الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث 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

Filière Électronique
Spécialité électronique des systèmes embarqués

Présenté par

DROUICHE Ilhem

&

BELLAOUCHA Noufel Mehdi

Développement d'un modèle U-Net amélioré pour la segmentation des micro-anévrismes rétiniens, et d'un dispositif optique expérimental de photocoagulation laser multi-spots par DMD

Proposé par : Mme. BOUGHERIRA Hamida

Année Universitaire 2024-2025

Le numéro du PFE

ESE20

Remercîments

C’est avec grand plaisir que nous consacrons cette page aux remerciements à tous ceux qui ont contribué à la réalisation de ce travail.

Nous remercions tout d’abord notre encadrante Mme BOUGHERIRA HAMIDA pour sa patience, ses précieux conseils, son soutien moral, son suivi, ses orientations continues, et pour nous avoir toujours poussés vers le meilleur.

Nous exprimons aussi notre profonde gratitude aux membres du jury, M. KABIR et M. NAMANE, pour avoir accepté d’évaluer ce travail et pour leurs remarques constructives qui ont permis d’enrichir notre mémoire.

Nous remercions également la responsable de spécialité Mme NACEUR DJAMILA pour ses encouragements et ses grands efforts en faveur de la réussite de notre formation en Master Électronique et Systèmes Embarqués.

Nous n’oublions pas de remercier aussi la docteure LAMIA DAHAM qui nous a consacré de son temps pour répondre à nos questions et nous aider à créer une base de données à partir de zéro.

Enfin, nous remercions toute personne ayant contribué, de près ou de loin, à la réalisation de ce projet.

Dédicaces

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non âmes exigences et qui n'a épargné aucun effort pour me rendre heureuse : ma mère abordée, Hayet.

A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect : mon père, Omar.

A mon adorable petite sœur et mon frère, qui n'ont pas cessée de me conseiller, encourager et soutenir tout au long de mes étude. Que dieu les protège et leurs accorde chance et bonheur.

A mes amis et collègues puisse ce travail vous exprime mes souhaits de succès, et mes sincère sentiments envers vous.

Sans oublier mon binôme Ilhem pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

Noufel

Dédicaces

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

Je tien à dédie cet humble travail à :

A ma tendre mère et mon très cher père, mais aucune dédicace ne serait témoin de mon profond amour, mon immense gratitude et mon plus grand respect, car je ne pourrais jamais oublier ma tendresse et l'amour dévoué par lesquels ils m'ont toujours entouré depuis mon enfance.

A mes chères sœurs et mon frère, pour leur présence réconfortante et leurs mots simples mais si puissants, qui m'ont souvent redonnée.

Je tiens à adresser une pensée particulière à mon binôme Noufel, pour son sérieux, sa complicité et sa persévérance tout au long de cette aventure, travailler ensemble a été un vrai plaisir et ce mémoire est autant le lien que le mien.

Enfin, a toutes les personnes qui, de près ou de loin, ont contribué à l'aboutissement de ce travail : vos conseils, votre inspiration et votre confiance ont été précieux.

Merci à tous.

Ilhem

ملخص

تم تطوير نموذج U-Net المعزز من الشبكات العصبية الالتفافية (CNN) لاكتشاف الميكروأنيوريزمات (MA) في صور قاع العين. قمنا بتطبيق التعلم بالنقل (Transfer Learning) من خلال استبدال جزء التشفير الخاص بـ U-Net بنموذج VGG-16(visual geometry group) مدرب مسبقًا، وذلك لتقليل وقت التدريب. تم تنفيذ التعلم على قاعدة بيانات خضعت للوسم، والمعالجة المسبقة، وزيادة البيانات لتحسين أداء النموذج، مع الأخذ بعين الاعتبار خصائص الميكروأنيوريزمات من حيث الحجم (تقسيم الصور إلى 16 جزءًا مع تطبيق الزوم)، والشكل، واللون (إزالة القناة الحمراء لزيادة التباين). تم إنشاء قاعدة البيانات ووسمها داخل عيادة طبية بمساعدة طبيب عيون، وكانت تتضمن 141 صورة لقاع العين. أظهرت النتائج دقة عالية جدًا في اكتشاف المناطق المصابة، سواء التي تم اكتشافها أو التي لم يتم اكتشافها من قبل طبيب العيون. بعد ذلك، تم إرسال الصورة الثنائية (المجزأة) إلى جهاز DMD(Digital Micromirror Device) عبر نظام بصري تجريبي قمنا بتصميمه، بهدف التحكم بدقة في تموضع شعاع الليزر المعدل بواسطة المرايا الدقيقة الخاصة بـ DMD، لإثبات إمكانية تنفيذ العلاج الضوئي بالليزر (Photocoagulation) متعدد النقاط.

الكلمات المفتاحية

اعتلال الشبكية السكري(DR) ، التوسع الدقيق(MA) ، التعلم العميق، الشبكة العصبية التلافيفية(CNN) ، U-Net، تقسيم الصور، الشبكية، ليزر، جهاز المرايا الدقيقة الرقمي.

Résumé

Un modèle U-Net amélioré de réseaux de neurones convolutifs a été développé pour la détection des micro-anévrismes (MA) dans les images de fond d'œil. Nous avons appliqué le transfer learning en remplaçant la partie encodeur de U-Net par un modèle VGG-16 (visual geometry group) pré-entrainé, pour réduire le temps d'apprentissage. L'apprentissage a été effectué sur une base de données ayant subi un étiquetage, un prétraitement et une augmentation de données pour améliorer les performances du modèle, en tenant compte des particularités des micro-anévrismes en termes de taille (division des images par 16, et zoom), de forme, et de couleur (suppression du canal rouge pour augmenter le contraste). La base de données a été constituée et étiquetée dans un cabinet médical, avec l'aide d'un ophtalmologue, et comprenait 141 images de fond d'œil. Les résultats obtenus ont montré une très grande précision dans la détection des zones atteintes, détectées et d'autres, non détectées, par l'ophtalmologue. L'image binaire segmentée a ensuite été envoyée à un DMD (Digital Micromirror Device), à travers un dispositif optique expérimental que nous avons conçu, pour contrôler avec précision le positionnement d'un faisceau laser modulé par les micromiroirs du DMD, afin de démontrer la faisabilité d'une photocoagulation multi-spots.

Mots clés

Rétinopathie diabétique (DR), Micro-anévrisme (MA), Apprentissage profond, Réseau neuronal convolutif (CNN), U-Net, Segmentation d'image, Rétine, laser, dispositif a micro-miroirs numerique (DMD).

Abstract

An improved U-Net model based on convolutional neural networks (CNN) was developed for the detection of microaneurysms (MA) in fundus images. Transfer learning was applied by replacing the encoder part of U-Net with a pre-trained VGG 16 (visual geometry group) model to reduce training time. The training was performed on a labeled dataset that underwent preprocessing, and data augmentation to improve model's performance, taking into account the specific characteristics of microaneurysms in terms of size (image division by 16 and zoom), shape, and color (removal of the red channel to enhance contrast). The dataset was created and labeled in a medical clinic with the assistance of an ophthalmologist and consisted of 141 fundus images. The results showed very high accuracy in detecting the affected areas, including some that were missed or not detected by the ophthalmologist. The segmented binary image was then sent to a DMD (Digital Micromirror Device) through an experimental optical setup we designed, allowing precise control of a laser beam modulated by the DMD's micromirrors, to demonstrate the feasibility of multi-spot photocoagulation treatment.

Keywords

Diabetic retinopathy (DR), Microaneurysm (MA), Deep learning, Convolutional neural network (CNN), U-Net, Image segmentation, Retina, laser, Digital micromirror device (DMD).

Table des matières

Liste des figures

Liste des tableaux

Liste des acronymes et abréviations

ADAM	Estimation adaptative des moments (adaptive moment estimation)
CNN	Réseau de neurones convolutifs (Convolutional Neural Network)
Conv	Convolution
DMD	Dispositif à micro-miroirs numerique (Digital Micromirror Device)
GPU	Unité de traitement graphique (Graphics Processing Unit)
JPG	Groupe d’experts en photographie (joint phtographic experts group)
MA	Micro-anévrismes (micro-aneurysms)
MLP	Perceptron multicouche (multi-layer perceptron)
PNG	Graphiques portables en reseau (portable network graphics)
RD	La rétinopathie diabétique (diabetic retinopathy)
ReLu	Unité linéaire rectifiée (rectified linear unit)
RNA	Un réseau de neurones artificiels
U-Net	Réseau en U (U-shaped network)
VAL	Validation
VGG	Visual geometry group
IA	Intelligence artificielle

Introduction Générale

Introduction générale

La segmentation automatique du fond d'œil est devenue une étape essentielle dans l'analyse et le diagnostic de la rétinopathie diabétique (Diabetic Retinopathy - DR). De nombreuses techniques d'intelligence artificielle ont été proposées pour segmenter ces images.

Grâce aux avancées rapides dans le domaine de l'intelligence artificielle, il est désormais possible d'analyser les données médicales de manière plus approfondie et plus efficace. Les algorithmes de vision par ordinateur connaissent une adoption croissante dans les systèmes d'imagerie médicale à travers le monde.

Les études ont montré que l'utilisation de réseaux de neurones profonds (Deep CNN) permet d'extraire des caractéristiques importantes à partir des images du fond d'œil, avec un niveau de performance comparable à celui des médecins spécialistes.

Dans ce projet, nous développons un système d'apprentissage profond (Deep Learning) basé sur ces réseaux neuronaux, en utilisant la plateforme Colab et le langage de programmation Python avec ses bibliothèques, notamment Keras et TensorFlow, ainsi que plusieurs autres outils. Nous appliquons le modèle U-Net pour détecter et localiser précisément les micro-anévrismes (Microaneurysms - MA) liés à la rétinopathie diabétique.

Pour optimiser l’efficacité de l’apprentissage nous utilisons le transfert Learning en combinant le modèle VGG-16 avec U-Net

La base d'apprentissage utilisée pour l'entraînement et les tests est un ensemble d'images du fond d'œil présentant des micro-anévrismes causés par la rétinopathie diabétique. Nous les avons créés et étiquetées nous-même avec le concours et le contrôle d'une ophtalmologiste.

Afin de booster les performances de notre architecture U-Net, nous lui avons soumis lors de son apprentissage une base de données qui tient compte des caractéristiques des micro-anévrismes.

- À cause de leur petite taille nous avons subdivisé les images de fond d'œil en 16 et zoomé sur chacune des 16 sections.
- À cause de la couleur des micro-anévrismes nous avons supprimé le canal rouge pour augmenter le contraste

Après leur traitement par notre système de reconnaissance, les zones affectées sont détectées et mises en évidence à l'aide de U-Net.

Une fois la segmentation des micro-anévrismes par U-Net effectuée, nous procédons à une simulation réaliste du traitement des micro-anévrismes au laser, en nous basant sur les prédictions générées par le modèle U-Net.

L'objectif de notre travail est la segmentation des images de fond d'œil pour la mise en évidence des zones des micro-anévrismes à traiter par laser dans une image N/B (zones blanches = micro-anévrismes).

Nous utilisons la capacité des micromiroirs à moduler un faisceau laser en un motif de rayons laser correspondant à une image noir et blanc affiché par le DMD (DLP7000).

Quand l'image segmentée par U-Net est envoyée au DMD, celui-ci réfléchit une partie du faisceau laser incident vers la cible (la rétine).

Nous concevons un dispositif optique et réalisons une expérience à base de ces micromiroirs afin de démontrer qu'une segmentation correcte, générée par U-Net, permet un ciblage précis et simultané, de tous les points d'impact, évitant aux patients inconfort et incidents graves.

Cette recherche est structurée en quatre chapitres :

Le chapitre 1 définit la rétinopathie diabétique, les pathologies et donne les notions d'imagerie médicale.

Dans le chapitre 2 nous présentons un état de l'art des Réseaux de neurones profonds et méthodes de segmentation d'images médicales.

Notre travail est décrit dans le chapitre 3. Notre contribution à la détection de la rétinopathie diabétique et sa segmentation par U-net, par la combinaison de VGG-16 avec U-Net pour le transfert Learning la modification de certains paramètres du modèle U-Net, et une approche innovante de l'augmentation des données.

Chapitre 4 Nous implémentons notre architecture U-Net et effectuons l'apprentissage sur Colab, en utilisant notre base de données.

Ensuite nous réalisons une expérience optique à base de micromiroirs pour simuler le traitement simultané de toutes les zones affectées simultanément.

Enfin nous terminons notre mémoire par une conclusion.

Chapitre 01 :

Généralités sur les rétinopathies diabétiques

1 Chapitre 1 Généralités sur les rétinopathies diabétiques

1.1 Introduction

La rétinopathie diabétique est l'une des complications les plus fréquentes du diabète, pouvant entraîner une baisse significative de la vision, voire la cécité.

Cette maladie résulte de l'effet de l'hyperglycémie sur les petits vaisseaux sanguins de la rétine, provoquant leur détérioration et leur fragilisation au fil du temps.

Dans ce chapitre, nous présenterons les différentes parties de l'œil, puis nous aborderons en détail la rétinopathie diabétique : ses types, ses symptômes, ses méthodes de diagnostic et de traitement en particulier celui par laser.

Nous mettrons ensuite en évidence l'importance de l'imagerie du fond d'œil dans la détection précoce des signes de la maladie, notamment les micro-anévrismes (MA), en raison de leur rôle clé dans l'évolution de la pathologie.

Enfin, nous expliquerons comment identifier et traiter les MA de manière appropriée Nous avons également abordé les généralités sur les micromiroirs DMD.

1.2 Éléments (Composants) de fond d'œil

1.2.1 La rétine (Retina)

C'est la couche sensible à la lumière à l'intérieur de l'œil, qui contient deux types de photorécepteurs :

- Les bâtonnets (Rods) : responsables de la vision nocturne et de la perception des ombres.
- Les cônes (Cones) : responsables de la perception des couleurs et des détails fins.

La rétine (fig1.1) convertit la lumière en signaux électriques envoyés au cerveau par le nerf optique.

Elle peut être affectée par des maladies telles que La rétinopathie diabétique (une atteinte des vaisseaux sanguins due au diabète).

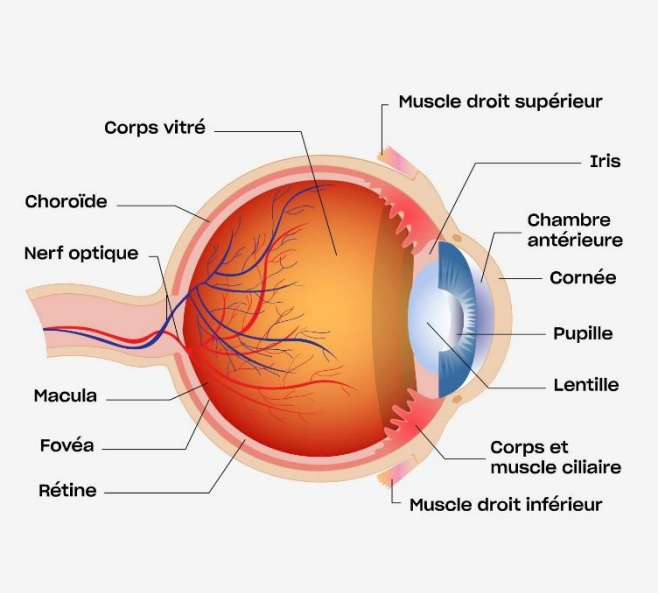


Figure 1.1: La rétine [29]

1.2.2 Le disque optique (Optic Disc)

C'est le point où le nerf optique quitte la rétine pour transmettre les signaux visuels au cerveau. Il ne contient pas de récepteurs lumineux, c'est pourquoi on l'appelle aussi la tache aveugle (Blind Spot).

1.2.3 Les vaisseaux sanguins rétiniens (Retinal Blood Vessels)

Leur fonction principale est d’apporter l'oxygène et les nutriments à la rétine.

Les vaisseaux sanguins se ramifient à partir du disque optique pour irriguer toute la rétine. On distingue deux types principaux :

- **Les artères rétiniennes (Arteries)**
  elles transportent le sang oxygéné du cœur vers la rétine.

- **Les veines rétiniennes (Veins)**
  elles évacuent le sang désoxygéné de la rétine vers le cœur.

Toute modification de ces vaisseaux, comme un rétrécissement, une dilatation ou des hémorragies, peut être un signe de maladies telles que l'hypertension artérielle ou la rétinopathie diabétique.

1.2.4 La macula (Macula)

Elle est responsable de la vision centrale précise, essentielle pour la lecture, la conduite et la reconnaissance des visages …

En son centre se trouve la foyère (Fovea), qui contient la plus forte concentration de cônes (Cones), les cellules responsables de la perception des couleurs et des détails fins.

Toute atteinte de cette zone peut entraîner une perte de la vision centrale.

1.3 La rétinopathie diabétique

La rétinopathie diabétique est une complication du diabète qui affecte la rétine, la couche sensible à la lumière située à l'arrière de l'œil.

Elle est causée par une hyperglycémie prolongée qui endommage les petits vaisseaux sanguins de la rétine.

L'un des premiers signes de cette atteinte est l'apparition de micro-anévrismes (fig1.2), de petites dilatations en forme de ballon dans les capillaires rétiniens, ces micro-anévrismes peuvent fuir et provoquer des saignements, entraînant des troubles visuels s'ils ne sont pas traités [1].

Figure 1.2: Les stades de la rétinopathie diabétique [2]

1.4 Les types de rétinopathie diabétique

Elle évolue généralement en deux stades principaux :

1.4.1 Rétinopathie diabétique non proliférante (NPDR)

Il s'agit du stade précoce de la maladie, où les vaisseaux sanguins sont fragiles mais sans formation de nouveaux vaisseaux anormaux.

Les signes peuvent inclure :

- **Microanévrysmes (Microaneurysms - MA**)
  petites dilatations des capillaires rétiniens, considérées comme le premier signe clinique visible de la rétinopathie diabétique.
- **Taches cotonneuses (Cotton Wool Spots)**
  petites zones blanches dues à l'obstruction des capillaires, entraînant un manque d'oxygène dans la rétine.
- **Modifications des vaisseaux sanguins (Venous Beading)**
  anomalies dans la structure des veines rétiniennes. [3]

1.4.2 Rétinopathie diabétique proliférante (PDR)

Il s'agit d'un stade avancé où de nouveaux vaisseaux sanguins anormaux commencent à se développer (néovascularisation).

Ces vaisseaux sont dangereux car :

- Ils sont fragiles et peuvent saigner facilement, entraînant une perte de vision.
- Ils peuvent provoquer un décollement de la rétine (Retinal Detachment) en raison de la fibrose induite par leur croissance anormale. [3]

1.5 Les symptômes de la rétinopathie diabétique

À un stade précoce, notamment en présence de micro-anévrismes, la maladie peut être asymptomatique. Toutefois, lorsque la rétinopathie progresse, les symptômes peuvent inclure :

- Une vision floue, causée par la fuite de liquide des micro-anévrismes.
- Des taches sombres ou corps flottants, résultant d'hémorragies capillaires.
- Une vision fluctuante, due aux variations de l'œdème rétinien.
- Des difficultés à voir la nuit, en raison de la détérioration progressive de la rétine.
- Une perte de vision, dans les cas avancés avec des lésions étendues.[4]

1.6 Le traitement de la rétinopathie diabétique

Le traitement dépend du stade de la maladie et de l'impact des micro-anévrismes sur la rétine :

1.6.1 Stade précoce (RDNP avec micro-anévrismes) :

- Contrôle glycémique : Maintenir des niveaux normaux de glycémie, de tension artérielle et de cholestérol pour limiter la progression des micro-anévrismes.
- Surveillance régulière : Examens fréquents pour détecter toute fuite ou hémorragie.

1.6.2 Stade avancé (RDNP sévère ou RDP)

- Thérapie au laser (photocoagulation focale) : Scelle les micro-anévrismes fuyants pour empêcher l'aggravation de l'œdème rétinien.
- Injections d'anti-VEGF : Bloque la croissance anormale des vaisseaux sanguins.

- Vitrectomie : Chirurgie pour éliminer le sang du vitré et éviter le décollement rétinien.

Les micro-anévrismes sont des indicateurs précoces de la rétinopathie diabétique, soulignant l'importance du dépistage précoce [3].

1.7 Système d'imagerie du fond d'œil iCare DRSplus avec technologie confocale TrueColor

Le système d'imagerie confocale du fond d'œil iCare DRSplus (fig1.3) utilise un éclairage LED blanc pour fournir des images TrueColor de haute qualité. La technologie confocale TrueColor, considérée comme une référence en matière de qualité d'image, offre des images riches en détails avec une netteté accrue, une résolution optique améliorée et un meilleur contraste par rapport aux caméras traditionnelles du fond d'œil.

Rapide et entièrement automatisé, l'iCare DRSplus permet d'obtenir des images à travers des pupilles aussi petites que 2,5 mm, sans nécessiter de dilatation, garantissant ainsi une expérience confortable pour le patient. Cet appareil facile à utiliser offre l'avantage d'un temps d'examen réduit et contribue à accélérer le flux de travail dans les cliniques [5].

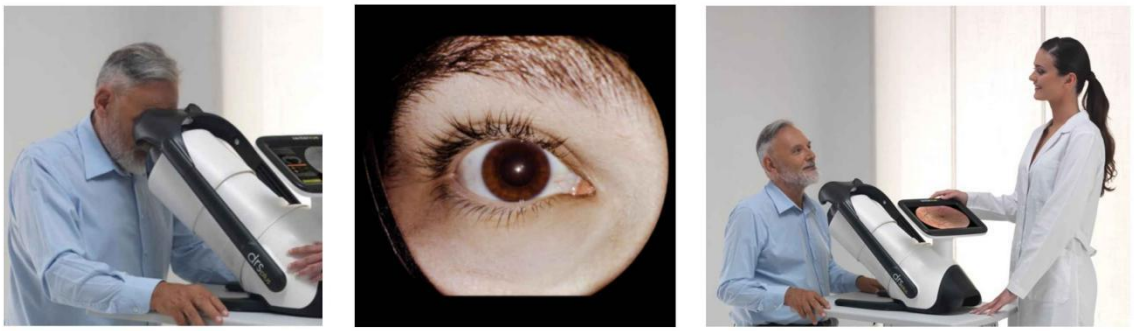


Figure 1.3: Système d'imagerie du fond d'œil iCare DRSplus avec technologie confocale TrueColor [5]

1.7.1 Différentes modalités d'imagerie pour obtenir des images riches en détails

La technologie confocale TrueColor avec éclairage LED blanc permet de capturer des images rétiniennes (fig1.4) détaillées à 45° et de scanner à travers la cataracte, aidant ainsi les médecins dans le diagnostic et la documentation des maladies oculaires.

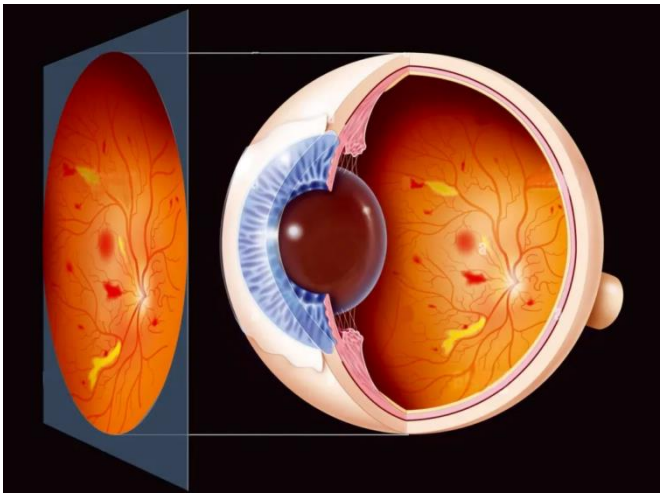


Figure 1.4: Vue schématique de la capture d'image rétinienne par la technologie confocale TrueColor [6]

Le filtrage haute résolution sans rouge améliore la visualisation de la vascularisation rétinienne, tandis que les images en bleu offrent une meilleure vue de la couche des fibres nerveuses (RNFL).

Le canal rouge permet à la lumière de pénétrer profondément dans les couches rétiniennes.

De plus, la lumière infrarouge fournit des informations détaillées sur la choroïde.

Les images de l'œil externe peuvent documenter les conditions de la surface de l'œil et de la cornée.

La technologie de visualisation stéréoscopique améliore la perception 3D de la tête du nerf optique.

Enfin, la fonction mosaïque combine automatiquement différents champs rétiniens sans intervention de l'utilisateur, créant ainsi des vues panoramiques allant jusqu'à 80° [5].

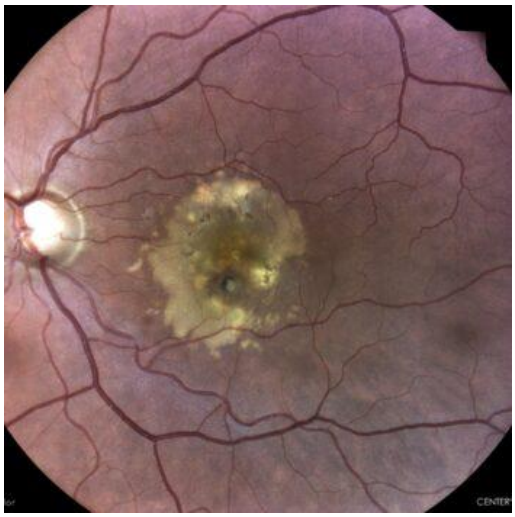


Figure 1.5: Image couleur standard du fond d'œil obtenue avec la technologie TrueColor en lumière LED blanche à 45° [5]

1.7.2 Les limites de iCare ILLUME pour la détection de la rétinopathie diabétique

Le système iCare ILLUME, combiné au dispositif iCare DRSplus, est une solution intégrée que facilite le dépistage automatisé de la rétinopathie diabétique à l'aide de l'intelligence artificielle.

Il permet une analyse rapide des images du fond d'œil et peut alerter sur la présence de signes de rétinopathie, même à un stade précoce (fig 1.6), ce qui en fait un outil utile dans les programmes de dépistage à grande échelle.

Cependant, certaines limites doivent être prises en compte :

- Le système ne permet pas de localiser précisément les lésions (comme les microanévrysmes, hémorragies ou néovascularisations).
- Il fournit un diagnostic de type binaire (présence ou absence de rétinopathie), sans indiquer la gravité ou le stade exact de la maladie.

Ainsi, bien que iCare ILLUME soit un excellent outil d'aide au dépistage, il ne remplace ni l'examen clinique détaillé, ni l'interprétation spécialisée, surtout pour les décisions thérapeutiques [5].

