Applications, 2019. Disponible sur: <https://doi.org/10.1007/s11042-019-07820-w>, consulté le 17/04/2021.
- [36] M. Goyal [et all.], « *Recognition of Ischaemia and Infection in Diabetic Foot Ulcers: Dataset and Techniques* », Centre for Advanced Computational Sciences, Manchester Metropolitan University, Manchester, UK. Disponible sur: <https://arxiv.org/pdf/1908.05317.pdf>, 2020, consulté le 17/04/2021.
- [37] N. Rania [et all.], « *Semantic Segmentation of Diabetic Foot Ulcer Images: Dealing with Small Dataset in DL Approaches* », PRISME Laboratory, Orléans University, France, 2020, consulté le 17/04/2021.
- [38] J.AMIN [et all.], « *An Integrated Design for Classification and Localization of Diabetic Foot Ulcer Based on CNN and YOLOv2-DFU Models* ». Department of Computer Science, University of Wah, Pakistan, 2020, Consulté le 22/04/2021.
- [39] D. YT Chino [et all.], « *Segmentation des ulcères cutanés et mesure de la zone de la plaie à l'aide de réseaux convolutifs profonds* », Institut des sciences mathématiques et informatiques, Université de Sao Paulo, Brésil, 2020. Consulté le 22/04/2021.
- [40] L. Alzubaidi [et all.], « *Novel Transfer Learning Approach for Medical Imaging with Limited Labeled Data* », Faculty of Electrical Engineering & Computer Science, University of Missouri, Columbia, USA, 2021. Disponible sur : <https://github.com/muthanak/Novel-Transfer-Learning-for-Medical-image-classification>, consulté le 22/04/2021.
- [41] Y. Moi Hoon [et all.], « *Analysis Towards Classification of Infection and Ischaemia of Diabetic Foot Ulcers* », Faculty of Science and Engineering, Manchester Metropolitan University, Manchester, UK, 2021, consulté le 22/04/2021.
- [42] H. Chettouh, « *Montage d'un Système de Reconnaissance des Expressions Faciales avec le Deep Learning* », Mémoire de Master en Informatique, Université Mohamed Khider – Biskra, Algérie, 2020, consulté le 16/05/2021.
- [43] P. Monasse, K. Nadjahi, « *Qu'est-ce qu'un Réseau de Neurones Convolutif (ou CNN)?* », Openclassrooms, 2021. Disponible sur : <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn>, consulté le 10/05/2021.
- [44] « *Convolutional Neural Network* », 2020. Disponible sur : <https://datascientest.com/convolutional-neural-network>, consulté le 10/05/2021.
- [45] P. Monasse, K. Nadjahi, « *Découvrez les différentes couches d'un CNN* », 2021. Disponible sur : <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5088816-apprenez-a-construire-un-cnn-et-gagnez-du-temps-avec-le-transfer-learning>, consulté le 06/05/2021.