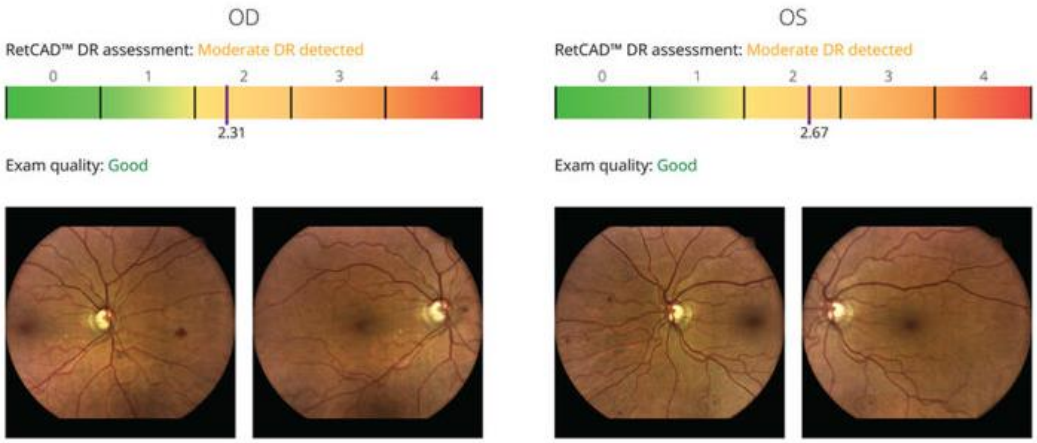


Figure 1.6: Détection assistée par IA de la rétinopathie diabétique [5]

1.8 Les micro-anévrismes

Les micro-anévrismes (fig1.7) sont de petites dilatations des parois des vaisseaux sanguins de la rétine et constituent le premier et principal signe précoce de la rétinopathie diabétique (Diabetic Retinopathy - DR).

Ils apparaissent en raison de l'affaiblissement des parois capillaires causé par une glycémie élevée, ce qui entraîne une accumulation de liquide ou une fuite de sang dans la rétine.

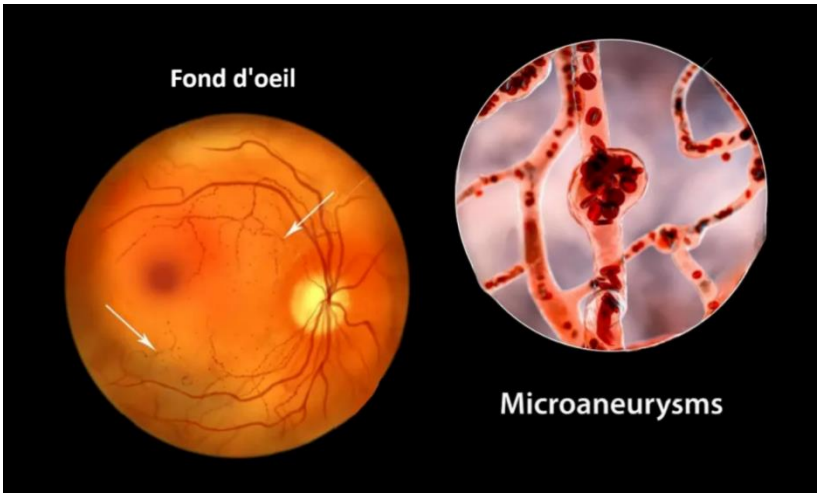


Figure 1.7 : Illustration 3D de micro-anévrismes rétiniens avec vue rapprochée des dilatations artérielles remplies de sang [7]

1.8.1 L'apparition des micro-anévrismes sur les images du fond d'œil

Ils apparaissent sous forme de petits points rouges dispersés dans la rétine, en particulier près de la macula.

Ils peuvent être isolés ou regroupés dans certaines zones.

Ils sont visibles grâce à :

- La photographie du fond d'œil en couleur (Color Fundus Photography).
- L'angiographie à la fluorescéine (Fluorescein Angiography - FA), où ils apparaissent comme des zones de fuite de colorant.

1.8.2 La dangerosité des micro-anévrismes

Au stade précoce de la rétinopathie diabétique non proliférante (NPDR), ils peuvent ne provoquer aucun symptôme.

Si leur nombre augmente ou s'ils se rompent, ils peuvent entraîner :

- Des hémorragies rétiniennes.
- Une ischémie rétinienne, causant une perte progressive de la vision.
- Un œdème maculaire diabétique (DME), responsable d'une vision floue ou déformée.

1.8.3 Le traitement des micro-anévrismes

Le traitement vise à limiter les dommages à la rétine et à prévenir l'évolution vers des stades plus graves de la rétinopathie diabétique.

Il repose sur plusieurs approches complémentaires, adaptées à la gravité des lésions observées :

- **Contrôle de la glycémie** La gestion du diabète réduit la progression de la maladie.
- **Photocoagulation au laser (Laser Photocoagulation)** Utilisée pour brûler les vaisseaux sanguins fragiles et prévenir les fuites de liquide.

- **Injections intraoculaires** Des anti-VEGF (ex : Avastin, Eylea) sont administrés pour bloquer la croissance des vaisseaux sanguins anormaux et réduire l'œdème.
- **Suivi régulier du fond d'œil** Un diagnostic précoce permet de prévenir des complications graves comme la rétinopathie proliférante (PDR) ou le décollement de la rétine. [8]

1.8.4 Traitement des micro-anévrismes par laser (Photocoagulation laser)

Le laser provoque une coagulation thermique localisée dans les vaisseaux sanguins anormaux (fig 1.8), permettant ainsi de limiter les exsudations et de stabiliser l'état de la rétine.

Les principales caractéristiques du laser utilisé sont les suivantes :

- Longueurs d'onde : 532 nm ou 577 nm (laser jaune), bien absorbées par les vaisseaux rétiniens.
- Types : Laser Argon, laser à colorant jaune (Yellow Dye Laser).
- Durée d'impulsion : 10 à 200 ms – impulsions courtes pour réduire les effets thermiques.
- Puissance : Entre 100 et 500 µW, ajustée selon le patient.
- Taille du spot : 50 à 200 µm, en fonction de la taille des lésions. [9][ChatGPT+Ophtalmologue]

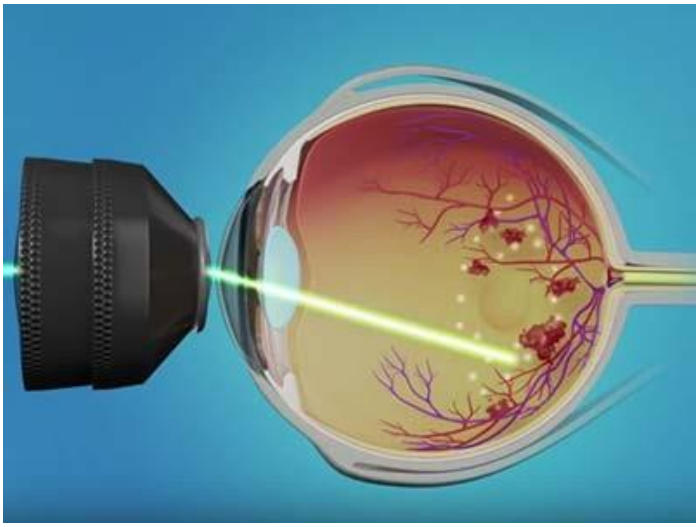


Figure 1.8: Laser de correction [31]

1.9 Généralités sur les micromiroirs DMD

1.9.1 Définition

Le DMD est un composant optoélectronique développé par Texas instruments, compose d'une matrice de micromiroirs mobile.

Chaque miroir représente un pixel individuel et peut s'incliner pour moduler la direction de la lumière incidente. Ce dispositif est au cœur de la technologie DLP (Digital Light Processing) et permet un contrôle très précis de la lumière.

1.9.2 Principe de fonctionnement

Chaque micromiroir est monté sur une micro charnière qui lui permet de basculer entre deux positions (typiquement ±12°), contrôlées électriquement. En position ON, le miroir reflète la lumière vers le système optique ; en position OFF, il la détourne. L'ensemble des miroirs peut ainsi former un motif lumineux binaire, utilisable pour l'affichage ou le traitement ciblé par faisceau laser.

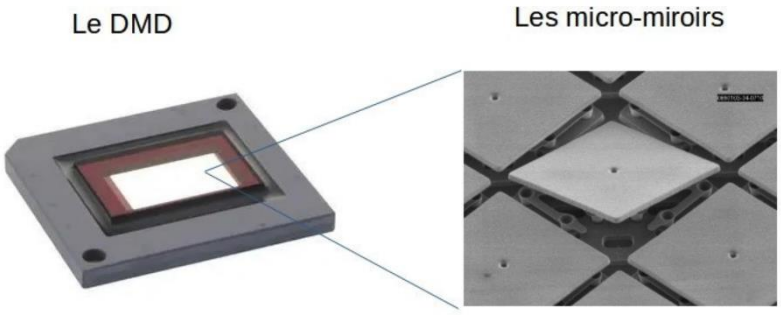


Figure 1.9: dispositif DMD avec agrandissement révélant les micromiroirs [24]

1.9.3 Caractéristiques techniques principales (modèle DLP7000)

Le DMD DLP7000 (fig 1.10) présente les caractéristiques techniques suivantes qui le rendent particulièrement adapté aux applications de modulation optique de haute précision :

- Résolution : 1024 × 768 pixels
- Taille de la puce DMD : 0,7 pouce de diagonale
- Nombre de micro-miroirs : 786 432 miroirs (un par pixel)

- Fréquence d'affichage des motifs : jusqu'à 32 552 Hz pour les motifs binaires (1 bit)
- Débit de transfert de données : 25,2 Gbit/s
- Plage de longueurs d'onde : 400–700 nm (lumière visible)
- Température de fonctionnement : de +10 °C à +65 °C
- Taille de chaque micromiroir : 13,6 micromètres [25]

Figure 1.10: DLP7000

1.9.4 Application

Les DMD sont largement utilisés dans :

- Les vidéoprojecteurs DLP.
- L'imagerie médicale (fluorescence, diagnostic).
- La photocoagulation laser en ophtalmologie.
- La fabrication additive (impression 3D).
- Les systèmes optiques adaptatifs dans les laboratoires de recherche. [25]

1.10 Conclusion

Nous nous sommes intéressés dans ce chapitre à la RD en particulier aux micro-anévrismes, leurs mises en évidence par des méthodes d'intelligence artificielle, et leur traitement par des équipement à base de laser.

Le but de notre projet étant le développement d'une architecture U-Net pour la segmentation des micro-anévrismes (pour leur traitement par laser), nous exposons dans le chapitre suivant les notions de base de l'intelligence artificielle ainsi que les réseaux de neurones convolutifs.

Chapitre 2 :

Généralités sur l’intelligence artificielle et les réseaux de neurones à convolution profonds

2 Chapitre 2 Généralités sur l'intelligence artificielle et les réseaux de neurones à convolution profonds

2.1 Introduction

Pour détecter les micro-anévrismes (MA), nous utiliserons l'apprentissage profond (DL), qui est une branche de l'apprentissage automatique (Machine Learning), lui-même considéré comme un domaine clé de l'intelligence artificielle.

Dans ce chapitre, nous présenterons le concept des réseaux neuronaux (Neural Networks) et leurs différentes architectures, en mettant l'accent sur les réseaux neuronaux convolutifs (CNN - Convolutional Neural Networks) et leurs multiples couches.

Nous aborderons également les algorithmes de détection des micro-anévrismes, notamment U-Net, ainsi que les concepts mathématiques fondamentaux sur lesquels repose l'apprentissage profond, qui jouent un rôle crucial dans la précision et l'efficacité des modèles utilisés.

Nous avons enfin abordé le modèle VGG-16 ainsi que la technique du Transfert Learning, en raison de leur importance dans l'amélioration des performances des réseaux neuronaux lorsque les données disponibles sont limitées.

2.2 Définitions

2.2.1 Intelligence artificielle

L'intelligence artificielle (IA) est un domaine de l'informatique qui vise à créer des systèmes capables d'accomplir des tâches qui nécessitent normalement l'intelligence humaine, telles que la perception visuelle, la reconnaissance vocale, la prise de décision et la traduction automatique.

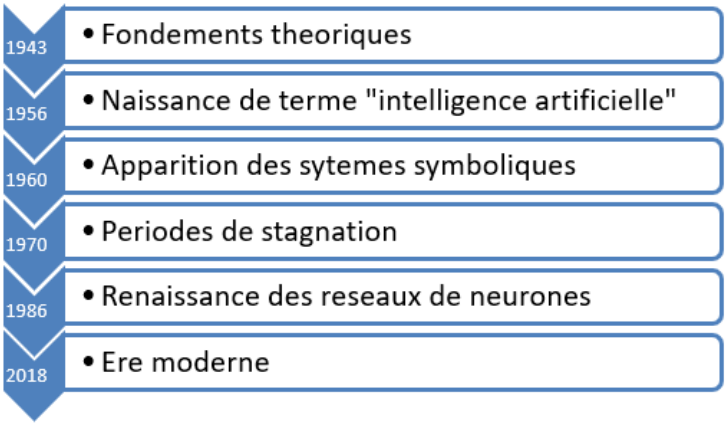


Figure 2.1: Principales étapes du développement de l'IA

2.2.2 La classification

La classification d'images est l'un des domaines fondamentaux de la vision par ordinateur. Elle consiste à attribuer une seule étiquette (ou classe) à une image entière, en se basant sur son contenu visuel. Les réseaux de neurones convolutifs (CNN) sont utilisés pour extraire des caractéristiques et reconnaître des motifs dans les images. Cette technique est utilisée dans de nombreuses applications, notamment pour le diagnostic des maladies à partir d'images médicales (radiographies, scanners, etc.) [10].

2.2.3 La segmentation

La segmentation d'images vise à diviser une image en régions distinctes (segments), de manière à ce que chaque pixel soit classé comme appartenant à un objet ou à l'arrière-plan. Cette technique est essentielle dans les applications nécessitant une grande précision, comme la délimitation des tumeurs dans les images médicales. Des modèles tels que U-Net et Mask R-CNN sont largement utilisés à cette fin [11].

2.2.4 La détection

La détection d'objets est une technique utilisée pour reconnaître les objets présents dans une image et localiser leur position à l'aide de cadres de délimitation (bounding boxes). Contrairement à la classification, cette tâche permet d'identifier plusieurs catégories d'objets ainsi que leur emplacement précis dans l'image. Parmi les modèles les plus connus figurent YOLO (You Only Look Once) et Faster R-CNN. Cette technologie est largement utilisée dans des domaines tels que la sécurité, la surveillance et les véhicules autonomes [12].

Tableau 2.1: Propriétés des fonctions de classification segmentation détection

	Classification	Segmentation	Détection
Objectif	Attribuer une étiquette a l'image entière	Associer chaque pixel a une classe (objet / fond)	Identifier et localiser plusieurs objets dans l'image
Sortie	Une seule classe pour toute l'image	Masque pixel par pixel indiquant les contours précis	Boites + classe pour chaque objet
Précision	Global	Très élevé (niveau pixel)	Moyen a élevé
Nombre d'objets détectés	Un seul	Plusieurs objets avec leur position	Plusieurs objets avec leur position
Exemples d'application	Reconnaissance de maladies ou de scènes	Segmentation d'orgones	Caméras de surveillance
Modèles populaires	ResNet	U-Net, Mask R-CNN	YOLO, Faster R-CNN

2.3 L'apprentissage (entrainement)

L'apprentissage est une étape essentielle dans le développement du réseau neuronal, au cours de laquelle les poids et les paramètres du réseau sont ajustés afin d'optimiser ses performances et d'atteindre le comportement souhaité avec précision. On peut distinguer les types d'apprentissage suivants :

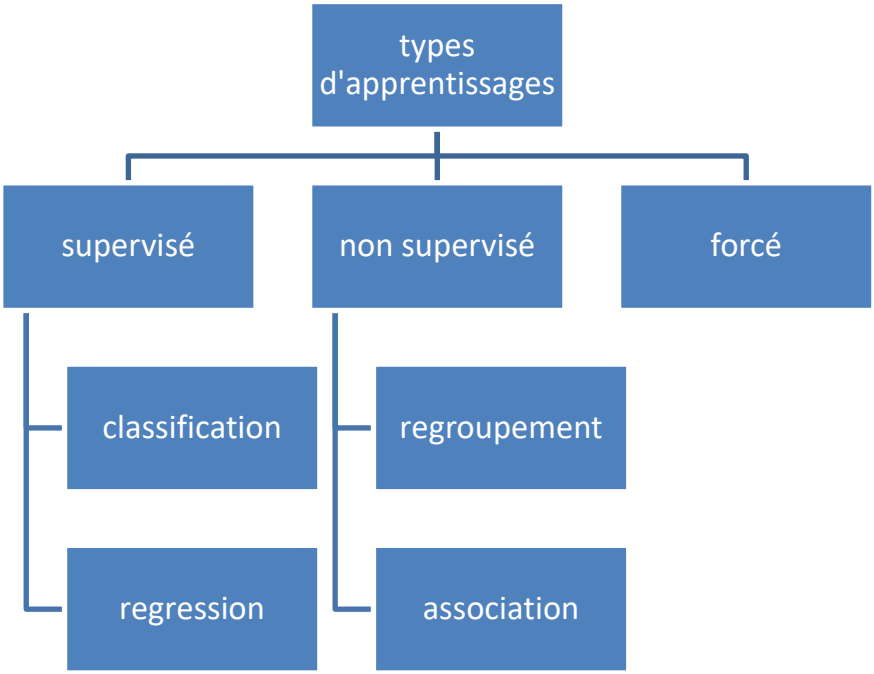


Figure 2.2: Schéma synoptique des types d'apprentissage

2.3.1 L'apprentissage supervisé

Dans l'apprentissage supervisé (fig 2.3), un système IA est présenté avec des données qui sont étiquetées (labeled), ce qui signifie que chaque donnée est étiquetée avec l'étiquette correcte.

- L'étiquette est utilisée pour entrainer le modèle supervisé.
- Une fois le modèle est entrainé, il est testé par des données de la base de test (nouvelles données) et vérifier si le modèle est capable de prédire la sortie correcte.

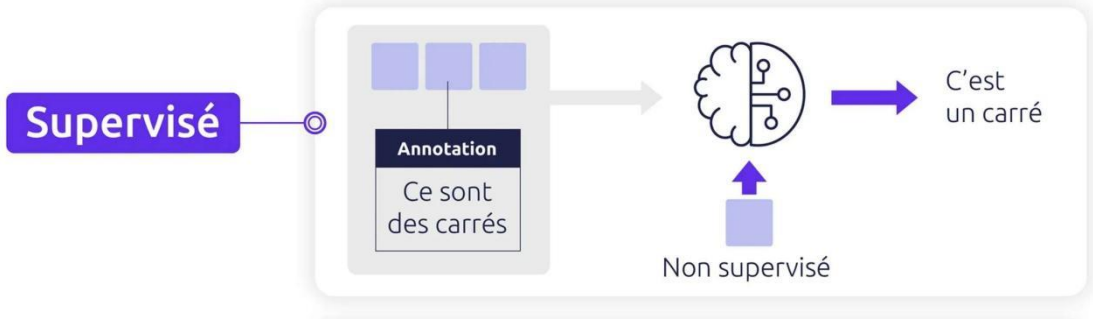


Figure 2.3: schéma illustratif du processus d'apprentissage supervisé [13]

Types d'apprentissage supervisé :

- **Classification :** on parle d'un problème de classification quand la variable de sortie est une catégorie, tels que ; 'rouge' et 'bleu', ou 'maladie' et 'pas de maladie'.
- **Régression :** on parle d'un problème de régression quand la variable de sortie est une valeur réelle, tels que ; 'dollars' ou 'poids'

2.3.2 L'apprentissage non supervisé

Dans l'apprentissage non supervisé (fig 2.4), un système IA est présenté avec des données qui sont non étiquetées (labeled), non catégorisées.

- Dans la base de données, les données n'ont pas d'étiquettes correspondantes,
- Le modèle non supervisé est capable de séparer les données moyennant les types de ces données et modélise la structure sous-jacente (implicite, pas claire) ou distribution dans les données de telle façon à apprendre mieux.

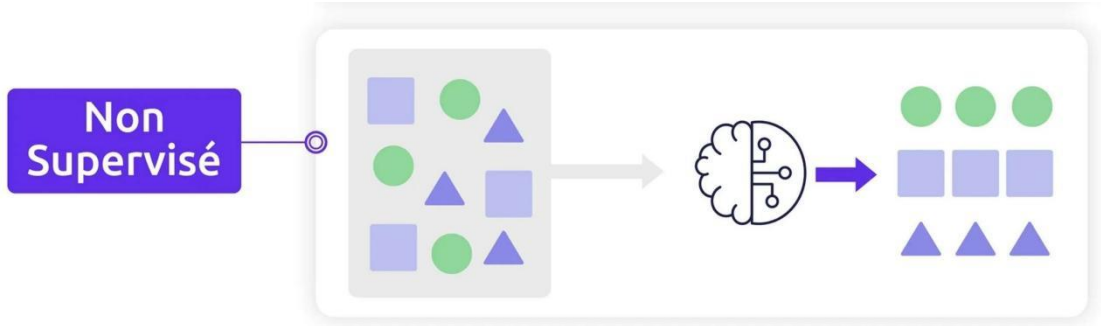


Figure 2.4: schéma illustratif du processus d'apprentissage non-supervise [13]

Types d'apprentissage non supervisé :

- **Regroupement (clustering) :** on parle d'un problème de clustering quand on veut découvrir les groupements inhérents (liés, inséparables) dans les données (data), tel que, les groupements des clients par le comportement d'achat,

- **Association :** on parle d'un problème d'association quand on veut découvrir les règles qui décrivent les grandes portions des données, tel que la population qui achète X et aussi tend à acheter Y, donc on va créer un groupement de X + Y.

2.3.3 L'apprentissage forcé

Un algorithme d'apprentissage forcé (fig 2.5), agent, apprend par interaction avec son environnement.

L'agent reçois des récompenses pour avoir effectué des actions correctes et des pénalités pour des actions incorrectes.

L'agent apprend (tout seul) sans l'intervention de l'homme par maximisation des récompenses et minimisation des pénalités.

C'est un type de programmation dynamique qui entraine les algorithmes en utilisant un système de récompenses et pénalités

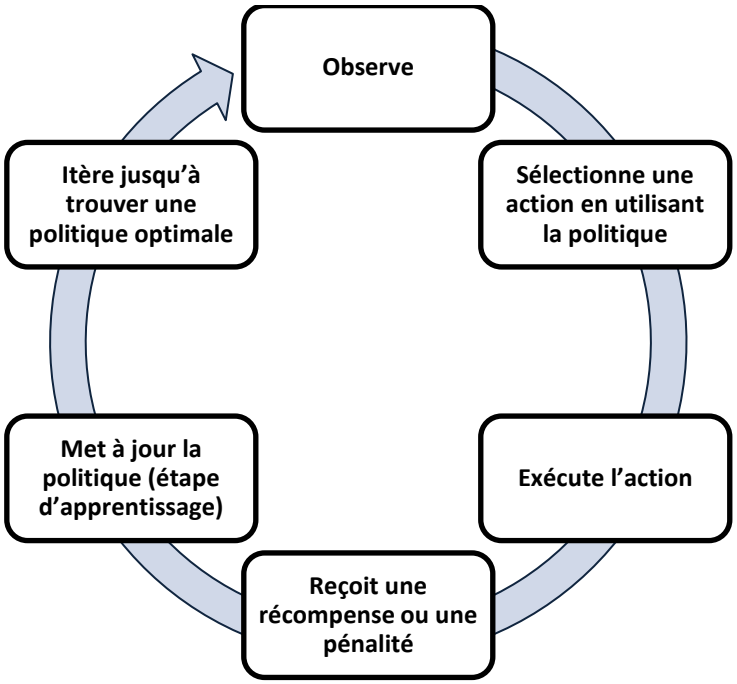


Figure 2.5: schéma illustratif du processus d'apprentissage forcé

2.4 Les fonctions d'activation

2.4.1 La fonction ReLu

Les réseaux neuronaux imitent le cerveau humain, capable de traiter des problèmes non linéaires (complexes). Sans non-linéarité, un réseau neuronal deviendrait une simple fonction linéaire, ce qui signifie qu'il ne pourrait pas résoudre des problèmes complexes.

La fonction d'activation ReLU (Rectified Linear Unit) transforme toutes les valeurs négatives en zéro tout en conservant les valeurs positives telles qu'elles sont :

$$f(x) = \max(0, x)$$

Cela signifie qu'elle ajoute une forme de non-linéarité à la couche neuronale.

Cette fonction présente plusieurs avantages :

- Élimination des valeurs inutiles : Lorsque les valeurs sont négatives (caractéristiques non pertinentes), ReLU les rend nulles, réduisant ainsi les calculs inutiles.
- Apprentissage des relations non linéaires : Grâce à la non-linéarité, le réseau peut comprendre des modèles plus complexes dans les données, comme les formes ou les images avec des structures compliquées.
- Efficacité des calculs : Comparée à d'autres fonctions d'activation (telles que Sigmoid ou Tanh), ReLU est plus simple et rapide à calculer, ce qui la rend idéale pour les réseaux profonds.[14]

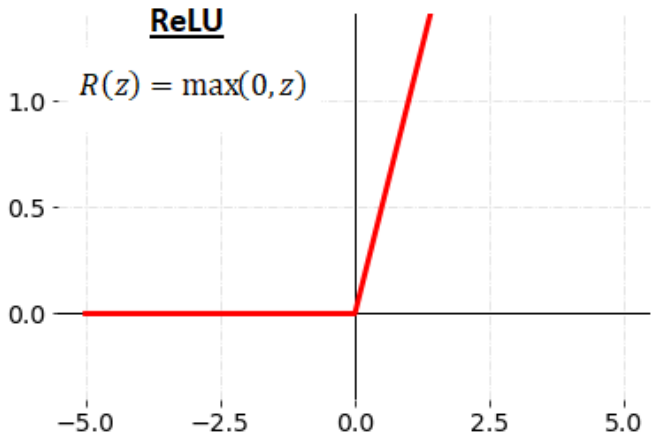


Figure 2.6: Graphe de la fonction ReLU

2.4.2 La fonction sigmoïd

La fonction Sigmoid est une fonction mathématique qui transforme les valeurs d'entrée en un intervalle compris entre 0 et 1, selon la formule suivante :

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Elle prend une valeur d'entrée x et produit une probabilité comprise entre 0 et 1.

- Les valeurs proches de 1 indiquent une forte probabilité de présence de la caractéristique ou de l'objet.
- Les valeurs proches de 0 indiquent une faible probabilité (absence).

Cette fonction est particulièrement utilisée dans des tâches telles que :

a) segmentation binaire des images

Dans des applications comme U-Net, la fonction Sigmoid est souvent utilisée pour la segmentation d'images, où chaque pixel doit être classé dans l'une des deux catégories :

- Catégorie 1 (Premier plan) : L'objet d'intérêt (par exemple, MA sur une image fond d'œil).
- Catégorie 0 (Arrière-plan) : Le reste de l'image.

Si la probabilité donnée par Sigmoid pour un pixel est 0.9, le modèle prédit que ce pixel appartient à la catégorie 1 avec une probabilité de 90 %.

Si la probabilité est 0.1, le pixel appartient probablement à la catégorie 0.

b) Réduction de la complexité calculatoire

- Simplicité : La fonction Sigmoid est relativement simple à calculer comparée à d'autres fonctions.
- Stabilité : Elle rend l'entraînement du modèle plus stable et fluide en fournissant une sortie normalisée, ce qui est essentiel pour optimiser les poids efficacement. [16]

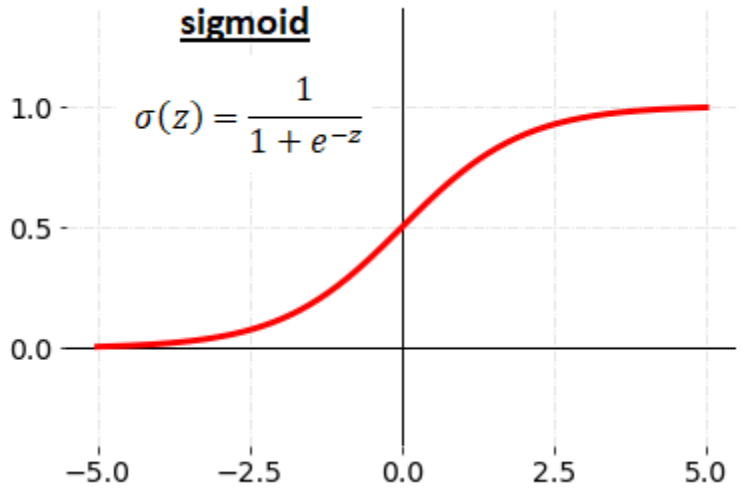


Figure 2.7: Graphe de la fonction sigmoïde

2.5 Fonction de perte (Dice Loss)

Les fonctions de perte quantifient l'écart entre les prédictions du réseau neuronal et la vérité.

Dice Loss est dérivée du coefficient de Dice, une mesure statistique utilisée pour évaluer la similarité entre deux ensembles. Le coefficient de Dice est défini comme suit :

$$Dice\ Score = \frac{2|MaskReal \cap MaskPredit|}{|MaskReal| + |MaskPredit|} = F1\ Score$$

La Dice Loss est calculée comme suit :

$$Dice\ Loss = 1 - Dice = 1 - F1\ Score$$

Si la similarité entre les prédictions et masque réel est élevée, le coefficient de Dice est proche de 1, et la perte est donc faible.

La Dice Loss est un bon choix grâce à plusieurs avantages importants :

Gestion du déséquilibre des classes

- Dans les problèmes de segmentation d'images, l'arrière-plan occupe souvent une portion beaucoup plus grande que l'objet à segmenter.
- La Dice Loss accorde une importance plus grande aux petites régions (comme les objets d'intérêt) qu'à l'arrière-plan, ce qui la rend efficace pour apprendre les contours précis des petits objets.

Amélioration de la similarité entre prédictions et masques réels

- Contrairement aux métriques qui se concentrent uniquement sur la précision au niveau des pixels (Pixel Accuracy), la Dice Loss évalue la correspondance globale entre les formes des objets.

Idéale pour la segmentation binaire

- La Dice Loss est particulièrement adaptée aux scénarios binaires (par exemple, classifier un pixel comme appartenant à un objet ou à l'arrière-plan). Elle vise à maximiser la similarité entre le masque prédit et le masque réel. [16]

2.6 Les réseaux de neurones convolutifs (CNN)

Les réseaux neuronaux convolutifs (CNN) (fig2.8) sont des modèles d'apprentissage profond capables d'apprendre des fonctions de mappage très complexes, notamment lorsqu'ils sont entraînés sur un ensemble de données suffisamment vaste et varié. Ils sont largement utilisés en traitement d'images et en vision par ordinateur grâce à leur capacité à extraire automatiquement des caractéristiques importantes à partir d'images brutes.

À un niveau fondamental, un CNN est composé de plusieurs couches principales :

Couches de convolution (Convolutional Layers) :

Elles appliquent des filtres (kernels) pour extraire des motifs et des caractéristiques locales à différents niveaux de l'image, comme les contours, textures et formes complexes.

Couches de pooling (Pooling Layers) :

Elles réduisent la dimension spatiale des données tout en conservant les informations essentielles, améliorant ainsi l'efficacité du modèle et réduisant le risque de surapprentissage (overfitting).

Couches entièrement connectées (Fully Connected Layers) :

Elles combinent les caractéristiques extraites par les couches précédentes et effectuent la classification finale. [17]

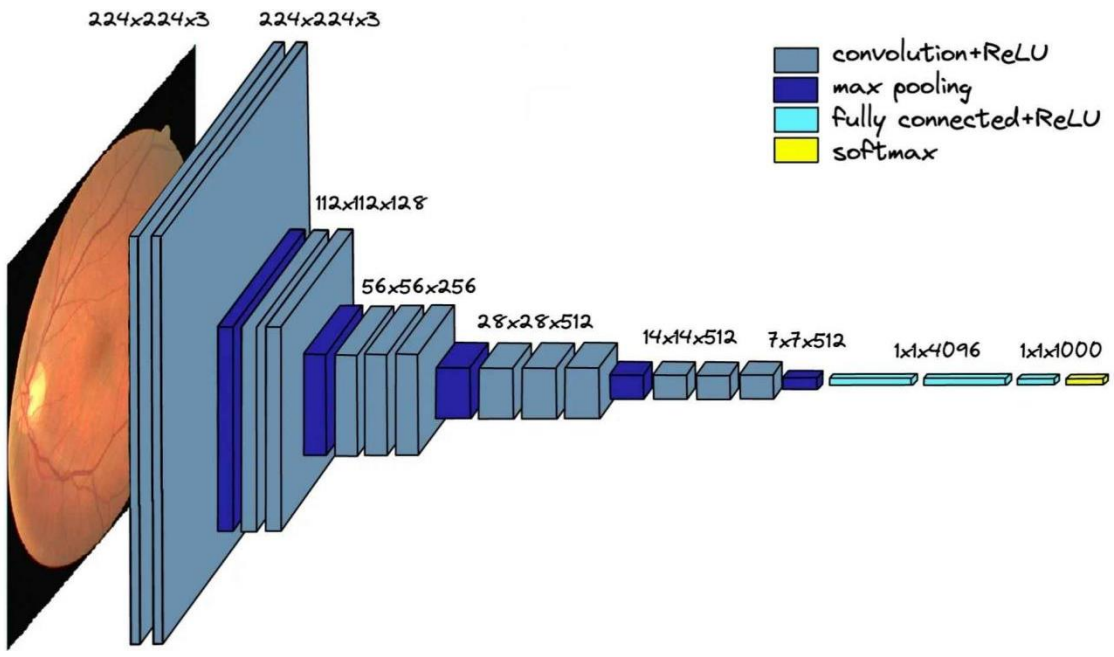


Figure 2.8: Représentation d'un réseau de neurones convolutif

2.6.1 La convolution

L'objectif de cette couche est de détecter et extraire des caractéristiques visuelles telles que les bords, les textures, les motifs et les variations de couleur présentes dans une image. La convolution de l'image (fig 2.9) avec un filtre génère des cartes de caractéristiques de sortie (feature maps), qui représentent les informations essentielles détectées à différentes échelles.

Plus le nombre de filtres dans les couches de convolution est élevé, plus le réseau peut capturer des détails complexes et identifier des motifs spécifiques à différents niveaux d'abstraction. L'opération de convolution commence généralement dans le coin supérieur gauche de la matrice d'entrée et se déplace progressivement sur toute l'image, appliquant le filtre à chaque position en effectuant un produit scalaire entre les pixels de l'image et les valeurs du noyau (kernel). Cette transformation permet de mettre en évidence les structures importantes tout en réduisant les informations redondantes.

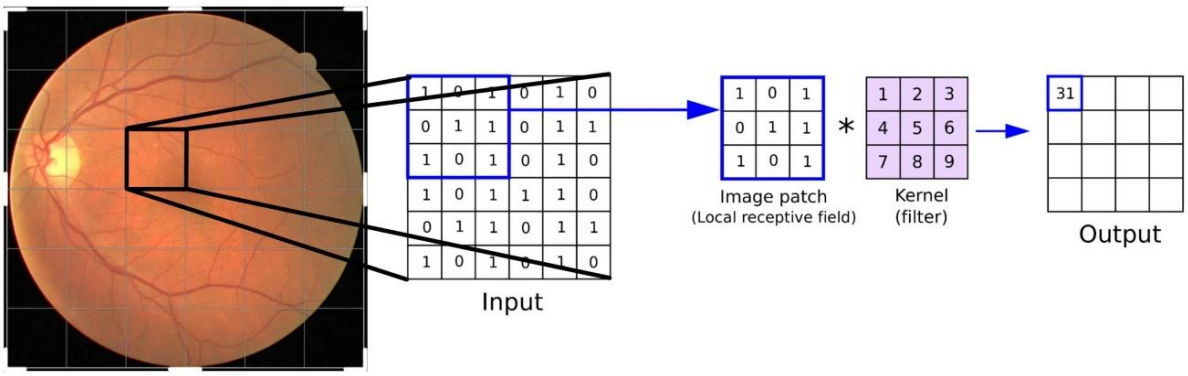


Figure 2.9: Exemple de produit de convolution

2.6.2 Le pooling

Max Pooling :

Le Max Pooling (fig2.10) est une technique de réduction de dimension utilisée dans les réseaux de neurones convolutionnels (CNN) pour extraire les caractéristiques les plus importantes tout en réduisant la complexité computationnelle. Il fonctionne en divisant l'image en petites régions (par exemple, des fenêtres de 2x2 ou 3x3) et en sélectionnant la valeur maximale dans chaque région. Cela permet de préserver les caractéristiques les plus saillantes, comme les contours …

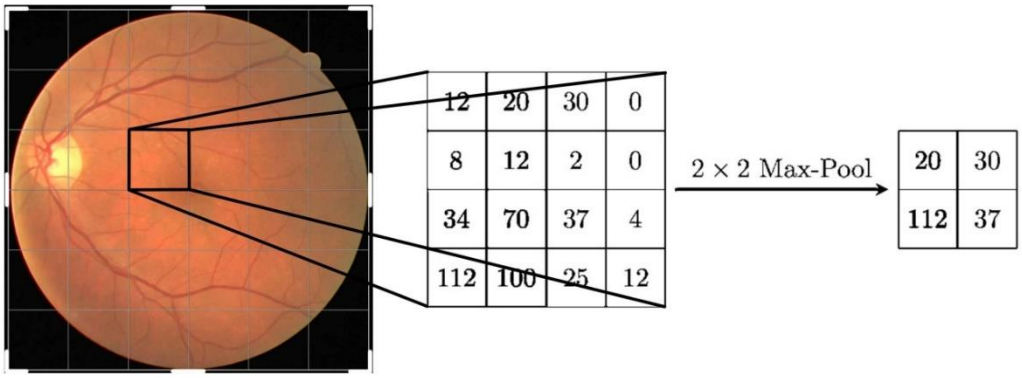


Figure 2.10: Exemple du Max Pooling

Moyenne Pooling :

Contrairement au Max Pooling, qui sélectionne uniquement la valeur maximale d'une fenêtre donnée, le Moyenne Pooling calcule la moyenne des valeurs des pixels dans chaque région. Cette approche permet de réduire le bruit et de lisser les caractéristiques extraites.

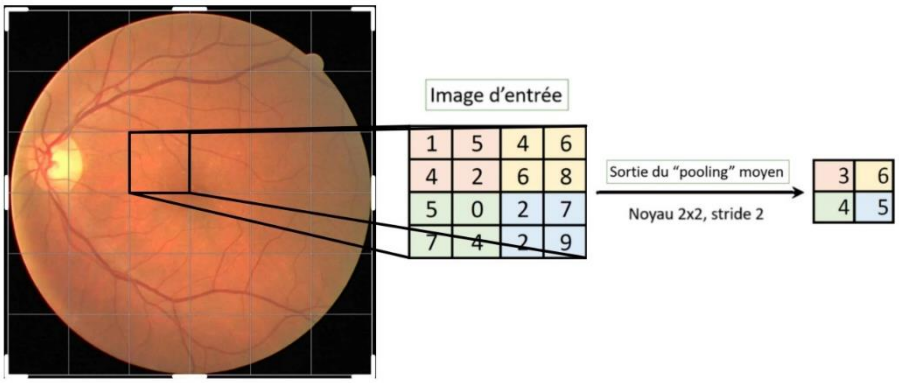


Figure 2.11: Exemple du Moyenne Pooling

Global Average Pooling :

Contrairement aux couches de moyenne pooling classiques, qui opèrent sur des petites régions, le GAP applique une moyenne sur l'ensemble de chaque carte de caractéristiques

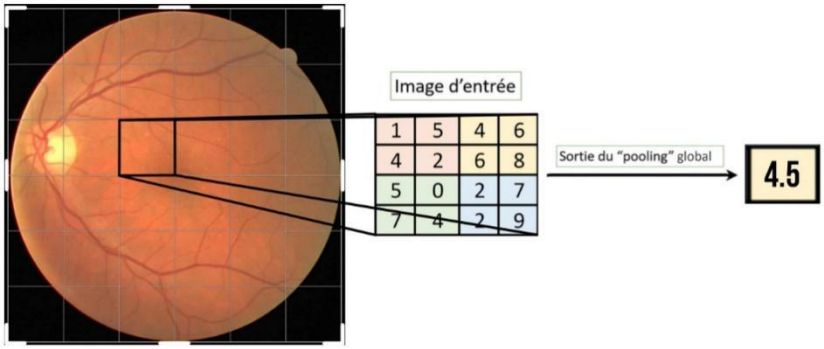


Figure 2.12: Exemple du Global Average Pooling

2.6.3 Carte des caractéristiques « Features Map »

La carte des caractéristiques (Feature Map) (fig2.13) est le résultat obtenu après l'application d'un noyau (kernel) sur une image dans un réseau de neurones convolutionnels (CNN). Elle met en évidence les caractéristiques importantes de l'image, comme les contours, les textures ou les formes spécifiques.

Chaque feature map correspond à un filtre appliqué, ce qui permet au réseau de capturer différents détails à chaque couche. Plus on avance dans le réseau, plus les cartes des caractéristiques deviennent abstraites.

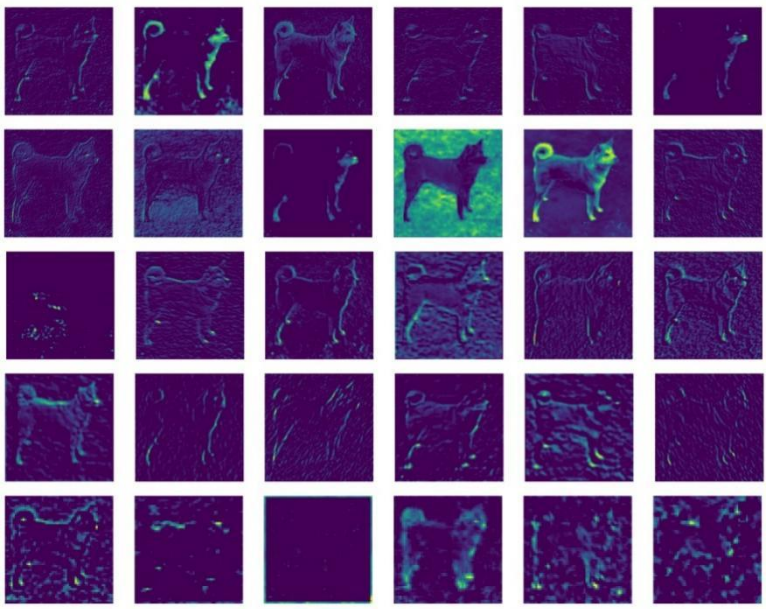


Figure 2.13: Example de cartes des caractéristiques extraites d'une image par un U-Net non entraîné

2.6.4 Le pas « Stride »

Le pas (stride), c'est juste le nombre de pixels que le filtre se déplace à chaque fois quand il parcourt l'image pendant la convolution. Si le pas est 1, le filtre avance pixel par pixel, ce qui permet d'obtenir une carte des caractéristiques très détaillée. Mais si on utilise un pas plus grand (comme 2 ou 3), l'image de sortie sera plus petite et le modèle ira plus vite, même si on risque de perdre un peu d'informations.

2.6.5 Le remplissage (Padding)

Le remplissage (padding) consiste à ajouter des pixels (souvent de valeur zéro) autour de l'image avant la convolution. Cela permet de conserver la même taille après le filtrage et d'éviter une réduction progressive des dimensions.

Bien que le padding n'ajoute aucun paramètre supplémentaire, il peut légèrement augmenter le temps de calcul, mais son impact reste généralement négligeable.

$$L_{Out} = (\frac{L_{in} - K + 2P}{S})$$

$$H_{Out} = (\frac{H_{in} - K + 2P}{S})$$

Avec :

L_{in} : Largeur de l'entrée

H_{in} : Hauteur de l'entrée

K : taille de filtre (kernel) K=3 pour un filtre 3x3

P : padding

S : stride

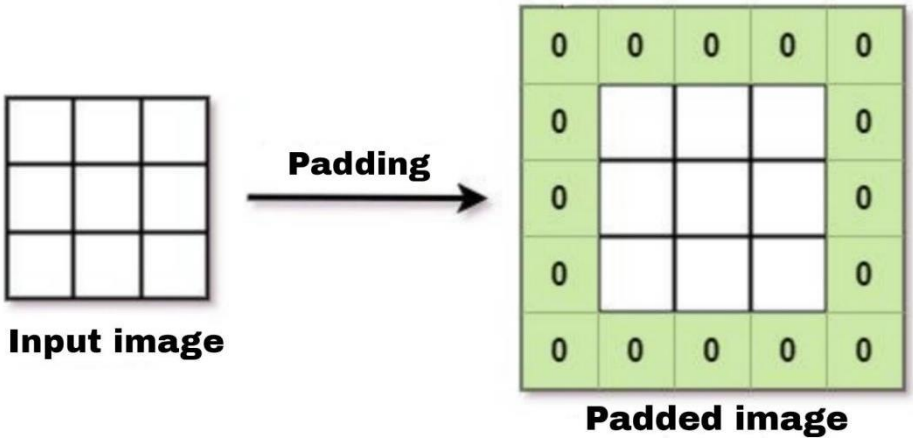


Figure 2.14: exemple de remplissage (Padding)

2.7 Le Modèle U-net

Le modèle U-Net (fig 2.15) est une architecture de réseau de neurones conçue pour la segmentation d'image en particulier dans le domaine médical.

Sa forme en ''U'' lui permet d'extraire les caractéristiques importantes tout en maintenant une bonne résolution spatiale grâce aux connexions entre les étapes d'encodage et de décodage

Objectif principal de U-Net :

Détecter les contours des objets (comme les MA) dans les images et segmenter ces images en parties significatives.

Structure générale : U-Net se distingue par sa forme en "U", qui permet :

- D'extraire les caractéristiques importantes (Encoder).
- De compresser ces informations (Bottleneck).
- De reconstruire une carte de segmentation détaillée avec ses dimensions d'origine et une segmentation claire (Décodeur). [11]

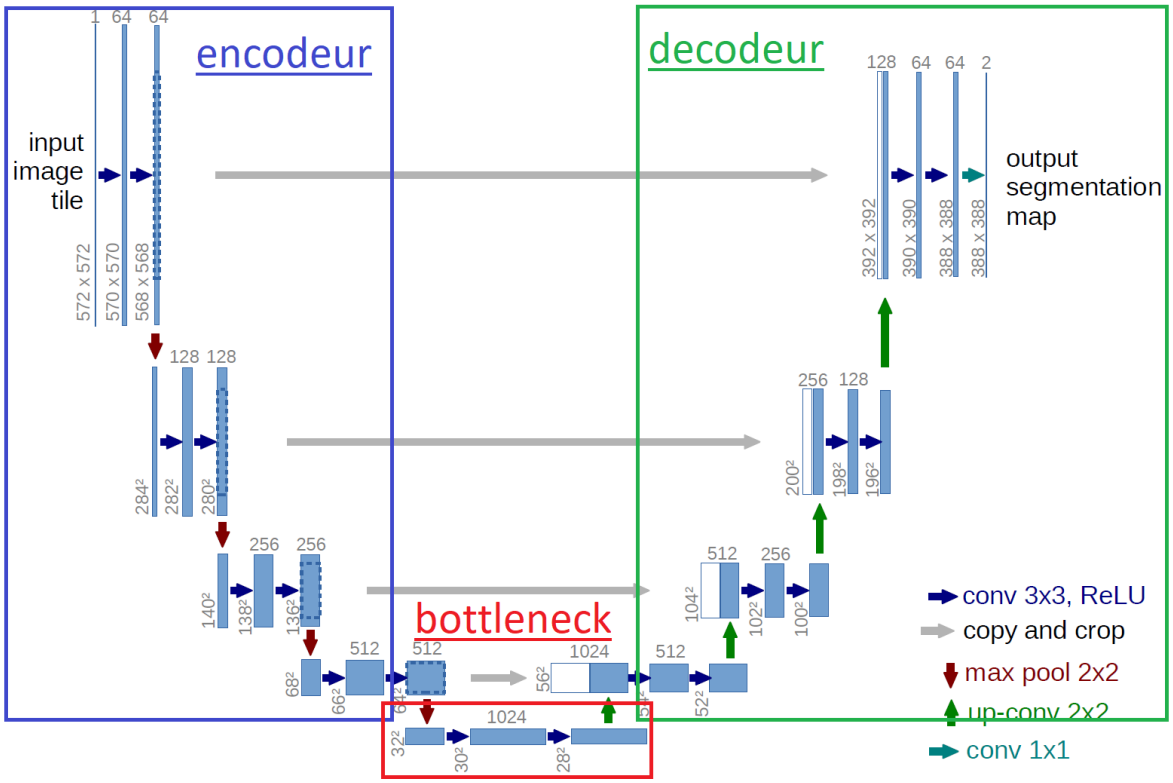


Figure 2.15: Visualisation de l'architecture U-Net

Le modèle est composé de trois parties principales :

2.7.1 L'encodeur

Rôle :

Partie qui réduit la dimension et extrait les caractéristiques importantes de l'image d'entrée (features maps) .

Etapes :

- Couches convolutives (Conv Layers) : des filtres de taille 3×3 sont appliqués pour extraire les caractéristiques.

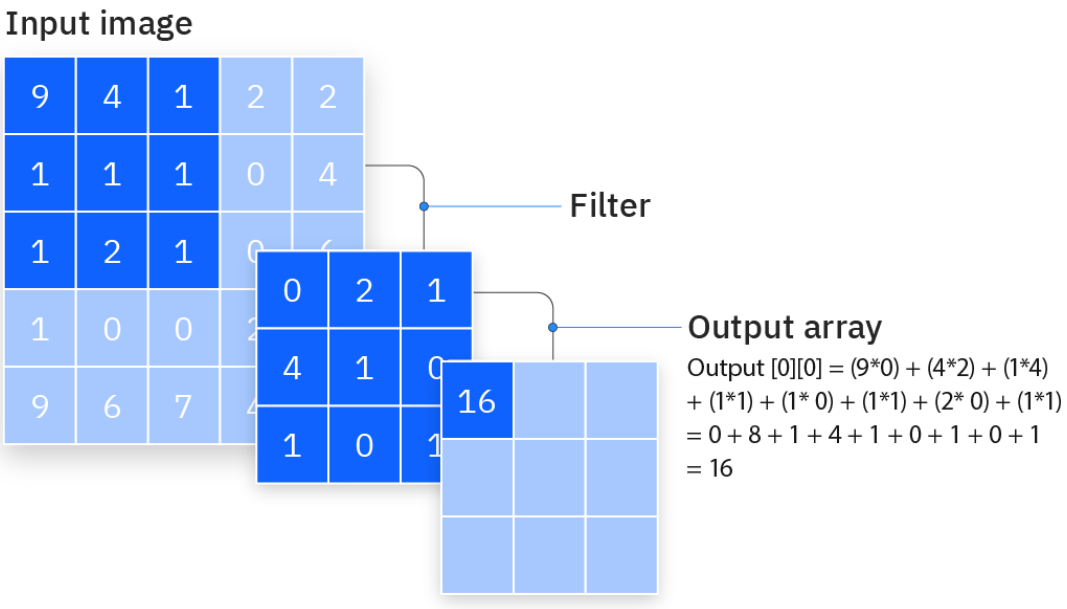


Figure 2.16: Couches convolutives [18]

- ReLU : pour introduire de la non-linéarité.
- Max Pooling : pour réduire les dimensions spatiales et augmenter la concentration des informations (conserve des informations lourdes).

Sortie :

Un ensemble de cartes de caractéristiques (Feature Maps) réduites en taille mais riches en informations. [11]

2.7.2 Bottleneck

Rôle :

Section intermédiaire du U-Net (transition entre l'Encodeur et le Décodeur), où l'information est compressée au maximum. Elle agit comme un pont entre l'encodeur et le décodeur, capturant les caractéristiques les plus significatives tout en perdant les détails les moins pertinents.

Caractéristiques :

- Couches convolutionnelles profondes avec un grand nombre de filtres (1024 filtres).
- Retient uniquement les informations essentielles. [11]

2.7.3 Le décodeur

Le décodeur est le processus de reconstruction de l'image originale à partir des caractéristiques extraites par l'encodeur. Cela se fait à l'aide des couches de convolution transposée (Transposed Convolutions) et de la concaténation (Concatenation).

Étapes principales :

1/-Upsampling (convolution transposée) :

Augmentation des dimensions spatiales grâce à des convolutions transposées.

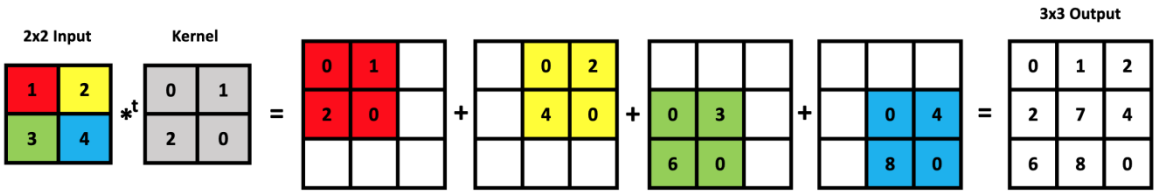


Figure 2.17: exemple de UpSampling [19]

2/-Skip Connections (concatenate) :

Combinaison des caractéristiques (features maps) du décodeur avec celles de l'encodeur correspondantes pour améliorer la précision.

Bien que cette opération ajoute une charge computationnelle, elle présente des avantages notables :

Avantage :

- L'augmentation du nombre de canaux signifie que le modèle reçoit des informations supplémentaires utiles (des détails fins de l'Encoder et des caractéristiques plus profondes du Décoder).
- Ces détails aident à améliorer la précision du processus de segmentation.

Défi :

- L'augmentation du nombre de canaux entraîne une consommation accrue de la mémoire (GPU Memory) et un nombre d'opérations plus élevé (Complexité computationnelle).

- Si le réseau est trop grand, vous pourriez rencontrer des problèmes de performance ou de mémoire.

Pour faire face à ces défis, U-Net applique certaines stratégies :

Utilisation de couches de Convolution après Concatenate :

- À chaque étape après la concaténation, une couche de convolution est utilisée pour réduire le nombre de canaux si nécessaire.
- Cela aide à atténuer l'augmentation du volume des données.

Équilibre entre les détails et la performance :

- Choisir un nombre de canaux plus petit dans l'Encoder peut réduire l'impact sur la charge computationnelle.

3/-Convolutions (Conv Layers) :

Réduire le nombre de canaux, de manière à "compresser" l'information à travers le filtrage. Seules les informations les plus importantes capturées par les filtres sont conservées. Pour affiner les cartes de caractéristiques.

Sortie :

Une carte de segmentation finale ayant les mêmes dimensions que l'image d'entrée. [11]

Résumé des étapes du U-Net :

- Entrée : Une image du Fond d'Œil – Rétinographie en 2D.
- Encodeur : Extraction progressive des caractéristiques tout en réduisant la taille.
- Bottleneck : Compression et traitement des caractéristiques.
- Décodeur : Reconstruction des dimensions et segmentation.
- Sortie : Une carte de segmentation montrant les régions segmentées.