- [46] P. Monasse, K. Nadjahi , « *Apprenez à construire un CNN et gagnez du temps avec le Transfer Learning* », 2021. Disponible sur : <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5088816-apprenez-a-construire-un-cnn-et-gagnez-du-temps-avec-le-transfer-learning>, consulté le 06/05/2021.
- [47] « *VGG16 architecture* », Computing Expertise & Legacy. Disponible sur : <https://iq.opengenus.org/vgg16/>, consulté le 12/07/2021.
- [48] « *Understanding the VGG19 Architecture* ».Disponible sur: <https://iq.opengenus.org/vgg19-architecture/>, consulté le 12/08/2021.
- [49] « *inceptionresnetv2* », HelpCenter,2021. Disponible sur : <https://www.mathworks.com/help/deeplearning/ref/inceptionresnetv2.html;jsessionid=c8da9caa455398c0f2b773f169a4>, consulté le 06/07/2021.
- [50] A. Alemi, « *Improving Inception and Image Classification in TensorFlow* », Google AI Blog,2016. Disponible sur : <https://ai.googleblog.com/2016/08/improving-inception-and-image.html>, consulté le 06/07/2021.
- [51] Medium, « *Intro to Deep Learning — performance metrics(Precision, Recall, F1, ROC, PR, PRG)* »,2021.Disponible sur: <https://hk3342.medium.com/intro-to-deep-learning-performance-metrics-precision-recall-f1-roc-pr-prg-87f5073f9354>, consulté le 12/08/2021.
- [52] B.Cassidy,« *Dfuc2020: Analysis towards diabetic foot ulcer detection*»? DOI/10.17925/EE.2021.1.1.1.5,2020.
- [53] « *Qu'est-ce que Colaboratory ?* ». Disponible sur : <https://colab.research.google.com/notebooks/intro.ipynb?hl=fr>, consulté le 01/10/2021.
- [54] « *Qu'est-ce que le Python et quelle est son utilisation ? Conseils, Emploi & Tech* », 2020. Disponible sur : <https://mobiskill.fr/blog/conseils-emploi-tech/quest-ce-que-le-python-et-quelle-est-son-utilisation/>, consulté le 01/10/2021.
- [55] « *Python -Logo-Png-Image* », 2017. Disponible sur : <https://blog.savoirfairelinux.com/fr-ca/2017/revue-de-presse-inno-7/python-logo-png-image/>, consulté le 01/10/2021.
- [56] « *Why TensorFlow* ». Disponible sur : <https://www.tensorflow.org/>, consulté le 01/10/2021.
- [57] « *Keras* », Disponible sur : <https://keras.io/> , consulté le 01/10/2012.
- [58] S. Sanya, « *Build User Interface with GRADIO for your Deep Learning Project!* », Analytics Vidhya, 2021. Disponible sur : <https://www.analyticsvidhya.com/blog/2021/06/build-user-interface-with-gradio-for-your-deep-learning-project/>, consulté le 03/10/2021.

---

# **Annexes**

---



## Annexe 1 : Le Code source des Bibliothèques utilisées

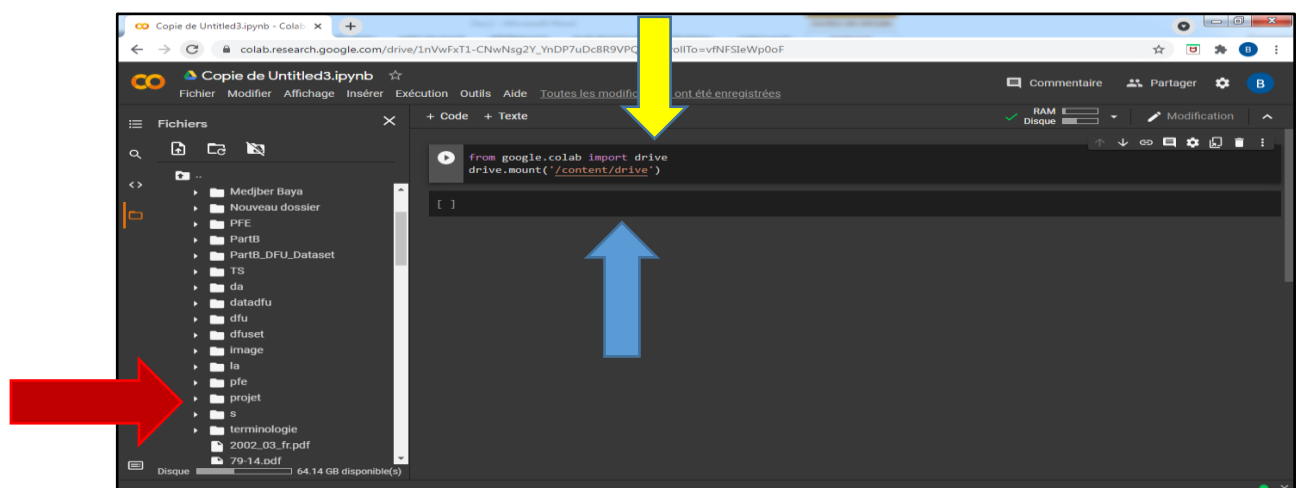
```
import matplotlib.pyplot as plt
import os
import cv2
import numpy as np

import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.callbacks import TensorBoard
from keras.callbacks import EarlyStopping, ModelCheckpoint
import tensorflow.keras as keras
from keras.preprocessing.image import img_to_array, load_img
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import roc_auc_score
```