Tableau 2.2: Description des différentes parties de l'architecture U-Net et leurs fonctions

Partie	Operations	Résultats
Encodeur	Conv-> ReLU->MaxPooling	Un ensemble de cartes de caractéristiques (Feature Maps) réduites en taille mais riches en informations
Bottleneck	Conv->ReLU	Une représentation compacte et dense des caractéristiques.
Décodeur	UpSampling->SkipConnections-> Conv->ReLU	Une carte de segmentation finale ayant les mêmes dimensions que l'image d'entrée

2.8 Le modèle VGG-16

VGG-16 est l'un des modèles de réseaux de neurones convolutifs (CNN) les plus connus (fig2.18), utilise pour la classification d'images et l'extraction de caractéristiques.

Le chiffre 16 fait référence au nombre de couche contenant des paramètres appris (les poids).

Ces couches sont comme suit :

- 13 couches convolutives
- 3 couches fully connected

Ces dernières couches sont utilisées pour la classification.

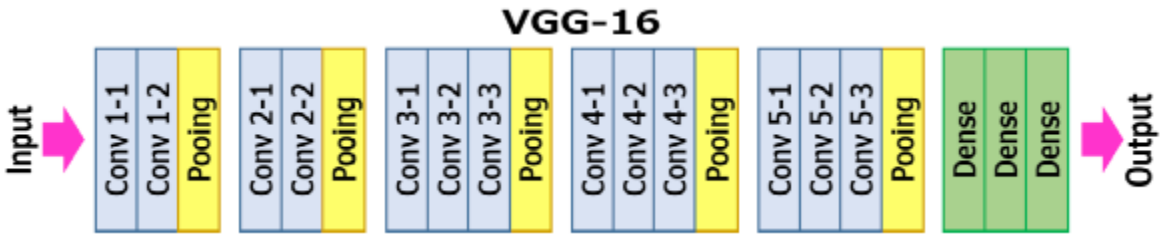


Figure 2.18: VGG-16 [20]

VGG16 peut être utilisé pour plusieurs applications, à savoir :

- **Classification d'images**
- **Extraction de caractéristiques (Feature) :** souvent utilisé sans les couches finales pour alimenter d'autres réseaux.
- **Apprentissage par transfert (Transfer Learning) :** réutilisation des couches pré-entraînées pour accomplir de nouvelles tâches.
- **Segmentation :** utilisé comme base dans des architectures comme U-Net. [21]

2.9 Transfert Learning

L'apprentissage par transfert (Transfer Learning) [fig.2.19] est une technique d'apprentissage automatique qui consiste à utiliser un modèle préalablement entraîné sur une tâche source pour améliorer l'apprentissage sur une tâche cible différente mais liée.

Cette approche permet de réduire le besoin en grandes quantités de données annotées pour la tâche cible et accélère le processus d'entraînement. [22]

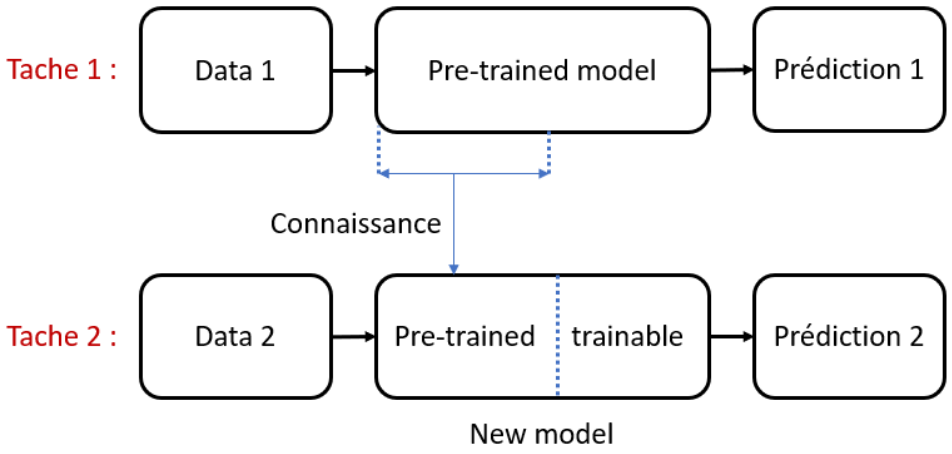


Figure 2.19: transfert learning

2.10 Comparaison de quelques architectures de deep Learning

Tableau 2.3: Comparaison des principales architectures de deep Learning pour la segmentation d'images médicales

Architecture	Avantages	Inconvénients
U-Net	Spécialement conçu pour les tâches de segmentation, performant sur les images médicales même avec peu de données.	Peut avoir des difficultés à détecter des détails très fins dans des images complexes.
Mask RCNN	Permet une segmentation précise des régions pathologiques, prend en charge les images couleur et multicouches.	Nécessite plus de ressources informatiques et un grand volume de données pour l'entraînement.
VGG-16 + U-Net	Combine la précision de VGG-16 dans l'extraction des caractéristiques et la puissance de U-Net en segmentation.	VGG16 est volumineux et consomme beaucoup de mémoire.
DeepLabV3+	Utilise des techniques avancées comme les Atrous Convolutions, permettant une détection à différentes échelles.	Relativement complexe et potentiellement excessif pour ce type de tâche.

2.11 Matrice de confusion dans le cadre d'une segmentation binaire

Dans le contexte de la segmentation binaire, comme dans l'identification automatique des zones pathologiques dans une image de fond d'œil (ex. micro-anévrismes liés à la rétinopathie diabétique), la matrice de confusion est un outil essentiel pour évaluer les performances d'un modèle d'apprentissage profond tel que U-Net. [26]

Elle permet de comparer, pixel par pixel, les prédictions du modèle à la vérité terrain (masques annotés). Chaque pixel est classé comme « pathologique » (positif) ou « sain » (négatif).

Grâce à cette matrice, on peut calculer des indicateurs de performance précis :[27]

- Accuracy (taux de bonne classification) : proportion totale de pixels correctement prédits.
- Précision : fiabilité des pixels classés comme pathologiques.
- Rappel (Recall ou Sensitivity) : capacité à détecter tous les pixels réellement pathologiques.
- F1-Score , Dice Score : mesure harmonique entre précision et rappel, utile en cas de classes déséquilibrées.

Tableau 2.4: Exemple concret de TP, TN, FP, FN

Abréviation	Terme complet	Signification
TP	Vrai positif (True Positive)	Le pixel est pathologique, et le modèle le prédit comme tel
TN	Vrai négatif (True Negative)	Le pixel est sain, et le modèle le prédit comme tel
FP	Faux positif (False Positive)	Le modèle prédit le pixel comme pathologique alors qu'il est en réalité sain
FN	Faux négatif (False Negative)	Le modèle prédit le pixel comme sain alors qu'il est en réalité pathologique

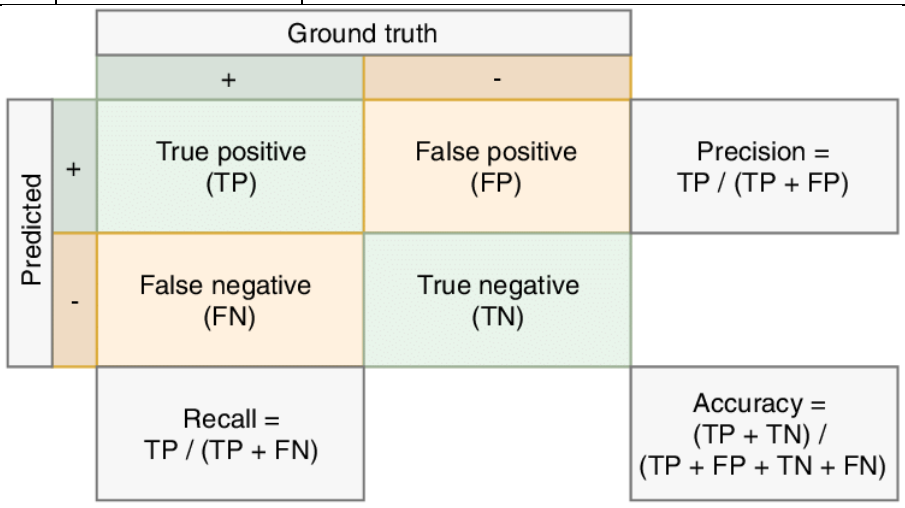


Figure 2.20 : Un exemple de matrice de confusion [28]

À partir de ces quatre quantités (TP, TN, FP, FN), on peut calculer :

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Précision

$$Precision = \frac{TP}{TP + FP}$$

Recall

$$Recall = \frac{TP}{TP + FN}$$

F1-score

$$F1score = \frac{2 \times precision \times Recall}{precision + Recall} = Dice\ Score$$

2.12 Conclusion

Dans ce deuxième chapitre, nous avons présenté différents types de réseaux neuronaux, en décrivant leur fonctionnement théorique ainsi que leurs caractéristiques pour les besoins de la segmentation et de l'analyse d'images.

Dans le troisième chapitre, nous choisissions U-Net et nous procédons à mise en place d'une architecture de Deep Learning, à la constitution d'une base d'apprentissage pour la segmentation de la RD, et la conception d'une architecture U-Net-VGG16 pour l'utilisation du transfert Learning.

Chapitre 3 :

Conception d’une architecture U-Net pour la segmentation des MA, et d’un dispositif optique expérimental

3 Chapitre 3 Conception d’une architecture U-Net pour la segmentation des MA, et d’un dispositif optique expérimental

3.1 Introduction

Dans ce chapitre, nous décrivons la conception détaillée de notre modèle de segmentation des micro-anévrismes basé sur l’apprentissage profond. Nous présentons la méthodologie adoptée, qui combine une architecture U-Net enrichie par un transfert learning via VGG16.

Nous détaillons également les étapes de création, de prétraitement, d’augmentation des données, ainsi que la préparation et la division de la base d’images. Enfin, nous abordons la conception du système optique de projection laser contrôle par l’image segmentée et assisté par DMD, synchronisé avec les signaux ECG, afin de réaliser la photocoagulation multispot ciblée.

3.2 Contexte de notre projet

Le projet s’inscrit dans le cadre du projet de recherche de l’équipe de recherche biorétine du laboratoire Latsi de l’université de Blida, il a pour objectif la conception et réalisation d’un photocoagulateur laser multispot embarqué connecté basé sur des micromiroirs.

Notre travail vise à développer un modèle d’apprentissage profond pour la détection et la segmentation automatique des micro-anévrismes rétiniens. Un dispositif optique basé sur la technologie DMD a été développé afin de valider le principe de la photocoagulation laser multispot.

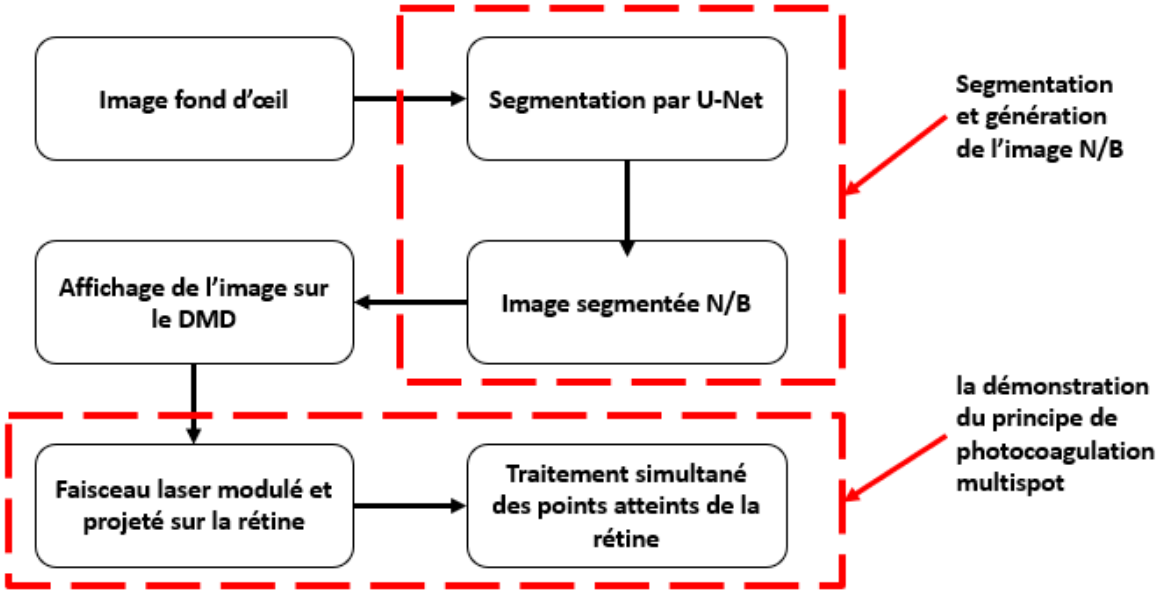


Figure 3.1: Schéma synoptique du contexte de notre projet

3.3 Objectifs du projet

Ce projet vise à développer une approche innovante et intégrée pour le diagnostic assisté et le traitement ciblé de la rétinopathie diabétique à travers deux axes principaux et complémentaires :

a) Conception d'une méthode de Deep Learning améliorée autour de U-Net

Le premier objectif consiste à développer un système performant de segmentation automatique des micro-anévrismes à partir d'images du fond d'œil. Pour cela :

- Une architecture U-Net modifiée est mise en place, intégrant le modèle VGG-16 préentraîné dans le cadre du transfert learning, afin d'exploiter des représentations visuelles de haut niveau.
- Une stratégie d'augmentation des données est appliquée, spécifiquement adaptée à la nature localisée et de petite taille des micro-anévrismes (la division des images en sous-blocs).
- Un pipeline complet de traitement d'images est conçu, incluant le redimensionnement, la normalisation, et la suppression du canal rouge.
- L'entraînement et la validation du modèle sont réalisés à partir d'une base de données annotée manuellement, validée par une spécialiste en ophtalmologie, garantissant la qualité des régions d'intérêt.

b) Mise en œuvre d'une expérience optique utilisant un DMD pour la démonstration du principe de photocoagulation multispot

Le second objectif sert à démontrer comment la segmentation obtenue peut être utilisée pour traiter simultanément plusieurs micro-anévrismes par photocoagulation laser ciblée, à l'aide d'un Dispositif à Micromiroirs Numérique (DMD).

- L'image binaire résultant de la segmentation est transmise au DMD afin de moduler un faisceau laser en un motif correspondant aux régions pathologiques.
- Le système optique est conçu pour projeter le faisceau façonné avec précision sur une rétine artificielle, cible simulée du traitement réel.
- Cette démonstration vise à valider la faisabilité d'un système intelligent de photocoagulation multispot, capable de réduire le temps d'intervention, minimiser l'inconfort du patient, et améliorer la précision thérapeutique.

3.4 Méthode d'apprentissage profond améliorée basée sur U-Net

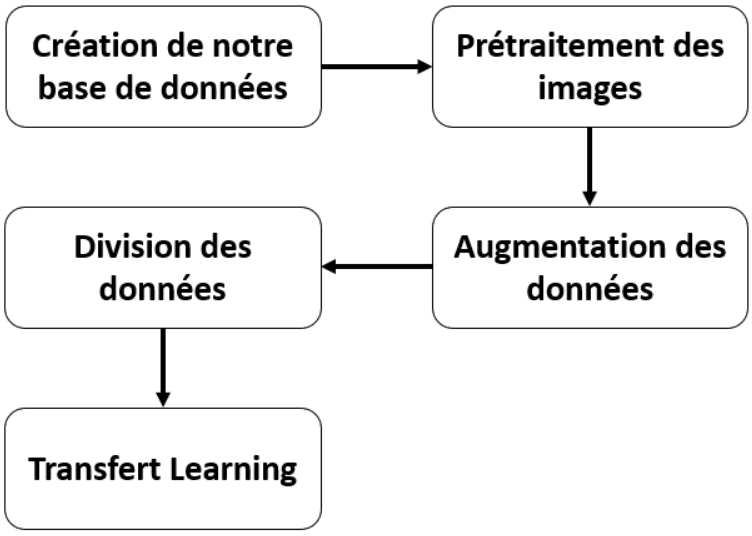


Figure 3.2: Architecture synoptique de la méthode U-Net améliorée

3.4.1 Base de données (Annotation médicale)

L'identification précise des micro-anévrismes constitue une étape déterminante dans la construction d'un modèle de segmentation efficace. Dans ce cadre, une collaboration a été établie avec une spécialiste en ophtalmologie (fig3.3), Dr. LAMIA DAHAM, en charge d'identifier et d'annoter manuellement les régions d'intérêt sur les images du fond d'œil.

Figure 3.3:Chirurgienne ophtalmologiste ayant contribué à la création et à la validation de notre base de données

Un total de 141 images rétiniennes a été capturé pour constituer une base de données dédiées à ce projet. Deux outils complémentaires ont été mobilisés pour l'annotation :

- PicsArt : application mobile d'édition d'images, utilisée pour effectuer des annotations rapides grâce à des outils de dessin simples.
- Apeer : plateforme en ligne spécialisée dans le traitement et l'annotation des images scientifiques, offrant un cadre structuré et précis pour les annotations médicales.

Les annotations réalisées ont toutes été supervisées et validées par l'experte médicale, assurant la fiabilité des données d'apprentissage, indispensables pour l'entraînement supervisé des modèles d'intelligence artificielle.

Enfin, l'organigramme suivant (fig3.4) illustre la méthode de création des masques à l'aide de PicsArt :

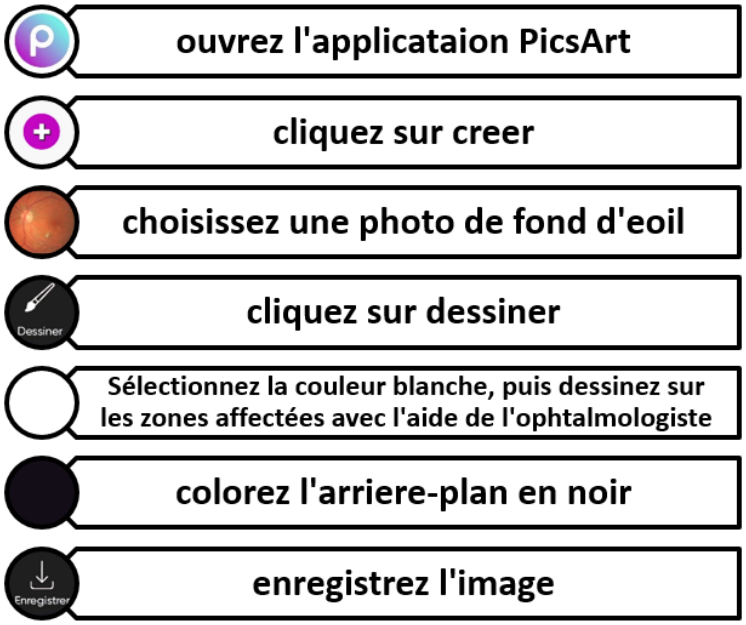


Figure 3.4: Organigramme de la méthode de création de masques avec PIcsArt à partir d'images du fond d'œil

Voici la hiérarchie des dossiers contenant les images et masques utilises (fig 3.5) :

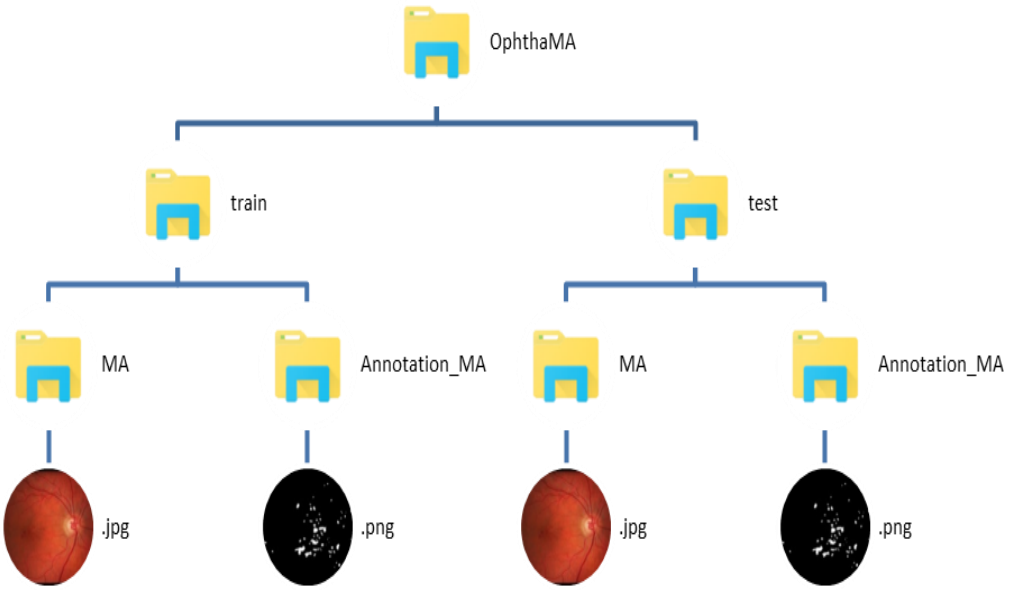


Figure 3.5: Division de la base de données

3.4.1.1 Caractéristiques des images originales en format .jpg

- Compression efficace : Le format JPG utilise une compression avec perte qui réduit la taille des fichiers tout en maintenant une qualité acceptable.
- Adaptation aux images naturelles : Idéal pour les photographies et les images médicales, où les variations de couleur sont fluides et peu affectées par la compression.

- Traitement plus rapide : Grâce à sa taille réduite, il permet un chargement et un traitement plus rapides pendant l'entraînement du modèle.

3.4.1.2 Caractéristiques des annotations enregistrées en format .png

- Sans perte de données (Lossless) : Le format PNG conserve tous les détails sans altération due à la compression, ce qui est essentiel pour les masques de segmentation.
- Support de la transparence : Peut être utile pour gérer différentes couches d'annotations.
- Nombre limité de couleurs : Les annotations sont souvent des images binaires (Binary Masks) ou multi-classes (ex : 0 pour l'arrière-plan, 1 pour les régions atteintes), et PNG prend en charge ces images sans ajouter de bruit, contrairement à JPG.

3.4.2 Prétraitement des images

Avant l'entraînement du modèle U-Net, un ensemble d'opérations de prétraitement a été appliqué afin d'uniformiser les données, d'améliorer la qualité des images et de faciliter l'extraction des caractéristiques pertinentes. Ce processus comprend les étapes suivantes :

3.4.2.1 Redimensionnement

Dans un réseau de neurones convolutionnel tel que U-Net, la taille des images en entrée doit être uniforme pour permettre un traitement cohérent tout au long des couches du modèle.

Nous avons choisi une taille de 512×512 pixels, qui offre un bon compromis entre la préservation des détails pathologiques (notamment les micro-anévrismes très petits) et les contraintes de mémoire GPU.

Ce redimensionnement a été appliqué non seulement aux images originales mais aussi aux masques annotés par le médecin.

3.4.2.2 Normalisation

Pour assurer une convergence rapide du modèle pendant l'entraînement, toutes les images ont été normalisées dans l'intervalle [0, 1] en divisant les valeurs de pixels par 255. Cela permet de stabiliser les gradients et d'accélérer l'apprentissage.

3.4.2.3 Suppression du canal rouge

L'étude des spécifications techniques du système iCare DRSplus montre que les images du fond d'œil sont capturées en mode confocal trichrome (Rouge, Vert, Bleu – R, G, B), avec une option permettant l'acquisition sans le canal rouge. Ce mode améliore significativement la visualisation des structures vasculaires rétiniennes, tout en offrant une meilleure visibilité de la couche des fibres nerveuses (RNFL) grâce à la lumière bleue. En effet, le canal rouge permet à la lumière de pénétrer plus profondément dans les couches de la rétine, tandis que l'éclairage infrarouge apporte des informations précieuses sur la choroïde. [5]

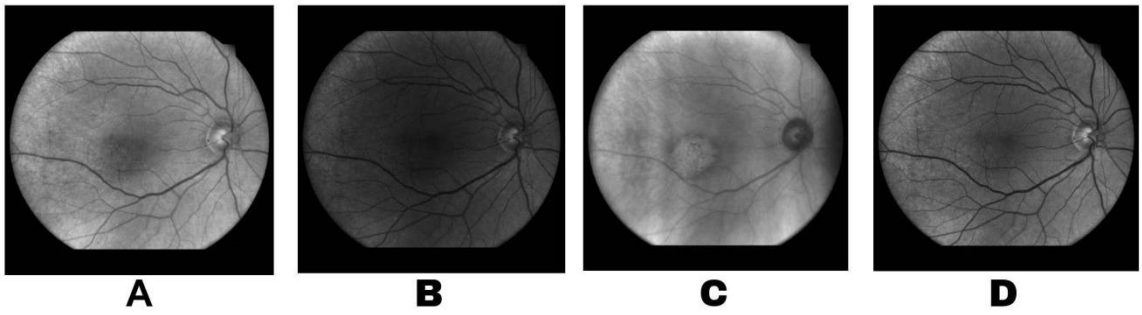


Figure 3.6: Illustration des différentes modalités d'imagerie confocale TrueColor pour l'analyse rétinienne (A:Rouge, B:Blue, C:Infra-rouge, D:Sans rouge) [5]

Dans le cadre de notre projet, et afin d'améliorer la visibilité des vaisseaux sanguins rétiniens dans notre base de données personnalisée, nous avons supprimé le canal rouge des images. Cette démarche renforce le contraste des structures vasculaires et de l'arrière-plan rétinien, ce qui est particulièrement utile pour détecter les micro-anévrismes, petites dilatations localisées des capillaires.

La suppression du canal rouge a été réalisée à l'aide d'un script Python selon les étapes suivantes (fig3.7) :

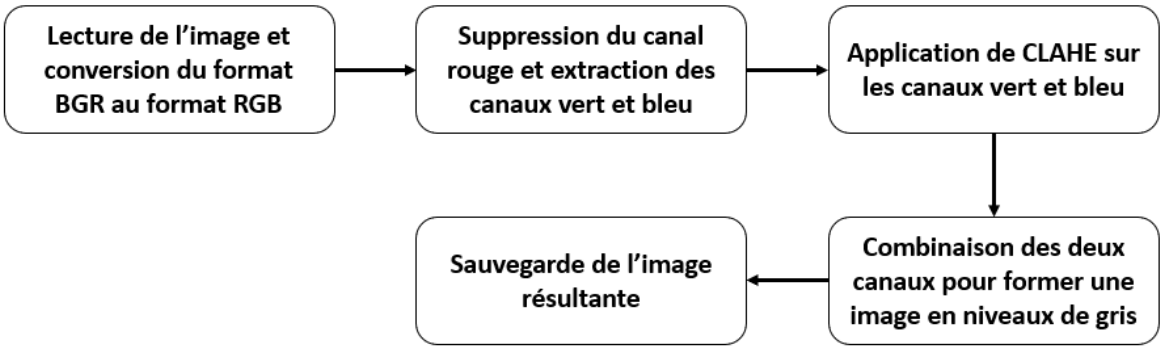


Figure 3.7: Organigramme des étapes de suppression du canal rouge dans une image

1/ Lecture de l'image et conversion du format BGR au format RGB

Les images sont lues à l'aide de la bibliothèque OpenCV, qui utilise par défaut l'ordre de canaux BGR. Pour assurer une représentation correcte des couleurs, une conversion vers le format RGB est effectuée. Cela permet une visualisation et un traitement plus cohérents avec d'autres bibliothèques de traitement d'image.

2/ Suppression du canal rouge et extraction des canaux vert et bleu

Une fois l'image convertie en RGB, la composante rouge est supprimée (mise à zéro), tandis que les composantes verte et bleue sont conservées pour une utilisation ultérieure. Cette étape vise à éliminer les informations provenant des couches profondes de la rétine et à se concentrer sur les structures plus superficielles comme les vaisseaux.

3/ Application de CLAHE sur les canaux vert et bleu

CLAHE (Contrast Limited Adaptive Histogram Equalization) est appliqué séparément aux canaux vert et bleu pour améliorer localement le contraste. Cette technique divise l'image en petites zones (tiles), applique une égalisation d'histogramme locale dans chaque zone, et limite l'amplification du contraste pour éviter d'accentuer le bruit. Cela permet de révéler plus de détails dans les zones peu contrastées.

4/ Combinaison des deux canaux pour former une image en niveaux de gris

Les deux canaux traités (vert et bleu) sont combinés en une image monochrome en calculant la moyenne de leurs intensités. Cela donne une image en niveaux de gris

améliorée, où les détails des structures vasculaires sont mieux mis en évidence sans interférence de la composante rouge.

5/ Sauvegarde de l'image résultante

L'image finale, améliorée et sans canal rouge, est enregistrée pour être utilisée dans les étapes suivantes de traitement et d'analyse, notamment pour la détection des micro-anévrismes et autres anomalies liées à la rétinopathie diabétique.

3.4.3 Augmentation des données

Il existe de nombreuses techniques d'augmentation des données. Certaines ont été choisies et sont présentées dans la figure 3.8.

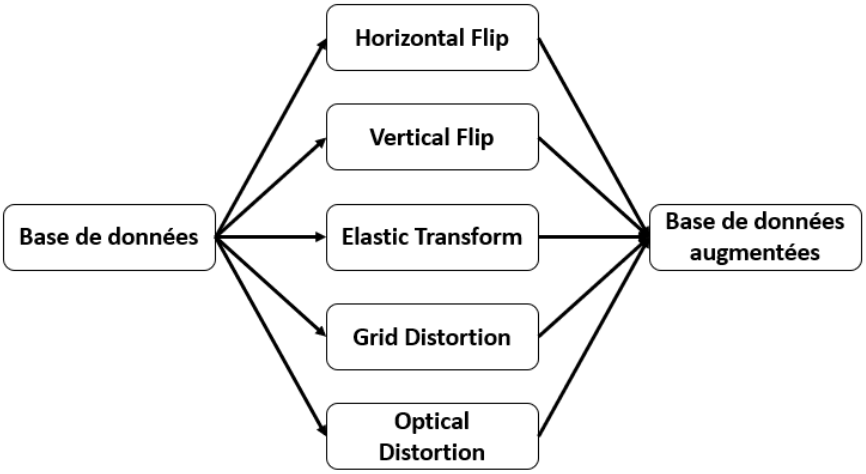


Figure 3.8: Schéma synoptique du principe de la technique d'augmentation des données

Contrairement aux approches classiques d'augmentation (rotation, flip, zoom…), nous avons adopté une stratégie plus fine et ciblée, adaptée à la nature des micro-anévrismes, qui sont des lésions très petites par rapport à la taille globale de l'image du fond d'œil.

Notre méthode consiste à diviser chaque image de fond d'œil en 16 sous-images (par découpage en blocs de 128×128 pixels) (fig3.9).

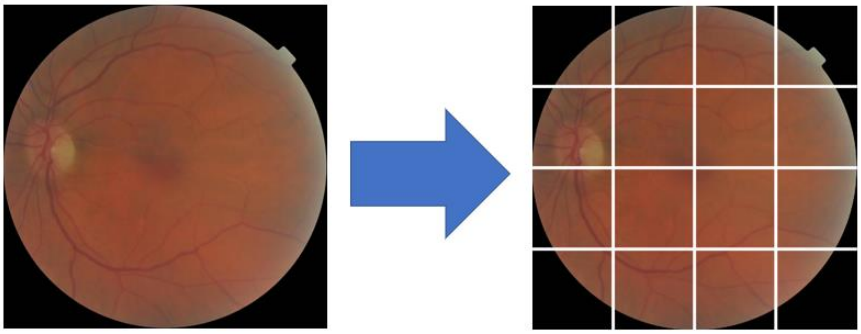


Figure 3.9:Exemple de division d'une image en 16 sous-images (4×4)

Cette approche permet :

- De centrer l'apprentissage sur des zones locales contenant potentiellement des anomalies.
- D'augmenter artificiellement la taille du jeu de données sans générer de transformations artificielles.
- De faciliter la détection des MA en réduisant le rapport de taille entre la lésion et l'image.

Cela permet d'entraîner un modèle plus sensible aux structures fines, comme les micro-anévrismes, qui peuvent facilement passer inaperçus dans des images de grande taille.

3.4.4 Chargement et division des données

Les données sont divisées en trois ensembles (fig3.10) principaux :

- Ensemble d'entraînement : Il sert à apprendre au modèle à reconnaître des motifs en ajustant ses paramètres internes (poids et biais).
- Ensemble de validation : Il sert à optimiser l'entraînement du modèle en évaluant ses performances sur des données qu'il n'a pas vues.
- Ensemble de test : Il sert à évaluer la performance finale du modèle sur des données totalement nouvelles, comme si c'était en conditions réelles.

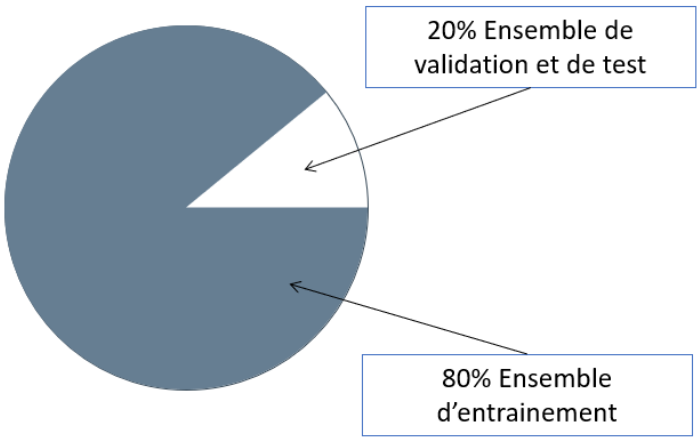


Figure 3.10: Division des données en ensembles d'entraînement, de validation et de test

3.4.5 Générateur de données

Le générateur de données constitue un composant fondamental pour l'entraînement efficace d'un modèle de segmentation, en particulier dans un contexte où la base de données est volumineuse et personnalisée.

Nous avons développé un générateur personnalisé en Python, avec les fonctionnalités suivantes (fig3.11) :

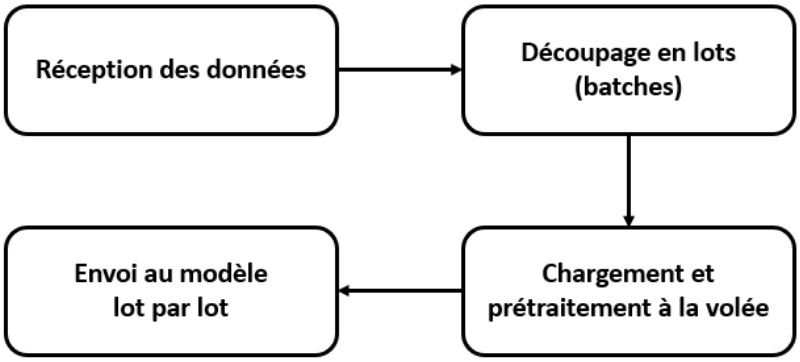


Figure 3.11:Schéma fonctionnel du flux de données dans le DataGenerator

- Chargement dynamique des images et de leurs masques pendant l'entraînement, sans avoir besoin de tout stocker en mémoire, ce qui est particulièrement crucial pour les grandes bases de données.
- Prétraitement automatique des images (redimensionnement, normalisation, et adaptation des dimensions des masques pour la compatibilité avec le modèle).

- Retour de lots d'images et masques prêts à l'entraînement sous forme de tableaux numpy.
- Amélioration de la vitesse et de la stabilité de l'entraînement

3.4.6 Transfert Learning

Le transfert Learning (fig.3.8) (ou apprentissage par transfert) est une technique qui consiste à réutiliser un modèle pré-entraîné sur une large base de données (comme ImageNet) pour l'adapter à une nouvelle tâche spécifique, souvent avec un volume de données limité. Cette approche est particulièrement utile dans le domaine médical, où l'obtention de grandes bases annotées est coûteuse et complexe.

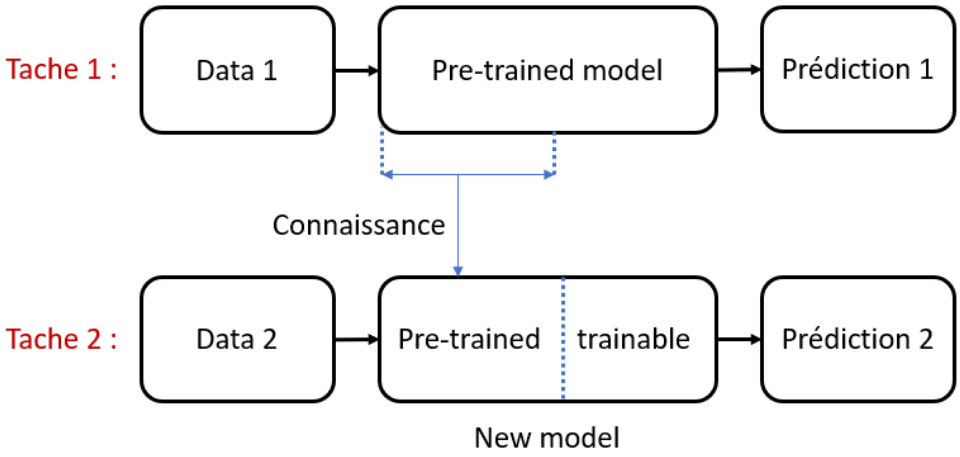


Figure 3.12: transfert Learning

3.4.6.1 Utilisation de VGG16 avec U-Net

Afin d'améliorer la capacité d'extraction des caractéristiques visuelles (features) à partir des images de la rétine, nous avons intégré le modèle VGG16 pré-entraîné dans l'architecture U-Net.

VGG16 est un réseau de neurones convolutifs profond, largement utilisé pour la classification d'images, notamment en raison de sa structure simple et efficace. En l'intégrant comme encodeur dans U-Net, nous bénéficions de poids pré-entraînés sur ImageNet, ce qui permet un transfert de connaissance utile, même dans le contexte d'images médicales.

Cette approche améliore la performance du modèle sur des jeux de données de taille limitée, comme c'est souvent le cas en médecine.

Cette intégration présente plusieurs avantages majeurs :

- **Vitesse d'apprentissage accrue :** les couches étant déjà entraînées, le modèle ne part pas de zéro.
- **Meilleure précision :** les caractéristiques extraites par VGG16 sont profondes et pertinentes.
- **Efficace avec peu de données :** l'apprentissage par transfert (Transfer Learning) réduit le besoin d'un grand volume de données.

Ainsi, en combinant la robustesse de VGG16 avec la structure efficace de U-Net pour la segmentation, nous obtenons un modèle performant et adapté aux exigences du domaine médical.

3.4.6.2 Comment intégrer VGG16 dans U-Net

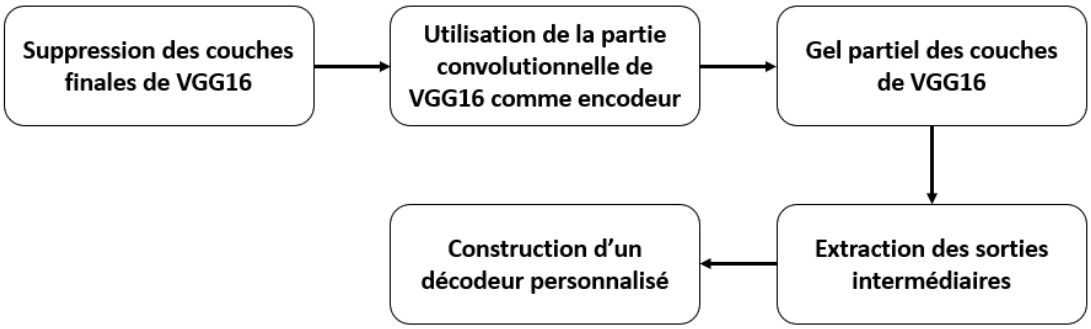


Figure 3.13: organigramme d'intégration de VGG16 Dans l'architecture U-Net

L'intégration du modèle VGG16 dans l'architecture U-Net suit les étapes suivantes:

- Suppression des couches finales de VGG16 : Les couches supérieures du réseau VGG16 sont supprimées, car elles sont conçues pour la classification en 1000 classes (ImageNet), ce qui n'est pas adapté à une tâche de segmentation comme la détection des micro-anévrismes (MA).
- Utilisation de la partie convolutionnelle de VGG16 comme encodeur : Toutes les couches convolutionnelles du VGG16 sont conservées pour extraire des caractéristiques riches. Ces couches fournissent une représentation hiérarchique de l'image, capturant des détails fins comme les contours et les textures.

- Gel partiel des couches de VGG16 : Les premières couches de VGG16 sont figées (non entraînées), car elles apprennent des caractéristiques générales communes à toutes les images, telles que les bords, les lignes et les contrastes.
- Extraction des sorties intermédiaires : Les sorties intermédiaires de chaque bloc convolutionnel sont sauvegardées pour être utilisées dans les connexions de saut (Skip Connections), élément essentiel de l'architecture U-Net. Ces connexions permettent de préserver des informations spatiales fines qui peuvent être perdues dans les couches profondes.
- Construction d'un décodeur personnalisé : La partie décodeur est construite de manière symétrique, en combinant des opérations d'upsampling et de concaténation avec les sorties intermédiaires issues de VGG16, afin de reconstruire une carte de segmentation précise.

Après avoir appliqué plusieurs modifications au modèle U-Net original, notamment l'amélioration du prétraitement, l'augmentation des données, ainsi que l'adaptation des couches aux caractéristiques des micro-anévrismes et à la taille de la base de données, nous présentons dans la figure 3.14 l'architecture finale proposée de notre modèle amélioré

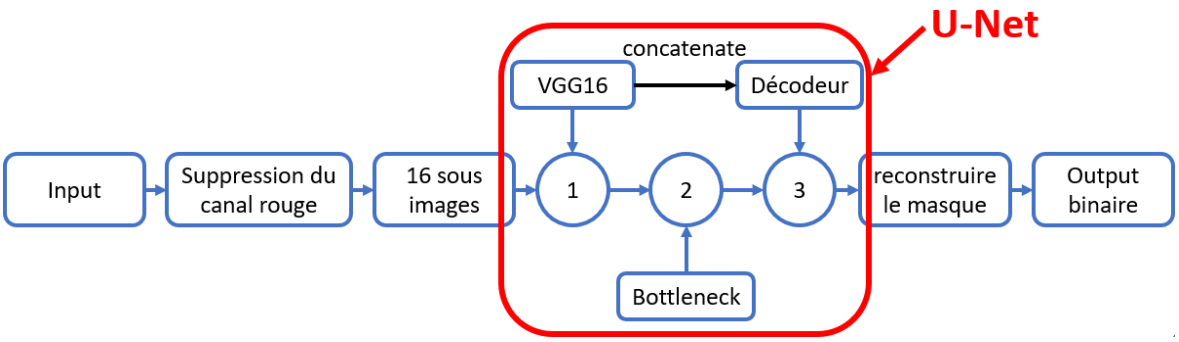


Figure 3.14: Architecture de notre modèle

3.5 Conception d'un dispositif optique de projection laser

Le système optique a été conçu et testé à l'aide d'un montage physique réel. L'objectif initial était de réaliser une simulation de photocoagulation laser multispot de haute précision, ciblant les micro-anévrismes détectés automatiquement.

3.5.1 Description du fonctionnement du système

Le système développé combine une méthode de segmentation automatique par Deep Learning et un montage optique réel pour simuler une photocoagulation multispot précise sur la rétine (fig3.15).

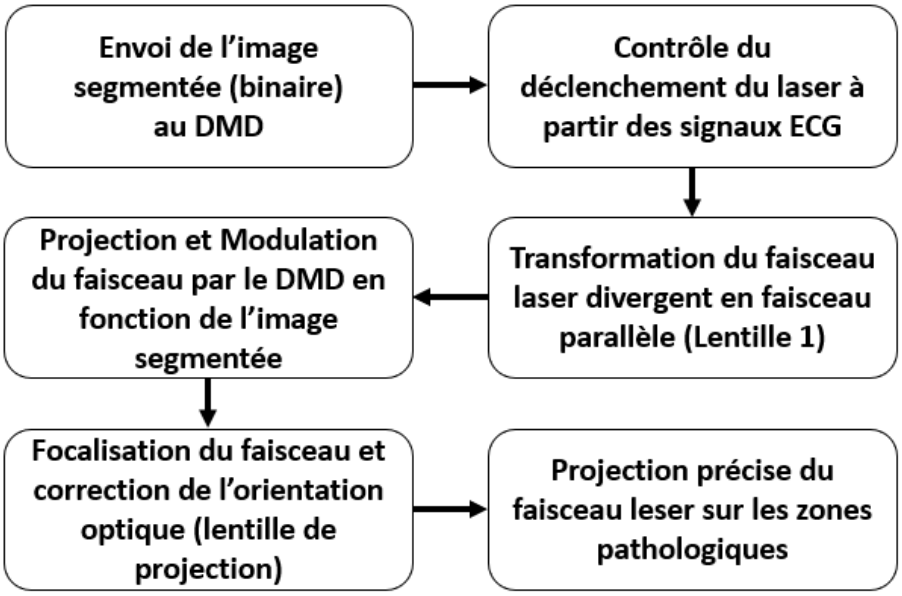


Figure 3.15: schéma fonctionnel du système de projection laser optique

1) Envoi de l'image binaire au DMD

L'image binaire segmentée est transférée directement au DMD DLP7000 à l'aide du logiciel ALPDemo.

- Les pixels blancs correspondent aux zones atteintes par les micro-anévrismes.
- Les pixels noirs représentent les régions saines.

Avant l'envoi, l'image est inversée horizontalement (gauche-droite) et verticalement (haut-bas). Ce renversement est nécessaire, car après réflexion par le DMD et passage à travers la lentille 2, l'image subit une seconde inversion optique, ce qui permet de retrouver l'orientation correcte de l'image projetée sur l'écran (ou la rétine dans une version clinique).

Ainsi, la symétrie initiale garantit que les zones pathologiques segmentées sont correctement alignées dans la projection finale.

2) Formation et façonnage du faisceau laser

Un laser du commerce est utilisé comme source lumineuse. Le chemin optique est organisé de la manière suivante :

a. Source laser → Lentille 1 à une distance de 25 cm

Cette lentille convergente de focale 25 cm (puissance optique de 4 dioptries) permet le collimatage de la source laser et rend le faisceau parallèle.

b. Lentille 1 → DMD

Le faisceau parallèle atteint perpendiculairement le DMD qui réfléchit les points blancs de l'image à +12° et les points noirs à -12°. La cible est donc placée dans l'axe de +12°.

c. DMD → Lentille 2 (à 25 cm)

Une deuxième lentille est placée pour reconcentrer ou diffuser le faisceau selon l'objectif :

- Lentille de focalisation : pour simuler la projection concentrée du faisceau sur la rétine.
- Lentille de diffusion : pour élargir la projection sur un écran de visualisation dans les tests.

d. Lentille 2 → Écran de simulation (à 25 cm)

La lumière projetée forme une image lumineuse des zones ciblées.

3) Déclenchement du laser par signaux ECG

Ce système d'activation du laser a été mis en place à partir des signaux (ECG), permettant une commande ON/OFF contrôlée. Ce système repose sur un seuil prédéfini (fig 3.16), paramétré par notre collègue Yousra, afin d'assurer que le laser ne s'active que lorsque les conditions d'alignement et de sécurité sont remplies. L'objectif est de garantir un fonctionnement sécurisé et précis du laser, en réponse aux mouvements oculaires détectés, tout en minimisant les risques d'erreur ou d'activation involontaire.

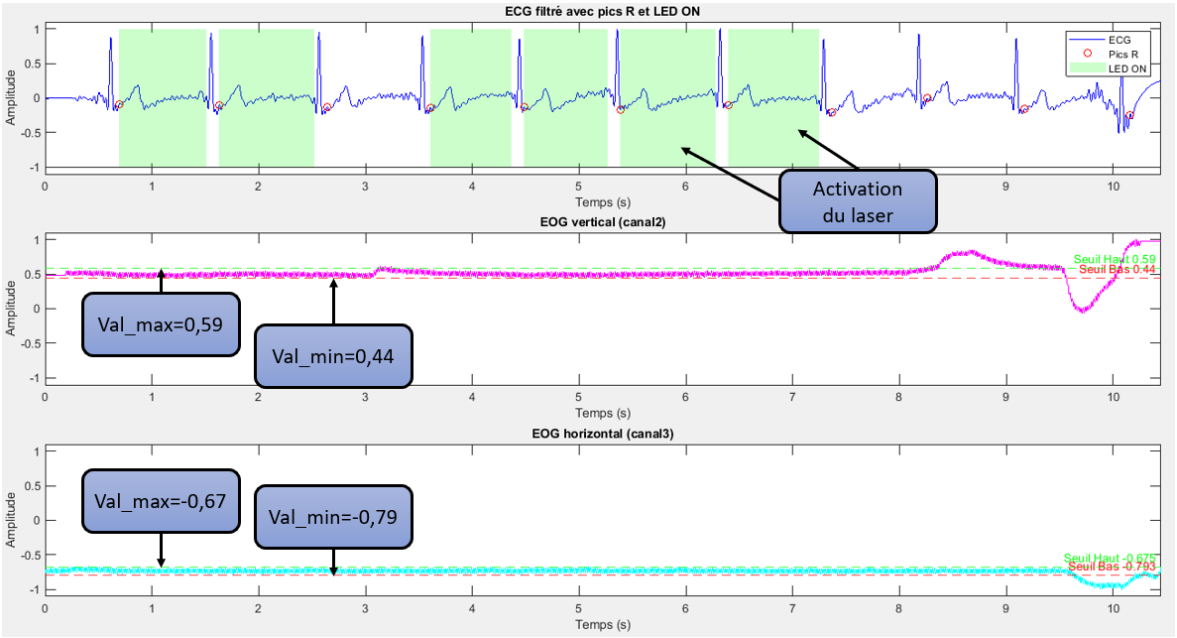


Figure 3.16: contrôle de laser à travers les signaux ECG et EOG :au-delà d'une certaine amplitude cardiaque, des pics EOG sont observés

La figure 3.17 illustre schéma finale du système de traitement laser assisté que nous avons développé, combinant les résultats de la segmentation automatique obtenus par Deep Learning avec un dispositif optique physique basé sur un DMD pour la photocoagulation multispot ciblée des micro-anévrismes, ainsi qu'un mécanisme de contrôle du déclenchement du laser par des signaux ECG afin d'assurer un fonctionnement sécurisé et synchronisé.

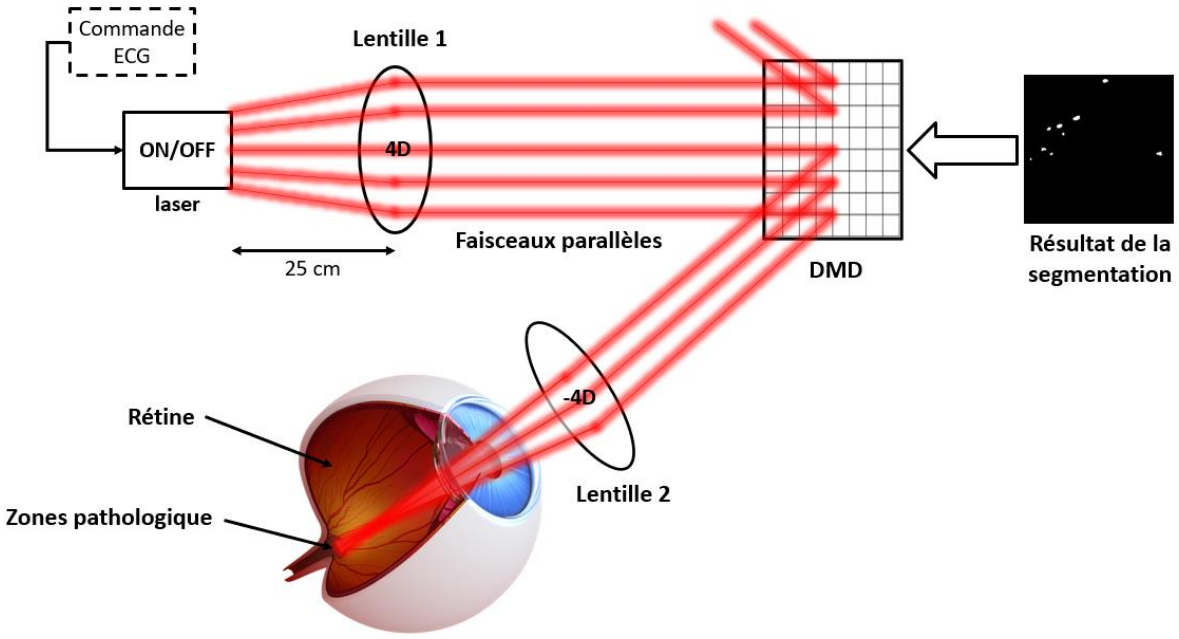


Figure 3.17: schéma du système de traitement laser rétinien base sur DMD

3.5.2 Les paramètres

Lentille 1 :

Pour définir les caractéristiques du système optique, le tableau 3.1 et 3.2 présente les paramètres choisis de la lentille 1 et 2 ainsi que les raisons de leur sélection.

Tableau 3.1: Choix des paramètres optiques pour la projection du faisceau sur le DMD (Lentille 1)

Paramètres	Valeur choisie	Justification
Focale (f)	f=25 cm (0.25m)	Pour obtenir un faisceau parallèle, on place la source a la distance focale
Puissance optique (P)	4 D	$P = \frac{1}{f} = \frac{1}{0.25} = 4D$ (Dioptries)
Type de lentilles	Sphérique convergente	Rendre parallèle et concentrer le faisceau
Diamètre	≥ 25 mm	Peut contenir l'ensemble du faisceau et est disponible chez l'opticien
Matériau	Verre ou plastique optique	Bonne qualité de transmission

Lentille 2 :

Tableau 3.2: Choix des paramètres optiques pour la projection du faisceau sur le DMD (Lentille 2)

Paramètres	Valeur choisie	Justification
Focale	f= -25cm	Pour obtenir un faisceau divergent
Puissance optique	-4 D	$P = \frac{1}{f} = \frac{1}{-0.2} = -4D$ (Dioptries)
Type de lentilles	Biconcave	Nécessaire pour diverger un faisceau parallèle
Diamètre	>= 25mm	Peut contenir l'ensemble du faisceau et est disponible chez l'opticien
Matériau	Verre ou plastique optique	Bonne qualité de transmission

La focale :

Chaque lentille possède une focale (f).

Lorsqu'on place le laser a cette distance focale (f), le faisceau sortant de la lentille devient parallèle (fig 3.18).