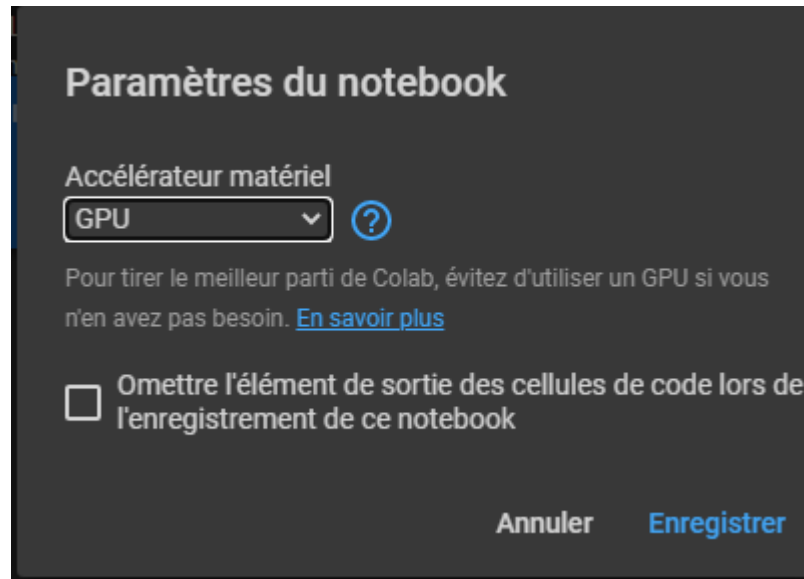
## Annexe 2 : Configuration du Notebook de Google Colab.

La figure suivante montre la configuration du Notebook :

- Flèche bleue : Conçu pour écrire notre code
- *Fleche* jaune : Ce code source va donner l'autorisation à ce notebook pour accéder à nos fichiers Google Drive.
- Flèche rouge : Montre les fichiers du Google Drive après la connexion.



- Avant de commencer l'exécution nous choisissons GPU comme Accélérateur matériel comme le montre la figure suivante :



## Annexe 3 : L'installation de Gradio

```
] !pip install gradio
Requirement already satisfied: gradio in /usr/local/lib/python3.7/dist-packages (2.3.6)
Requirement already satisfied: flask-cachebuster in /usr/local/lib/python3.7/dist-packages (from gradio) (1.0.0)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from gradio) (7.1.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from gradio) (3.2.2)
Requirement already satisfied: pycryptodome in /usr/local/lib/python3.7/dist-packages (from gradio) (3.10.4)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from gradio) (1.19.5)
Requirement already satisfied: analytics-python in /usr/local/lib/python3.7/dist-packages (from gradio) (1.4.0)
Requirement already satisfied: Flask-Cors>=3.0.8 in /usr/local/lib/python3.7/dist-packages (from gradio) (3.0.10)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from gradio) (1.1.5)
Requirement already satisfied: ffmpeg in /usr/local/lib/python3.7/dist-packages (from gradio) (0.3.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from gradio) (1.4.1)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from gradio) (2.23.0)
Requirement already satisfied: paramiko in /usr/local/lib/python3.7/dist-packages (from gradio) (2.7.2)
Requirement already satisfied: Flask-Login in /usr/local/lib/python3.7/dist-packages (from gradio) (0.5.0)
Requirement already satisfied: Flask 1.1.1 in /usr/local/lib/python3.7/dist-packages (from gradio) (1.1.1)
```

```
import gradio as gr
import requests
```

## Annexe 4 : L'interface Finale

IM

Drop Image Here  
- or -  
Click to Upload

GRADE

Grade 0  Grade 1  Grade 2  Grade 3

Interpret Screenshot Flag

Clear Submit