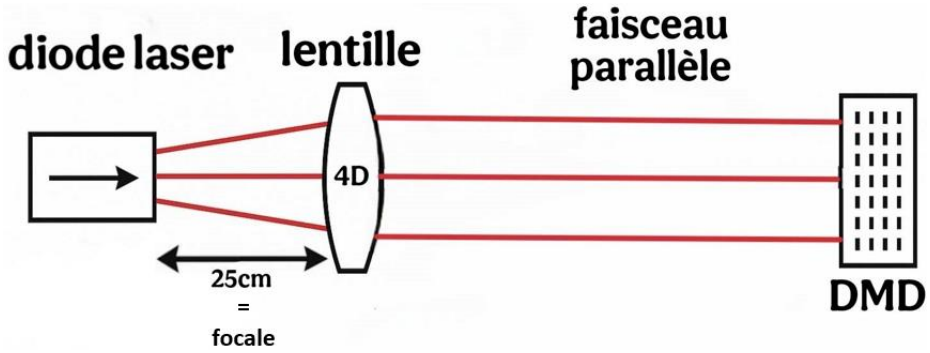


Figure 3.18: schéma de collimatage de la source laser

Ainsi, pour produire un faisceau parallèle, la position optimale du laser est à f=25cm.

Ce choix a été fait pour plusieurs raisons :

- Ne nécessite pas une grande distance
- Le type de lentille est disponible chez l'opticien
- Le système est équilibré et facile à réaliser

Afin de comparer les implications pratiques de différentes longueurs focales, le tableau 3.3 résume les avantages et les inconvénients d'une focale très longue et très courte.

Tableau 3.3: Impacts pratiques du choix d'une focale très longue ou très courte dans un montage optique

f très longue	f très courte
Nécessite un grand espace dans labo	Lentille très puissante
Stabilité difficile (vibration)	Très sensible

Dioptrie :

La dioptrie est l'unité de mesure de la puissance optique d'une lentille.

Définie par :

$$D = \frac{1}{f\ (en\ metres)}$$

Cela signifie que si l'on place un laser à une distance f devant la lentille, le faisceau sortira ensuite de manière parallèle.

Pour la fabrication des lentilles on utilise la formule suivante :

$$P = (n-1)(\frac{1}{R_1} - \frac{1}{R_2})$$

Où :

P : puissance optique (en dioptries)

n : indice de réfraction du matériau (verre=1.5)

R : rayons de courbure des surfaces

3.6 Conclusion

Dans ce chapitre, nous avons présenté la conception de notre modèle de segmentation basé sur une architecture U-Net modifiée par l'intégration de VGG16 pour le transfert learning. La suppression préalable du canal rouge et la découpe en sous-images permettent d'optimiser l'extraction des caractéristiques et la détection des micro-anévrismes.

Nous avons également détaillé la préparation de la base de données et les différentes étapes d'augmentation des données pour améliorer la robustesse du modèle. Enfin, la présentation du système optique combinant DMD et signaux ECG souligne l'aspect innovant du projet, en faisant le pont entre la segmentation numérique et la photocoagulation laser ciblée.

Le chapitre suivant sera consacré à la mise en œuvre expérimentale et à l'évaluation des performances de notre système.

Chapitre 4 :

Implémentations et Résultats

4 Chapitre 4 Implémentations et Résultats

4.1 Introduction

Dans ce chapitre, nous présentons la mise en œuvre pratique de notre méthode de segmentation des microanévrismes et les résultats obtenus.

De plus, ce chapitre aborde une comparaison entre modèle U-Net et notre modèle amélioré.

Enfin, nous présentons une expérience optique utilisant un DMD pour illustrer le principe de la photocoagulation multispot à haute précision.

4.2 Environnement de travail

Avant d'exécuter le programme, il est nécessaire d'installer les bibliothèques essentielles (fig 4.1) au bon fonctionnement du modèle.

Ces bibliothèques permettent de gérer le traitement des images, l'entraînement du modèle, ainsi que l'évaluation et l'affichage des résultats.

Parmi les bibliothèques utilisées :

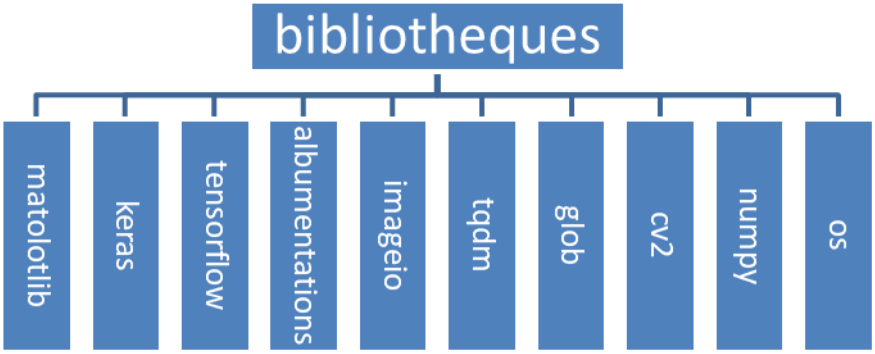


Figure 4.1: Les bibliothèques utilisée

4.3 La base d'apprentissage

4.3.1 Création

Dans notre projet, nous avons utilisé un ensemble de données personnalisé composé de 141 images du fond d'œil sans augmentation, et porté à 2320 images après augmentation.

La figure 4.2 présente un exemple concret de la base de données, montrant une image du fond d'œil avec son masque correspondant.

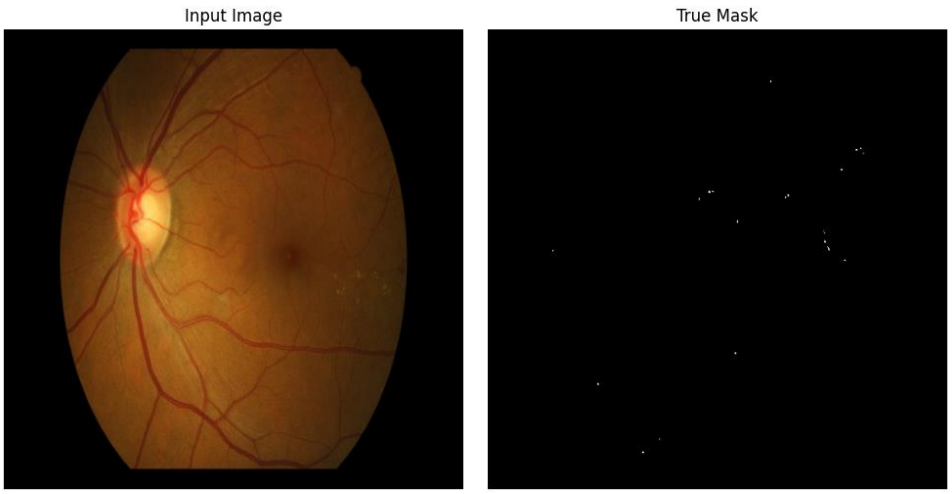


Figure 4.2: Exemple d'image du fond d'œil et de son masque annoté issu de notre base de données

4.3.2 Division de données :

Avant de passer aux étapes de redimensionnement et de normalisation, il est important de commencer par organiser les données. Cette étape vise à charger les images et leurs masques correspondants à partir d'un dossier structuré, puis à les diviser en 3 sous-ensembles (fig 4.3).

Les opérations principales sont les suivantes :

- Récupère les chemins d'accès des images et des masques avec glob, en ciblant les fichiers .jpg dans les sous-dossiers image et mask respectivement.
- Trie les fichiers pour garantir que chaque image correspond bien à son masque (même ordre dans les deux listes).

- Divise les données en deux parties grâce à train_test_split :
  80% pour l'entraînement
  20% pour la validation
  Ceci est contrôlé par le paramètre test_size=0.2.
- Fixe une graine aléatoire (random_state=42) pour que le découpage soit reproductible.

À la fin, la fonction retourne deux tuples contenant les chemins des images et masques pour l'entraînement et la validation.

```python
def load_data(path, test_size=0.2):
    """
    Charge les images et les masques à partir des chemins spécifiés
    et les divise en ensembles d'entraînement et de validation.
    """
    # Récupérer et trier les chemins des images
    images = sorted(glob(os.path.join(path, "image", "*.jpg")))

    # Récupérer et trier les chemins des masques
    masks = sorted(glob(os.path.join(path, "mask", "*.jpg")))

    # Diviser les données en ensembles d'entraînement et de validation
    train_images, valid_images, train_masks, valid_masks = train_test_split(
        images, masks, test_size=test_size, random_state=42
    )

    return (train_images, train_masks), (valid_images, valid_masks)
```

Figure 4.3:Code de division de données en ensemble d'entrainement et de validation

4.3.3 Prétraitement (Redimensionnement et normalisation)

Avant d'entraîner le modèle, il est essentiel de préparer les images et leurs masques (fig.4.4) afin d'assurer une cohérence dans les dimensions et les formats.

Cela se fait à travers les étapes suivantes :

- Charge l'image et le masque avec OpenCV (cv2.imread).
- Convertit le masque en niveaux de gris (cv2.IMREAD_GRAYSCALE) pour s'assurer qu'il est monochrome.
- Redimensionne l'image et le masque à 512x512 pour les adapter au modèle.
- Normalise les valeurs des pixels entre [0,1] pour optimiser l'entraînement.

```python
def preprocess_image(image_path, mask_path, img_size=(512, 512)):
    """
    Cette fonction permet de lire et de prétraiter une image ainsi que son masque associé.
    Elle redimensionne les données à une taille standard, puis applique une normalisation.
    """

    # Lecture de l'image couleur
    image = cv2.imread(image_path)

    # Lecture du masque en niveaux de gris
    mask = cv2.imread(mask_path, cv2.IMREAD_GRAYSCALE)

    # Redimensionnement de l'image et du masque à la taille spécifiée
    image = cv2.resize(image, img_size)
    mask = cv2.resize(mask, img_size)

    # Normalisation des valeurs de l'image entre 0 et 1
    image = image / 255.0
    mask = mask / 255.0

    # Ajout d'une dimension supplémentaire au masque (pour correspondre au format attendu par le modèle)
    mask = np.expand_dims(mask, axis=-1)

    return image, mask
```

Figure 4.4: Code de prétraitement des images et des masques (redimensionnement et normalisation)

4.3.4 Générateur de données

Afin d'optimiser l'utilisation des ressources lors de l'entraînement du modèle, un générateur de données (fig 4.5) a été utilisé.

Il charge progressivement les images et masques par petits lots pendant l'entraînement.

```python
class DataGenerator(Sequence):

    def __init__(self, image_paths, mask_paths, batch_size=8, img_size=(512, 512)):
        # Initialisation des chemins vers les images et masques, la taille des lots (batch)
        # et la taille cible des images.
        self.image_paths = image_paths
        self.mask_paths = mask_paths
        self.batch_size = batch_size
        self.img_size = img_size

    def __len__(self):
        # Retourne le nombre total de lots (batches) par époque.
        return int(np.ceil(len(self.image_paths) / self.batch_size))

    def __getitem__(self, idx):
        # Sélection des chemins d'images et de masques pour le lot courant
        batch_image_paths = self.image_paths[idx * self.batch_size:(idx + 1) * self.batch_size]
        batch_mask_paths = self.mask_paths[idx * self.batch_size:(idx + 1) * self.batch_size]

        images = []
        masks = []

        # Chargement et prétraitement des images et masques du lot
        for img_path, mask_path in zip(batch_image_paths, batch_mask_paths):
            image, mask = preprocess_image(img_path, mask_path, self.img_size)
            images.append(image)
            masks.append(mask)

        # Conversion des listes en tableaux NumPy pour être compatibles avec le modèle
        return np.array(images), np.array(masks)
```

Figure 4.5 : Code de création d'un générateur de données pour l'entrainement et validation par lots

4.4 Implémentation du notre modèle

Le modèle utilisé pour la segmentation des micro-anévrismes est basé sur l'architecture U-Net. Afin d'adapter ce modèle à la spécificité des images de fond d'œil, une série d'étapes de prétraitement et de transformation a été appliquée avant et après la segmentation.

Étape 1 : Prétraitement par CLAHE sans canal rouge

```python
def preprocess_clahe(image_path, target_size=(512, 512)):
    img = cv2.imread(image_path)
    if img is None:
        raise FileNotFoundError("Image introuvable")

    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img_rgb[:, :, 0] = 0  # suppression du canal rouge
    green = img_rgb[:, :, 1]
    blue = img_rgb[:, :, 2]

    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    green_clahe = clahe.apply(green)
    blue_clahe = clahe.apply(blue)

    gray_combined = ((green_clahe.astype(np.float32) + blue_clahe.astype(np.float32)) / 2).astype(np.uint8)
    gray_combined = cv2.resize(gray_combined, target_size)
    return gray_combined
```

Figure 4.6: code de suppression du canal rouge

Étape 2 : Division de l'image en 16 sous-images redimensionnées

Afin d'augmenter la résolution et de permettre une meilleure focalisation sur les micro-détails, l'image prétraitée a été découpée en 16 sous-images de taille 128×128 pixels. Chaque sous-image a ensuite été interpolée en 512×512 pixels pour être compatible avec l'entrée du modèle U-Net.

```python
def split_and_resize(image, sub_patch=(128, 128), resize_to=(512, 512)):
    blocks = view_as_blocks(image, block_shape=sub_patch)
    patches = blocks.reshape(-1, sub_patch[0], sub_patch[1])
    resized_patches = [cv2.resize(p, resize_to) for p in patches]
    return np.array(resized_patches)
```

Figure 4.7: Code de division de l'image en 16 sous-images redimensionnées

Étape 3 : Architecture U-Net avec VGG-16

```python
def conv_block(inputs, filters):
    x = Conv2D(filters, 3, padding="same")(inputs)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)
    x = Conv2D(filters, 3, padding="same")(x)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)
    return x

def decoder_block(inputs, skip, filters):
    x = Conv2DTranspose(filters, (2, 2), strides=2, padding="same")(inputs)
    x = Concatenate()([x, skip])
    x = conv_block(x, filters)
    return x

def build_unet_vgg16(input_shape=(512, 512, 3)):
    inputs = Input(input_shape)
    vgg = VGG16(include_top=False, weights='imagenet', input_tensor=inputs)
    for layer in vgg.layers:
        layer.trainable = False
    s1 = vgg.get_layer("block1_conv2").output
    s2 = vgg.get_layer("block2_conv2").output
    s3 = vgg.get_layer("block3_conv3").output
    s4 = vgg.get_layer("block4_conv3").output
    b1 = vgg.get_layer("block5_conv3").output

    d1 = decoder_block(b1, s4, 512)
    d2 = decoder_block(d1, s3, 256)
    d3 = decoder_block(d2, s2, 128)
    d4 = decoder_block(d3, s1, 64)

    outputs = Conv2D(1, 1, activation='sigmoid')(d4)
    model = Model(inputs, outputs, name="U-Net_VGG16")
    return model
```

Figure 4.8: Code de l'architecture U-Net avec VGG-16 comme encodeur

Étape 4 : Reconstruction du masque complet

Après avoir appliqué le modèle U-Net à chaque sous-image redimensionnée, les 16 masques prédits ont été réassemblés dans leur configuration spatiale d'origine. Chaque masque a été redimensionné à 128×128 pixels, puis repositionné pour reformer un seul masque global de taille 512×512 pixels. Cette étape est cruciale pour obtenir une prédiction de segmentation complète sur l'ensemble du fond d'œil.

```python
def reconstruct_mask(patch_masks, original_shape=(512, 512)):
    sub_size = original_shape[0] // 4
    mask = np.zeros(original_shape, dtype=np.float32)
    idx = 0
    for i in range(4):
        for j in range(4):
            resized = cv2.resize(patch_masks[idx], (sub_size, sub_size))
            mask[i*sub_size:(i+1)*sub_size, j*sub_size:(j+1)*sub_size] = resized
            idx += 1
    return mask
```

Figure 4.9: Code de reconstruction du masque complet a partir des sous-prédictions

Étape 5 : Visualisation du résultat

Le masque final reconstruit permet de visualiser l'ensemble des régions segmentées dans l'image initiale. Ce masque binaire indique les zones probables de présence de micro-anévrismes, fournissant ainsi une base pour les étapes ultérieures de traitement ou de thérapie (comme la projection optique ou la photocoagulation ciblée par laser).

```
# 1. Charger et prétraiter l'image
image_path = "DS000FGD.JPG"
gray_image = preprocess_clahe(image_path)

# 2. Créer les 16 sous-images redimensionnées
patches = split_and_resize(gray_image)

# 3. Préparer les données pour U-Net
patches_input = np.stack([cv2.merge([p, p, p]) / 255.0 for p in patches])  # RGB simulé

# 4. Charger le modèle
model = build_unet_vgg16()

# 5. Prédire les masques pour chaque patch
predicted_masks = model.predict(patches_input)
predicted_masks = predicted_masks.squeeze(axis=-1)  # (16, 512, 512)

# 6. Recomposer le masque final
final_mask = reconstruct_mask(predicted_masks)

# 7. Afficher
plt.imshow(final_mask, cmap='gray')
plt.title("Masque Reconstruit")
plt.axis("off")
plt.show()
```

Figure 4.10: Code de construction notre modèle pour la segmentation d'images rétiniennes

4.5 Comment utiliser le modèle sauvegardé

Pour éviter de réentraîner le modèle à chaque fois, nous avons utilisé une méthode qui permet de le sauvegarder et de le recharger facilement. Les étapes suivantes montrent comment charger les poids, le modèle complet, traiter une image, puis faire une prédiction.

- **Chargement des poids :**

```
model = load_model('/content/drive/MyDrive/model18022025.h5', compile=False)
```

Figure 4.11: Code pour charger les poids sauvegardes depuis le fichier .h5

- **Chargement du modèle complet**

```
def conv_block(inputs, num_filters):
    x = Conv2D(num_filters, 3, padding="same")(inputs)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    x = Conv2D(num_filters, 3, padding="same")(x)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    return x

def encoder_block(inputs, num_filters):
    x = conv_block(inputs, num_filters)
    p = MaxPool2D((2, 2))(x)
    return x, p

def decoder_block(inputs, skip_features, num_filters):
    x = Conv2DTranspose(num_filters, (2, 2), strides=2, padding="same")(inputs)
    x = Concatenate()([x, skip_features])
    x = conv_block(x, num_filters)
    return x

def build_unet(input_shape):
    inputs = Input(input_shape)

    s1, p1 = encoder_block(inputs, 64)
    s2, p2 = encoder_block(p1, 128)
    s3, p3 = encoder_block(p2, 256)
    s4, p4 = encoder_block(p3, 512)

    b1 = conv_block(p4, 1024)

    d1 = decoder_block(b1, s4, 512)
    d2 = decoder_block(d1, s3, 256)
    d3 = decoder_block(d2, s2, 128)
    d4 = decoder_block(d3, s1, 64)

    outputs = Conv2D(1, 1, padding="same", activation="sigmoid")(d4)

    model = Model(inputs, outputs, name="UNET")
    return model
```

Figure 4.12: code pour charger le modèle

- **Traitement de l'image**

```python
def preprocess_test_image(image_path, img_size=(512, 512)):
    """
    Charger et prétraiter une image de test de la même manière que pour l'entraînement du modèle.
    """
    image = cv2.imread(image_path)  # Charger l'image
    image = cv2.resize(image, img_size)  # Redimensionner à (512, 512)
    image = image / 255.0  # Normaliser les valeurs entre 0 et 1
    image = np.expand_dims(image, axis=0)  # Ajouter la dimension batch

    return image

# Exemple d'une image de test
test_image_path = "/content/drive/MyDrive/extracted_data/Test/Fundus_Images/007-1774-100_Patch_Matrix12.jpg"
test_image = preprocess_test_image(test_image_path)
```

Figure 4.13: Code pour prétraiter l'image d'entrée avant de faire une prédiction

- **Faire une prédiction**

```python
# Effectuer la prédiction
pred_mask = model.predict(test_image)[0]  # Supprimer la dimension batch

# Convertir les prédictions en une image binaire
threshold = 0.5
binary_mask = (pred_mask > threshold).astype(np.uint8)

# Afficher l'image originale et le masque prédit
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.imshow(cv2.imread(test_image_path)[..., ::-1])  # Convertir de BGR à RGB
plt.title("Image originale")

plt.subplot(1, 2, 2)
plt.imshow(binary_mask.squeeze(), cmap="gray")  # Afficher le masque en niveaux de gris
plt.title("Masque prédit")

plt.show()
```

Figure 4.14: Code pour utiliser le modèle charge pour faire une prédiction sur une nouvelle image

4.6 Entrainement

L'entraînement du modèle U-Net a été réalisé sur la plateforme Google Colab en utilisant TensorFlow/Keras. Le modèle a été compilé avec l'optimiseur Adam, reconnu pour sa rapidité de convergence et sa stabilité lors des tâches de segmentation.

Nous avons utilisé une fonction de perte adaptée à ce type de tâche, à savoir la Dice loss, qui permet d'améliorer la performance du modèle dans la détection de petites régions pathologiques.

Les hyperparamètres d'entraînement étaient les suivants :

- Taille du batch : 4
- Nombre d'époques : 30
- Taille d'entrée des images : 512 × 512 x 3
- Taux d'apprentissage initial : 0.001

Le taux d'apprentissage est l'un des paramètres les plus importants dans l'entraînement des réseaux de neurones, car il contrôle la taille du pas que fait le modèle lors de la mise à jour des poids afin de minimiser la fonction de perte.

Lorsqu'on choisit un taux d'apprentissage trop petit, le modèle apprend très lentement et peut nécessiter un très grand nombre d'époques pour obtenir un bon résultat.

En revanche, si le taux d'apprentissage est trop élevé, cela peut entraîner un dépassement du minimum de la fonction de perte, une instabilité de l'entraînement, voire l'échec total de l'apprentissage du modèle.

Nous avons choisi la valeur 0.001 au lieu de 0.0001 car elle offre un bon compromis entre rapidité et précision : elle permet au modèle d'apprendre à une vitesse modérée sans trop dépasser le point optimal.

L'entraînement a été réalisé à l'aide du générateur de données personnalisé décrit précédemment, ce qui a permis une gestion efficace de la mémoire tout en assurant un flux continu d'images et de masques pendant le processus d'apprentissage.

Afin d'améliorer la qualité de l'apprentissage et d'éviter le surapprentissage (overfitting), nous avons intégré trois callbacks principaux de Keras :

- ReduceLROnPlateau : réduit automatiquement le taux d'apprentissage si la fonction de perte de validation (val_loss) stagne pendant trois époques consécutives ;
- EarlyStopping : interrompt l'entraînement si la performance en validation ne s'améliore plus après cinq époques, tout en restaurant les meilleurs poids ;
- ModelCheckpoint : enregistre automatiquement le meilleur modèle obtenu selon la perte de validation.

4.7 Evaluation

L'évaluation d'un modèle de segmentation repose sur des métriques spécifiques permettant de mesurer la qualité des prédictions par rapport à la vérité terrain (Ground Truth), c'est-à-dire les masques annotés manuellement.

Dans le cadre de notre projet, dont l'objectif est de détecter avec précision les micro-anévrismes (MA) dans des images de la rétine, cette étape d'évaluation est cruciale pour valider la performance du modèle dans un contexte médical sensible.

Nous avons choisi comme métrique principale le coefficient de similarité de Dice qui est largement utilisé en segmentation d'images médicales. Il permet de quantifier le degré de correspondance entre la prédiction du modèle et le masque fourni par un expert.

Le coefficient de Dice a été utilisé à deux niveaux :

- Comme fonction de perte (dice_loss) pendant l'entraînement, afin d'orienter le modèle vers des prédictions plus précises.
- Comme métrique d'évaluation, pour mesurer la performance sur les ensembles de validation et de test.

Enfin, nous avons également calculé la matrice de confusion sur les données de test. Bien que cette matrice soit davantage utilisée en classification binaire ou multiclasse, elle s'avère utile dans notre cas en traitant chaque pixel comme une instance appartenant soit à la classe "pathologique" (MA) soit à la classe "saine".

4.8 Implémentation du dispositif optique

Afin de valider expérimentalement le concept de photocoagulation laser sélective, une expérience optique a été réalisée (fig 4.15) en utilisant un DMD comme modulateur spatial de lumière.

Cette expérience permet de simuler une photocoagulation multispot ciblée des micro-anévrismes détectés automatiquement via un modèle de segmentation.

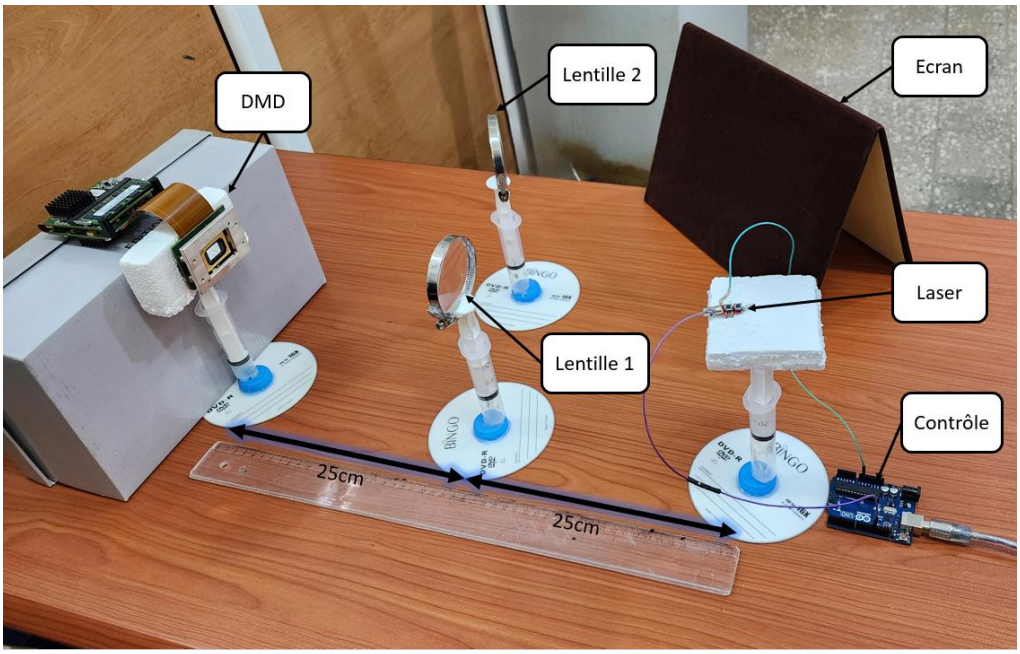


Figure 4.15: Système optique réel utilisant le DMD

4.9 Résultats

Dans cette section, nous présentons les résultats obtenus à travers les différentes étapes de notre système de segmentation des micro-anévrismes.

4.9.1 Résultats de prétraitement et d'augmentation

Plusieurs étapes ont été réalisées pour préparer les images et les masques à l'entraînement de notre U-Net, visant la segmentation des micro-anévrismes dans les fonds d'œil.

4.9.1.1 Prétraitement par CLAHE (sans canal rouge)

Un traitement d'amélioration du contraste a été appliqué en supprimant le canal rouge.

Résultat :

- Des images en niveaux de gris à fort contraste
- Meilleure visibilité des micro-anévrismes et structures vasculaires

Les effets de cette amélioration sur l'image du fond d'œil sont montrés dans la figure 4.16, où l'on observe une distinction plus marquée entre les structures.

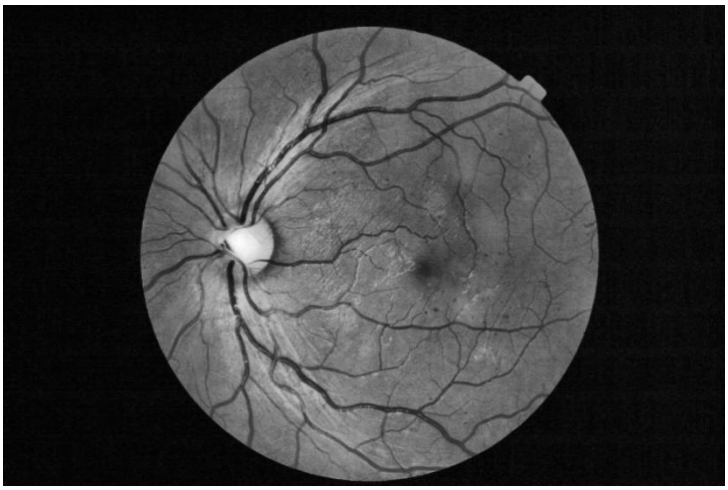


Figure 4.16: Résultat de l'amélioration du contraste du fond d'œil par suppression du canal rouge

La figure 4.17 illustre l'effet de cette technique sur la détection des micro-anévrismes.

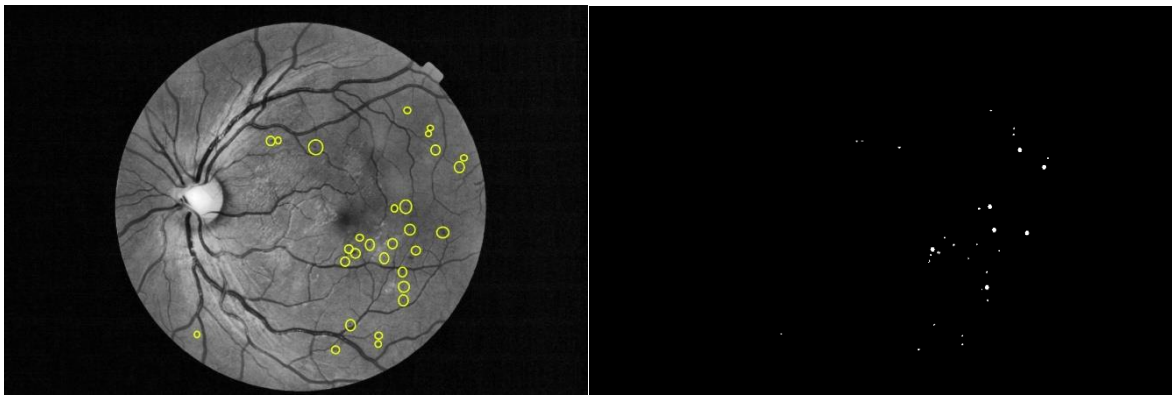


Figure 4.17: Illustration des micro-anévrismes (MA) et du masque correspondant après élimination du canal rouge

4.9.1.2 Découpage en sous-images

Chaque image a été divisée en 16 sous-images de taille 128×128 pixels (fig 4.18), puis redimensionnée à 512×512 pixels pour respecter la taille d'entrée du U-Net.

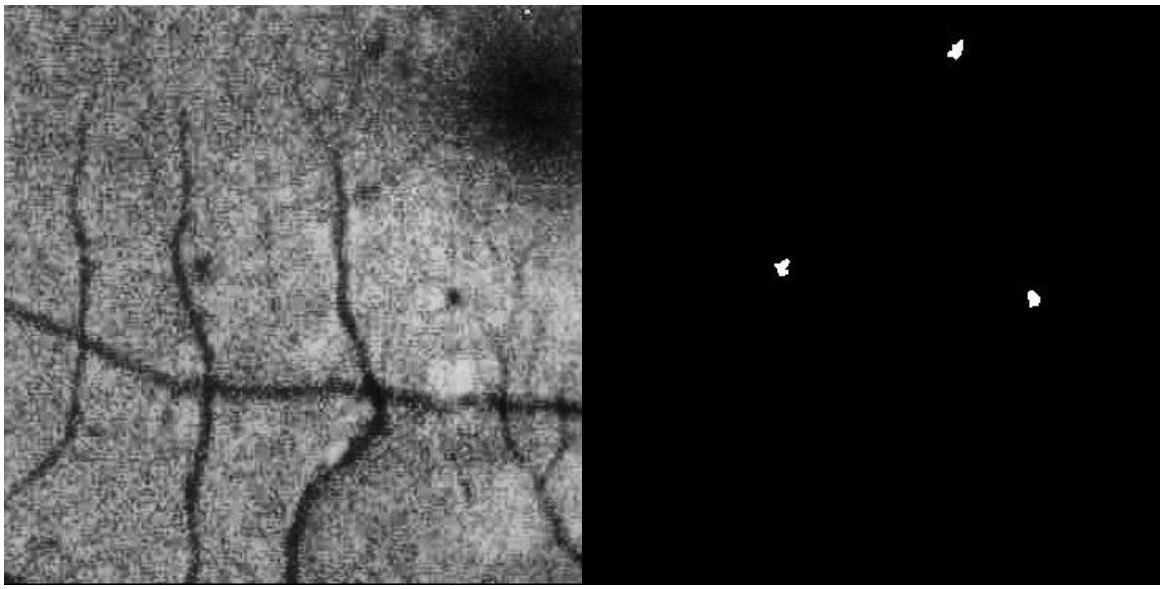


Figure 4.18: résultat de l'augmentation par découpage de l'image

4.9.2 Résultats de l'entrainement

Pour la tâche de segmentation des micro-anévrismes, deux approches d'entraînement ont été mises en œuvre afin d'évaluer l'efficacité de notre modèle sur une base de données restreinte. La fonction de perte utilisée dans tous les cas est la Dice Loss, particulièrement adaptée aux problèmes de segmentation avec classes déséquilibrées.

4.9.2.1 Étape 1 : U-Net avec augmentation de données

Dans un premier temps, nous avons entraîné un modèle U-Net sur notre propre base de données annotée, en appliquant des techniques d'augmentation de données pour améliorer la généralisation et compenser la taille réduite de l'ensemble d'entraînement. Les augmentations utilisées incluent :

- Flip horizontal
- Flip vertical
- Transformation élastique
- Distorsion par grille
- Distorsion optique

Un exemple de résultat d'une image ayant subit l'augmentation des données (fig.4.19) :

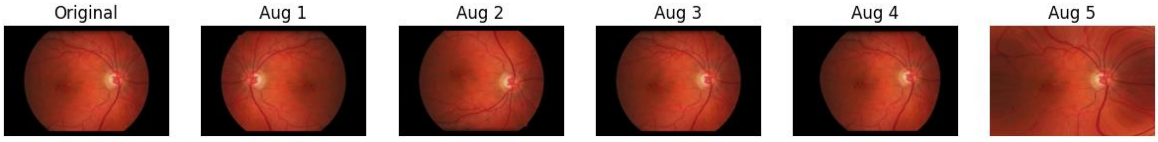


Figure 4.19: Un exemple de résultat d'une image ayant subit l'augmentation des données

Lors de la phase de traitement (Training), nous avons effectué l'entraînement du modèle sur seulement 13 époques (Epochs) en raison de la limitation du temps disponible sur la plateforme Google Colab, qui offre un temps d'exécution relativement court, insuffisant pour les projets nécessitant une durée plus longue. Malgré cela, les résultats obtenus sont présentés dans figure 4.20.

```
Epoch 1/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 4s/step - accuracy: 0.8057 - loss: 0.9893
Epoch 1: val_loss improved from inf to 0.97090, saving model to /content/drive/MyDrive/model04062025.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file forma
680/680 ━━━━━━━━━━━━━━━━━━━━ 3887s 6s/step - accuracy: 0.8059 - loss: 0.9893 - val_accuracy: 0.9601 - val_loss: 0.9709 - learn
Epoch 2/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 672ms/step - accuracy: 0.9919 - loss: 0.9147
Epoch 2: val_loss improved from 0.97090 to 0.76204, saving model to /content/drive/MyDrive/model04062025.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file forma
680/680 ━━━━━━━━━━━━━━━━━━━━ 512s 753ms/step - accuracy: 0.9919 - loss: 0.9146 - val_accuracy: 0.9958 - val_loss: 0.7620 - lea
Epoch 3/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 672ms/step - accuracy: 0.9955 - loss: 0.6901
Epoch 3: val_loss improved from 0.76204 to 0.69797, saving model to /content/drive/MyDrive/model04062025.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file forma
680/680 ━━━━━━━━━━━━━━━━━━━━ 562s 753ms/step - accuracy: 0.9955 - loss: 0.6901 - val_accuracy: 0.9947 - val_loss: 0.6980 - lea
Epoch 4/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 677ms/step - accuracy: 0.9957 - loss: 0.6036
Epoch 4: val_loss did not improve from 0.69797
680/680 ━━━━━━━━━━━━━━━━━━━━ 542s 797ms/step - accuracy: 0.9957 - loss: 0.6036 - val_accuracy: 0.9914 - val_loss: 0.8598 - lea
Epoch 5/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 672ms/step - accuracy: 0.9959 - loss: 0.5748
Epoch 5: val_loss improved from 0.69797 to 0.64228, saving model to /content/drive/MyDrive/model04062025.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file forma
680/680 ━━━━━━━━━━━━━━━━━━━━ 542s 797ms/step - accuracy: 0.9959 - loss: 0.5748 - val_accuracy: 0.9956 - val_loss: 0.6423 - lea
Epoch 6/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 683ms/step - accuracy: 0.9960 - loss: 0.5547
Epoch 6: val_loss did not improve from 0.64228
680/680 ━━━━━━━━━━━━━━━━━━━━ 516s 759ms/step - accuracy: 0.9960 - loss: 0.5546 - val_accuracy: 0.9959 - val_loss: 0.6536 - lea
Epoch 7/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 672ms/step - accuracy: 0.9960 - loss: 0.5259
Epoch 7: val_loss did not improve from 0.64228
680/680 ━━━━━━━━━━━━━━━━━━━━ 539s 792ms/step - accuracy: 0.9960 - loss: 0.5259 - val_accuracy: 0.9958 - val_loss: 0.6461 - lea
Epoch 8/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 672ms/step - accuracy: 0.9960 - loss: 0.5051
Epoch 8: val_loss improved from 0.64228 to 0.64183, saving model to /content/drive/MyDrive/model04062025.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file forma
680/680 ━━━━━━━━━━━━━━━━━━━━ 512s 753ms/step - accuracy: 0.9960 - loss: 0.5051 - val_accuracy: 0.9957 - val_loss: 0.6418 - lea
Epoch 9/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 678ms/step - accuracy: 0.9959 - loss: 0.4841
Epoch 9: val_loss did not improve from 0.64183
680/680 ━━━━━━━━━━━━━━━━━━━━ 543s 799ms/step - accuracy: 0.9959 - loss: 0.4842 - val_accuracy: 0.9955 - val_loss: 0.6591 - lea
Epoch 10/30
680/680 ━━━━━━━━━━━━━━━━━━━━ 0s 672ms/step - accuracy: 0.9959 - loss: 0.4742
Epoch 10: val_loss did not improve from 0.64183
680/680 ━━━━━━━━━━━━━━━━━━━━ 509s 749ms/step - accuracy: 0.9959 - loss: 0.4742 - val_accuracy: 0.9960 - val_loss: 0.7238 - lea
```

Figure 4.20: Visualisation de l'entrainement de 1ere étape

Afin de mieux comprendre les différents paramètres affichés pendant l'entraînement, le tableau suivant en présente les définitions :

Tableau 4.1: définition des paramètres affiches durant l'entrainement du modèle

Paramètre	Définition
Epoch X/Y	Indique le numéro de l'époque actuelle et le nombre total d'époques définies pour l'entraînement
Accuracy	Précision du modèle sur l'ensemble d'entraînement. Elle représente le pourcentage de prédictions correctes
Loss	La valeur de la fonction de perte (ici dice_loss). Une valeur plus faible indique une meilleure performance du modèle
Val_accuracy	Précision du modèle sur l'ensemble de validation
Val_loss	Perte calculée sur l'ensemble de validation. Si cette valeur augmente, cela peut indiquer un surapprentissage (overfitting)
Learning rate	Taux d'apprentissage de l'optimiseur (Adam)
Temps par epoch	Le temps nécessaire pour compléter une époque, mesuré en secondes par étape (s/step) Il dépend de la taille des données, de la complexité du modèle et des ressources matérielles (CPU/GPU/TPU)

Les résultats ont été présentés sous forme de courbes de perte (Loss Curve) et de courbes de précision (Accuracy Curve) (fig.4.21).

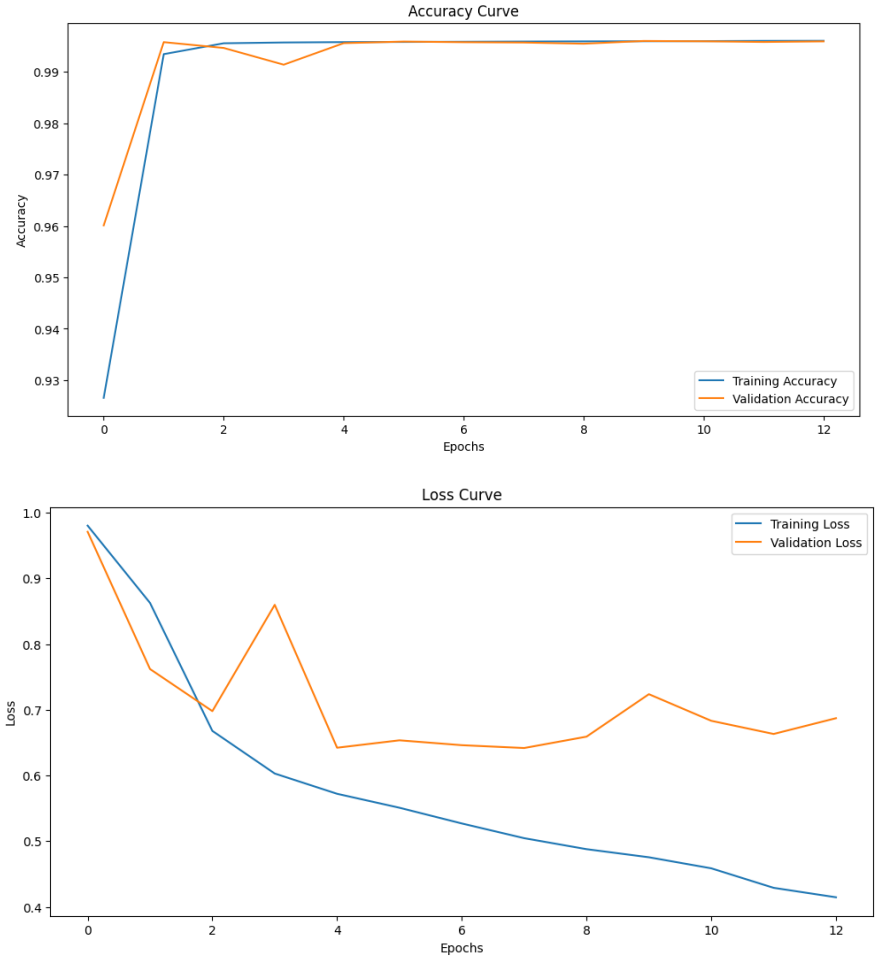


Figure 4.21:les courbes de accuracy et de perte pour l'entraînement et la validation

Le temps total d'entraînement pour les 13 époques est :

Le temps d'entrainement = 3887 + 512 + 562 + 542 + 542 + 516 + 539 + 512 + 543 + 509 + 509 + 509 + 539 = 10221s

4.9.2.2 Étape 2 : Modèle personnalisé

Dans cette deuxième phase, nous avons conçu un modèle personnalisé, intégrant:

- Un prétraitement CLAHE avec suppression du canal rouge
- Un découpage en sous-images
- Et l'application du transfert learning via l'architecture VGG-16 dans le U-Net.

Nous avons entraîné notre modèle personnalisé sur 30 époques (Epochs). Avec un suivi précis des performances

Les resultats ont ete présentés sous forme de courbes de perte (Loss Curve) et de courbes de précision (Accuracy Curve) (fig 4.22).

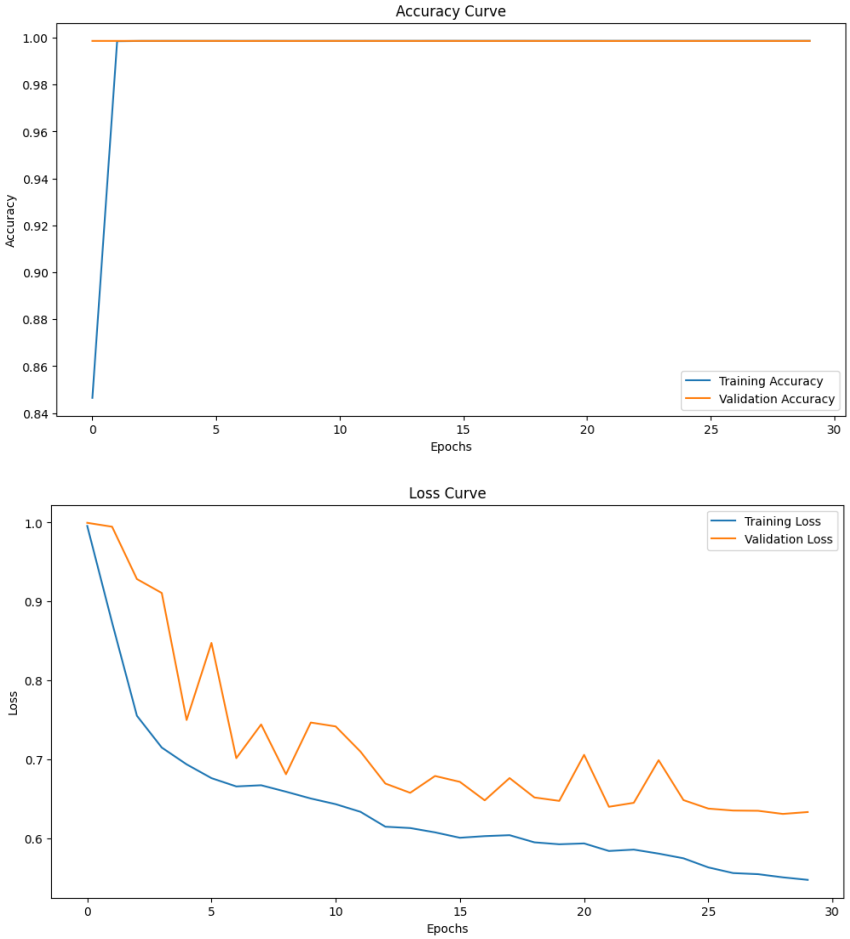


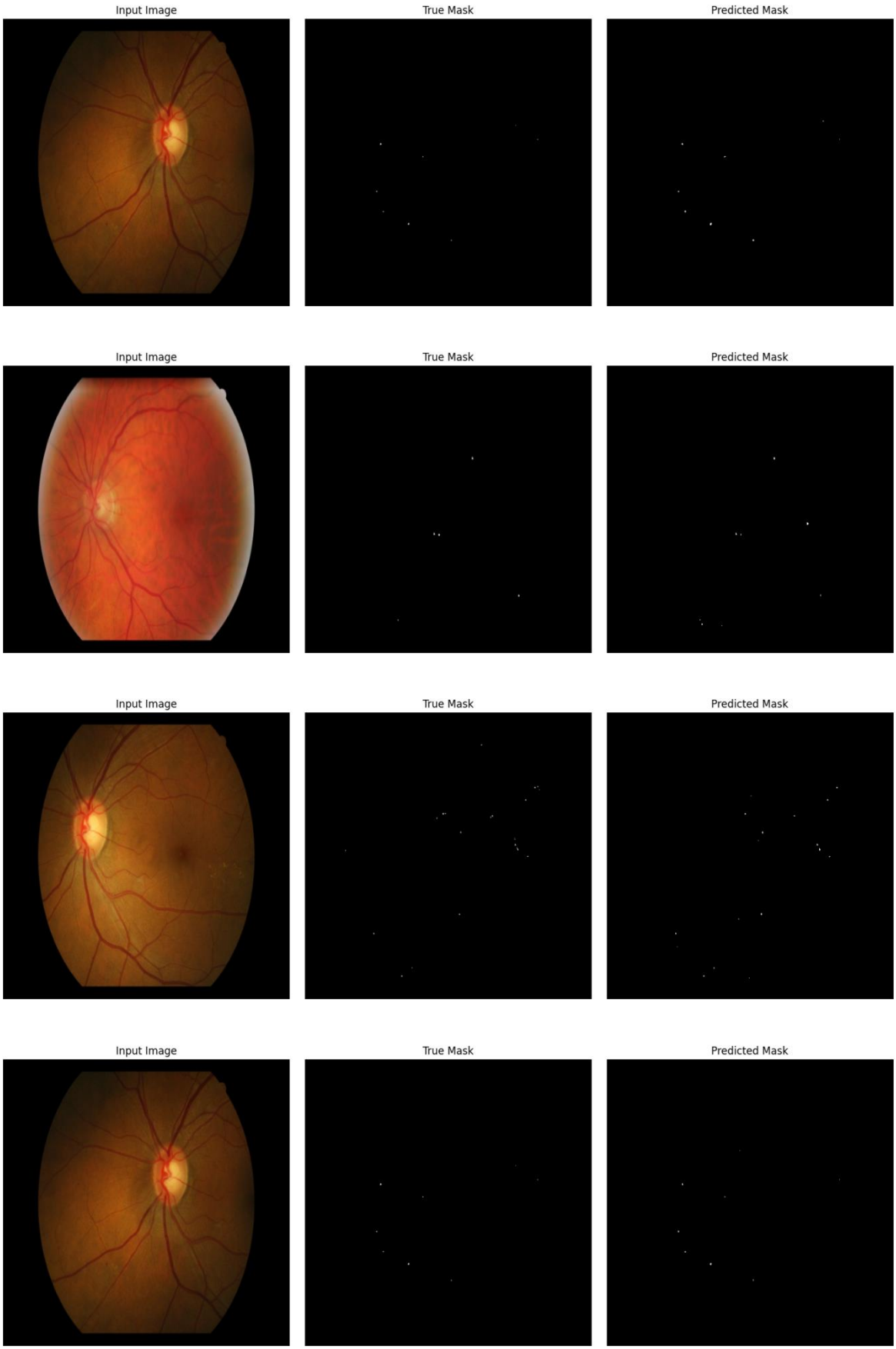
Figure 4.22:les courbes de acccuracy de perte pour l'entraînement et la validation de notre modèle

Le temps total d'entraînement pour les 30 époques est :

Le temps d'entrainement = 271 + 177 + 166 + 168 + 164 + 199 + 167 + 161 + 166 + 161 + 161 + 161 + 168 + 168 + 170 + 161 + 164 + 199 + 161 + 168 + 161 + 163 + 161 + 169 + 162 + 169 + 176 + 194 + 169 + 162 = 5167s

4.9.3 Résultats de test

Les résultats des tests menés sur les images de la base de données ont permis d'évaluer visuellement la performance de nos modèles. Comme illustré à la figure 4.23, le premier modèle, basé sur l'architecture U-Net, a réussi à détecter plusieurs zones pathologiques. Toutefois, certaines zones ont été identifiées par le modèle mais n'étaient pas annotées dans les masques. Après consultation avec une ophtalmologue, il a été estimé que ces zones sont effectivement de véritables lésions. Cela met en évidence la sensibilité et la précision du modèle face à certaines anomalies subtiles.


Input Image
True Mask
Predicted Mask
Input Image
True Mask
Predicted Mask
Input Image
True Mask
Predicted Mask
Input Image
True Mask
Predicted Mask

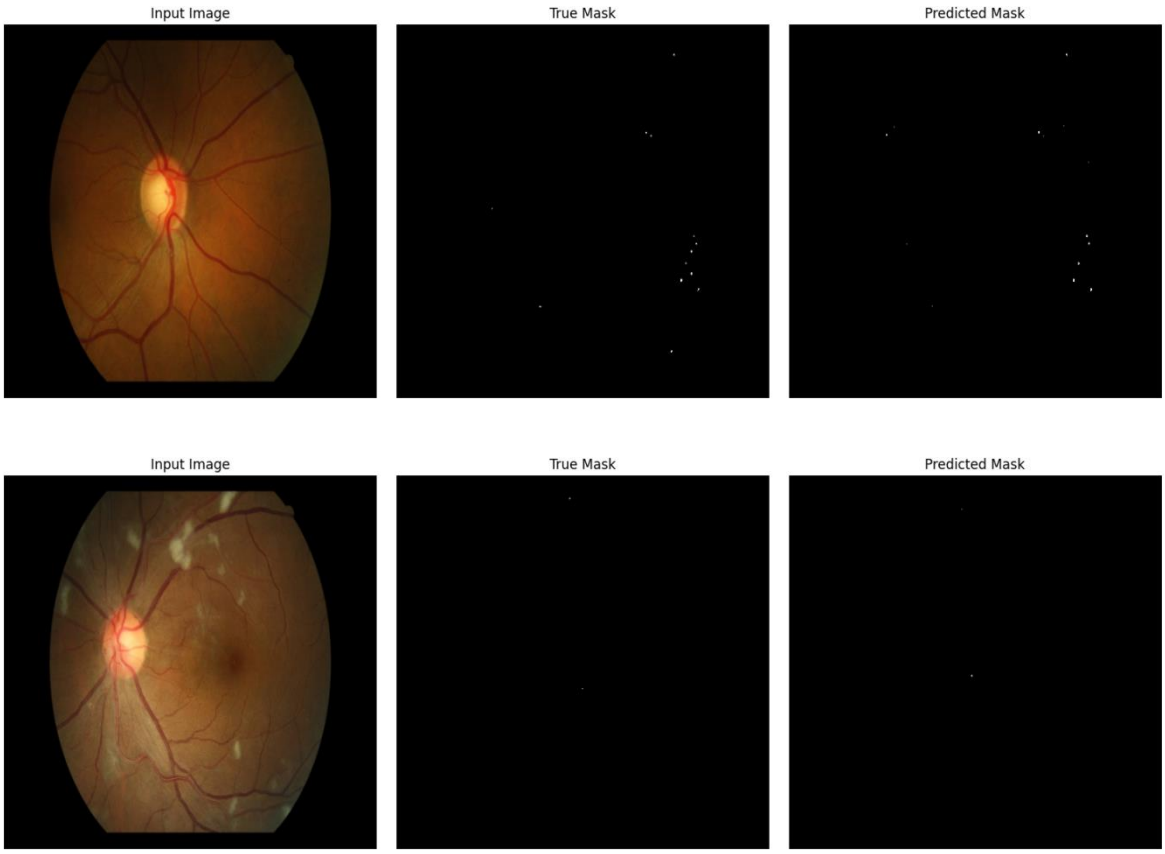
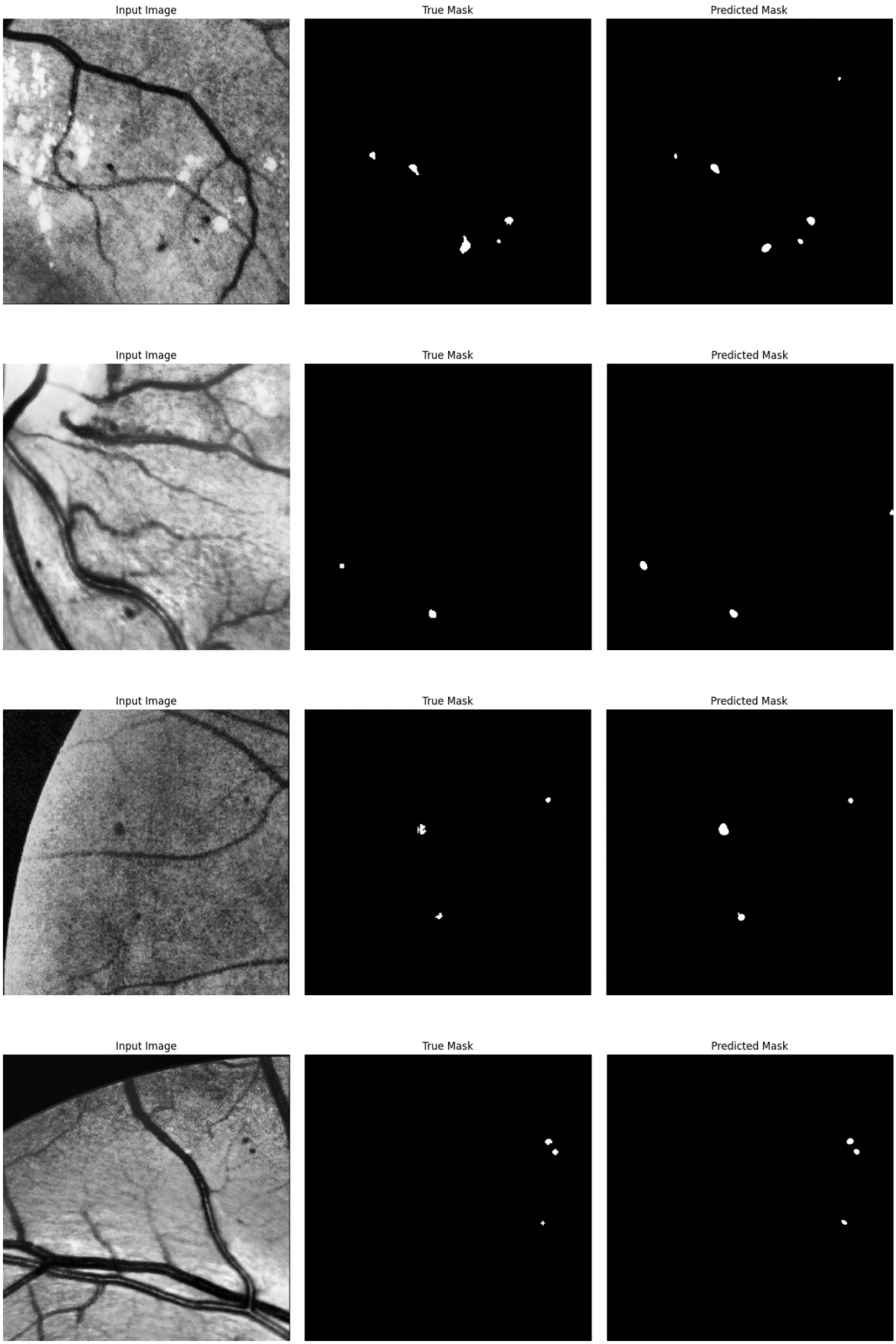


Figure 4.23: Résultats de prédiction sur les données de test

En parallèle, nous avons évalué les résultats du modèle 2, entraîné avec une stratégie d'apprentissage par transfert (Transfer Learning). Les prédictions obtenues sont présentées dans la figure 4.24, où l'on constate une meilleure précision morphologique des zones segmentées ainsi qu'une réduction du bruit de prédiction.

Enfin, en comparant visuellement les prédictions des deux modèles. Il apparaît clairement que le modèle amélioré (modèle 2) permet une segmentation plus fine et plus cohérente des zones pathologiques, confirmant ainsi l'intérêt des méthodes de prétraitement et du transfert learning dans notre contexte d'étude.


Input Image
True Mask
Predicted Mask
Input Image
True Mask
Predicted Mask
Input Image
True Mask
Predicted Mask
Input Image
True Mask
Predicted Mask

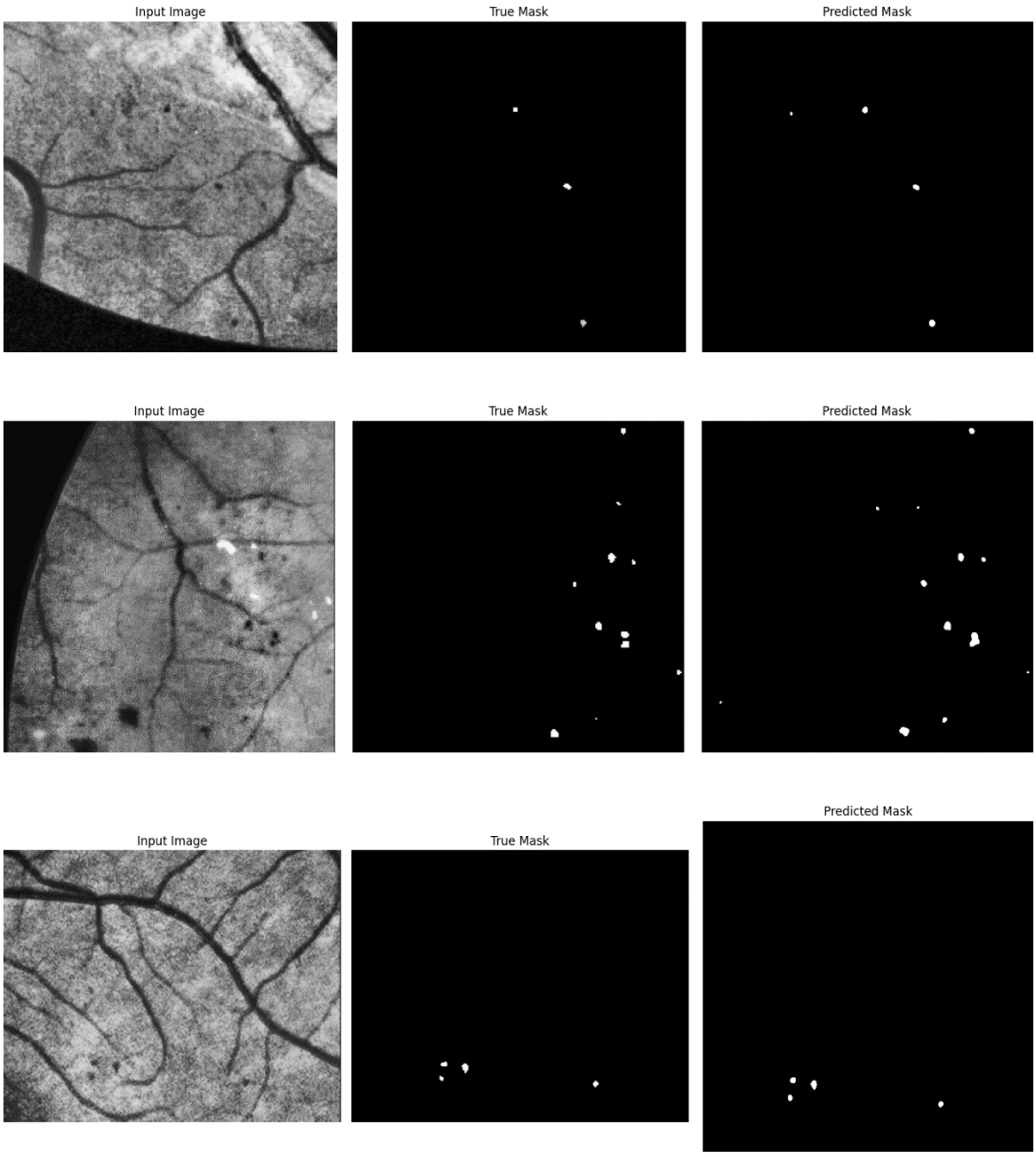


Figure 4.24: Résultats de prédiction avec transfert Learning

4.9.4 Résultat de la démonstration du principe de photocoagulation multispot

L'image projetée sur l'écran correspond fidèlement à la carte binaire issue du réseau de segmentation U-Net (fig 4.25). Les spots lumineux représentent les zones pathologiques détectées, prêtes à recevoir une impulsion laser.

Un point important observé lors des essais est que l'allumage du faisceau lumineux était rigoureusement contrôlé, avec un clignotement limité et précis, grâce au

système de déclenchement basé sur les signaux ECG. Ce dernier permettait d'autoriser ou de bloquer la projection en temps réel, en fonction des mouvements oculaires simulés, garantissant ainsi une activation sécurisée et contextuelle du système.

Cette expérience démontre la faisabilité d'une photocoagulation multispot ciblée, avec un contrôle spatial (via le DMD) et temporel (via les signaux ECG) très précis. Le système réagit uniquement lorsque les conditions d'alignement sont remplies, illustrant le potentiel d'une plateforme intelligente et sécurisée de traitement laser assisté par IA et biométrie.

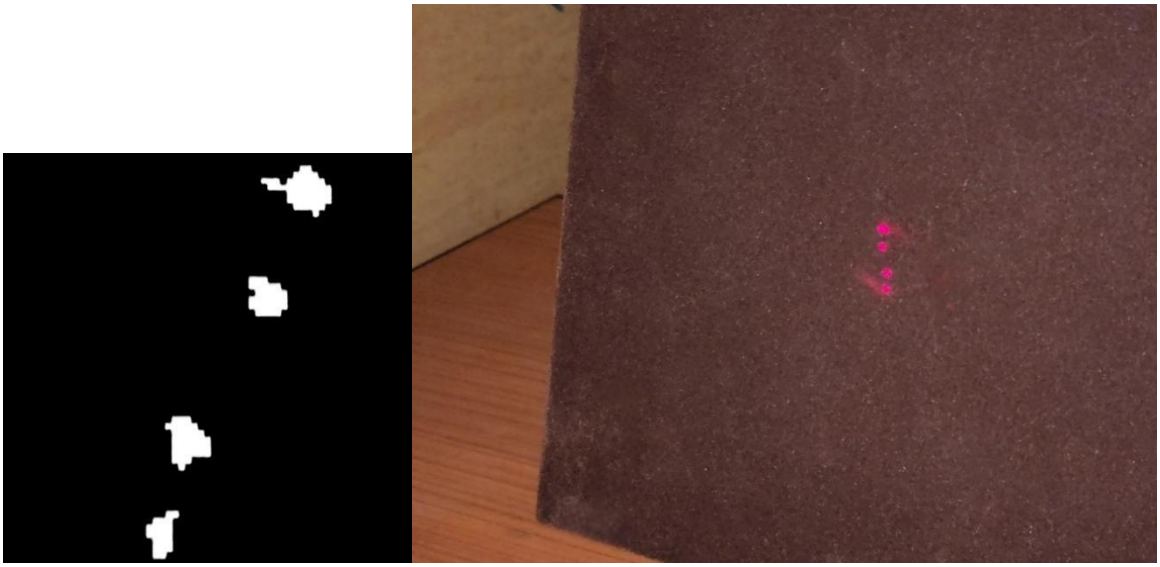


Figure 4.25: Visualisation du masque binaire segmente projeter sur l'écran -zones pathologiques ciblées par le DMD

4.10 Analyse les résultats

Malgré une durée d'entraînement relativement courte lors de l'utilisation de VGG16 (en dépit du nombre élevé d'époques), les deux modèles U-Net ont montré de bonnes performances dans la segmentation des microanévrismes, qui sont des structures très petites par rapport à l'arrière-plan.

4.10.1 Accuracy

Modèle 1

- L'accuracy d'entraînement augmente rapidement, atteignant environ 99,6 % à partir de la 8eme époque.
- L'accuracy de validation est aussi très élevée, autour de 99,5 %–99,6 %, ce qui indique que le modèle ne souffre pas de surapprentissage majeur et généralise bien aux données de validation.

Modèle 2

- L'accuracy d'entraînement démarre plus bas (70,68 %) mais atteint rapidement plus de 99,8 %.
- L'accuracy de validation est très stable et reste également autour de 99,8 %, confirmant la bonne capacité du modèle à généraliser.

Comparaison

Le modèle 2 montre une légère supériorité en termes de précision à la fois pour l'entraînement et la validation, avec une meilleure stabilité globale.

Cependant, il convient de noter que ces résultats peuvent être trompeurs, car l'arrière-plan du masque est très dominant.

C'est pourquoi nous avons évalué l'évolution de la courbe de perte (Loss Curve), en utilisant notamment une fonction de perte adaptée telle que la Dice Loss, car elle accorde une plus grande importance au déséquilibre des classes, ce qui permet une comparaison plus juste entre les modèles.

Pour évaluer les performances réelles du notre modèle, nous avons également calculé la matrice de confusion (Matrice de Confusion), qui offre une vision plus claire de la manière dont le modèle traite chaque classe (zones pathologiques vs arrière-plan), et permet d'analyser les points forts et les limites des résultats de segmentation.

4.10.2 Perte

Modèle 1

La perte d'entraînement diminue progressivement. En revanche, la perte de validation est plus fluctuante et reste généralement plus élevée que la perte d'entraînement ce qui peut indiquer un léger surapprentissage.

Modèle 2

La perte d'entraînement diminue également de manière continue. La perte de validation suit la même tendance, ce qui est considéré comme un bon indicateur montrant que le modèle apprend, améliore ses performances et se généralise bien, c'est-à-dire qu'il prédit plus correctement les zones pertinentes.

Comparaison

Le modèle 2 présente une perte globale plus faible, tant à l'entraînement qu'à la validation, avec moins de fluctuations, ce qui traduit un apprentissage plus stable et robuste.

4.10.3 Matrice de confusion

Pour évaluer les performances du modèle final sur des images test indépendantes, une matrice de confusion a été générée image par image. Cette matrice permet de calculer les métriques suivantes : True Positive (TP), True Negative (TN), False Positive (FP), et False Negative (FN). À partir de ces valeurs, plusieurs indicateurs de qualité ont été dérivés, notamment la précision (Précision), le rappel (Rappel), le F1-score, et l'Intersection over Union (IoU).

Le Tableau 4.5 présente les résultats détaillés de cette évaluation pour plusieurs images test, où l'on observe les valeurs de TP, TN, FP, FN ainsi que les métriques calculées pour chaque image :

Tableau 4.2: Résultats de l'évaluation des performances par image

	TP	TN	FP	FN	Précision	Rappel	F1-score	IoU
4.png	549	218914	22	7	0.961471	0.987410	0.974268	0.949827
48.png	350	219024	42	76	0.892857	0.821596	0.855746	0.747863
5.png	124	219326	40	2	0.756098	0.984127	0.855172	0.746988
6.png	108	219340	5	39	0.955752	0.734694	0.830769	0.710526
50.png	118	219325	48	1	0.710843	0.991597	0.828070	0.706587

La Figure 4.26 illustre la matrice de confusion pour deux images test représentatives.

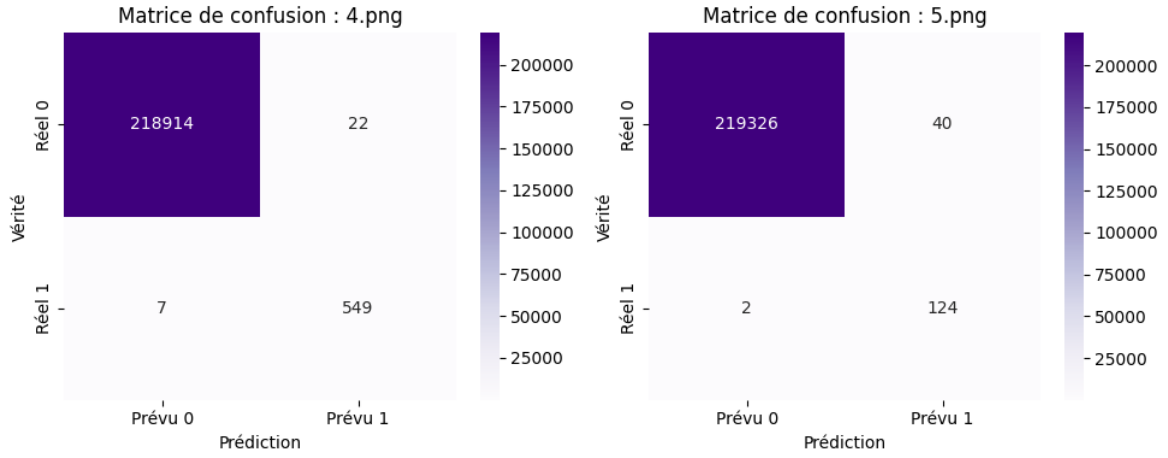


Figure 4.26 : Matrice de confusion pour 2 images test

L'analyse des résultats montre que dans la majorité des cas, le F1-score est supérieur à 0.85, ce qui indique un très bon compromis entre rappel et précision.

De plus, un IoU élevé, pouvant atteindre jusqu'à 0.95, démontre la capacité du modèle à localiser précisément les micro-anévrismes, malgré leur très petite taille.

Cette précision est également clairement visible lorsqu'on compare les valeurs de True Positives (TP) et True Negatives (TN) avec celles de False Positives (FP) et False Negatives (FN), où l'on constate que TP et TN sont nettement supérieurs, ce qui signifie que le taux d'erreur est faible.

Ces résultats confirment ainsi l'efficacité du modèle dans la segmentation fine des zones pathologiques sur les images du fond d'œil.

4.11 Points forts du modèle développé

Au terme de cette phase expérimentale, plusieurs points forts du modèle développé ont été identifiés :

- Capacité de généralisation : Grâce à l'augmentation des données et à la diversité introduite via des techniques telles que le la suppression du canal rouge, le modèle a appris à généraliser correctement même sur des images inédites.
- Robustesse face à la variabilité des images : L'architecture U-Net combinée à une base de données bien annotée et à la suppression du canal rouge permet une segmentation stable malgré les variations d'éclairage, de contraste et de structure anatomique entre les patients.
- Modularité et portabilité : Le modèle peut être sauvegardé et rechargé rapidement grâce au format .h5, facilitant son déploiement dans une application médicale ou une future phase de validation clinique.
- Optimisation des ressources : L'utilisation d'un générateur de données permet un entraînement efficace même sur des plateformes limitées comme Google Colab, tout en assurant une bonne gestion de la mémoire.
- Validité clinique : Les résultats obtenus ont été validés par une médecin ophtalmologue, confirmant la pertinence médicale du système proposé.

4.12 Avantages du dispositif optique

Ce dispositif offre plusieurs avantages cliniques importants :

- Ciblage précis : Le système permet de traiter uniquement les zones pathologiques, préservant ainsi les tissus sains.
- Traitement simultané : Grâce à la modulation du faisceau laser par le DMD, toutes les zones affectées peuvent être traitées en une seule projection, réduisant ainsi le temps de traitement.
- Réduction des erreurs humaines : L'automatisation du processus minimise les risques d'erreurs associées aux traitements manuels.
- Confort accru pour le patient : Le traitement simultané et précis réduit l'inconfort et les incidents potentiels pour le patient

4.13 Perspectives d'évolution

Même si notre système est actuellement fonctionnel, nous identifions plusieurs axes d'amélioration :

- L'agrandissement de la base de données.
- L'application de notre méthode de segmentation à d'autres pathologies rétiniennes.
- Le développement d'un système fonctionnant en temps réel.
- L'expérimentation de fonctions de perte plus complexes pour optimiser les performances.

4.14 Conclusion

Dans ce chapitre, nous avons implémenté deux modèles U-Net

- Le modèle U-Net original que nous avons entraîné sur une base de données de micro-anévrismes conventionnelle
- Un modèle U-Net, dans lequel l'encodeur a été remplacé par VGG16 préentraîné, que nous avons entraîné sur une base de donnes modifiée, d'abord par la suppression du canal rouge, pour améliorer le contraste, et ensuite par la division des images par 16, pour zoomer sur chaque section et mettre en évidence les micro-anévrismes de taille réduite.

La comparaison avec le modèle U-Net a prouvé que notre modèle amélioré et la base de données développée présentent une puissance et une efficacité supérieures, ce qui se traduit par une meilleure qualité et précision de la segmentation, détectant des micro-anévrismes non détectés par l'ophtalmologue elle-même.

L'expérience pratique utilisant le DMD a validé le principe d'appliquer le modèle dans un environnement optique réel, pour les applications de photocoagulation multispot.

Ce dispositif optique, basé sur l'utilisation de l'image segmentée N/B pour le contrôle du DMD, constitue un pont essentiel entre les résultats de segmentation automatique produits par le modèle U-Net et leur exploitation physique par projection laser, pour la photocoagulation rétinienne.

Conclusion générale

À travers ce travail, nous avons abordé un problème médical majeur : la détection des micro-anévrismes liés à la rétinopathie diabétique en utilisant l'architecture U-Net, l'une des principales causes de cécité dans le monde.

L'utilisation de la plateforme de google Colab nous a permis de tirer parti de la puissance du cloud computing pour mener efficacement les expériences

Après avoir constaté l'inefficacité du modèle U-Net associé à une base de données conventionnelle, nous avons apportés plusieurs améliorations qui ont contribué à optimiser les performances du modèle, telles que l'augmentation des données (en divisant chaque image en sous-images), l'utilisation du Transfer Learning avec VGG-16, ainsi que des techniques de traitement d'image (suppression du canal rouge). La détection de micro-anévrismes non détectés par l'ophtalmologue est un excellent résultat, car il promet d'offrir aux ophtalmologues un outil de détection des micro-anévrismes sur des images de fond d'œil, résultat immédiat et applicable sans avoir systématiquement recours aux angiographies, examens invasifs et coûteux.

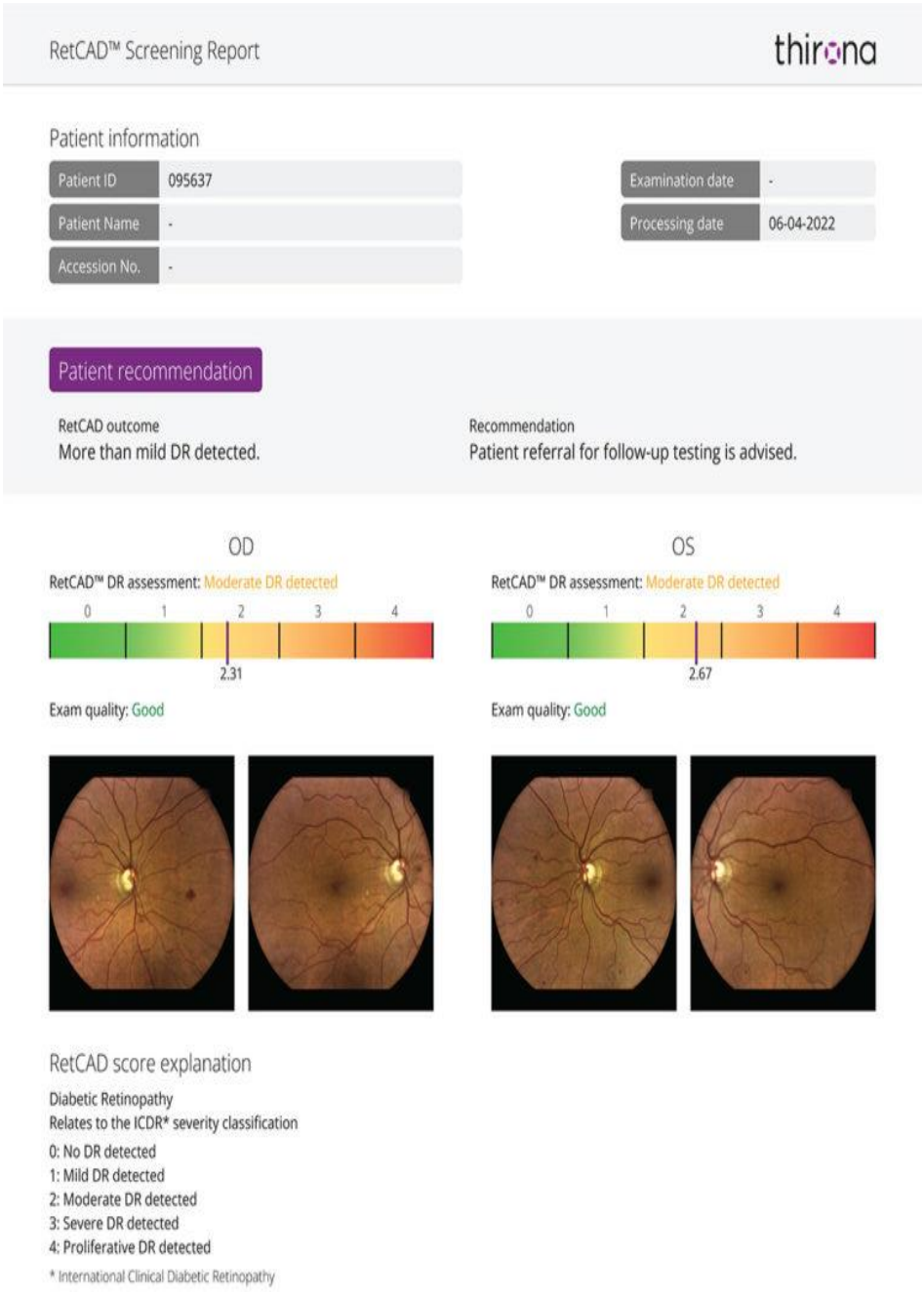
La phase de simulation et d'application pratique a permis de valider les choix méthodologiques adoptés. Les résultats obtenus ont montré que notre méthodologie offre de bonnes performances en termes de précision dans la détection des micro-anévrismes. Cette précision a été illustrée concrètement grâce au développement d'un système optique pour simuler et valider les résultats dans des conditions réelles, et l'utilisation de l'image segmentée pour contrôler le positionnement de rayons lasers simultanés sur une cible par le DMD, permettant ainsi de valider la photocoagulation multispot.

Cependant, le modèle présente certaines limites qu'il convient de souligner. Malgré l'utilisation de la technique d'augmentation de données, la base de données reste

relativement petite, ce qui limite la capacité de généralisation du modèle. En outre, la taille des micro-anévrismes est très réduite par rapport à l'arrière-plan de l'image, ce qui nécessite normalement l'usage de fonctions de perte complexes comme Tversky ou Focal Tversky pour obtenir des résultats précis, au lieu de la fonction dice loss imposée par les contraint de temps de colab .

Ce projet ouvre des perspectives prometteuses pour de futurs développements dans le domaine de l'intelligence artificielle, de l'imagerie médicale et de l'instrumentation médicale, notamment par l'intégration de réseaux neuronaux plus profonds, l'utilisation de bases de données plus vastes et le renforcement de la collaboration de l'équipe biorétine avec les professionnels de la santé. Afin de favoriser une application pratique et de transformer les soins ophtalmologiques, améliorant ainsi la qualité de vie des patients, le système propose la partie intelligence artificielle, et la partie optique pourrait être embarquée et connectée.

Annexe 1 : Dépistage de la rétinopathie diabétique avec iCare ILLUME



RetCAD™ Screening Report

thirona

Patient information

Patient ID	095637	Examination date	-
Patient Name	-	Processing date	06-04-2022
Accession No.	-		

Patient recommendation

RetCAD outcome
More than mild DR detected.

Recommendation
Patient referral for follow-up testing is advised.

OD

RetCAD™ DR assessment: Moderate DR detected

0 1 2 3 4

2.31

Exam quality: Good

OS

RetCAD™ DR assessment: Moderate DR detected

0 1 2 3 4

2.67

Exam quality: Good

RetCAD score explanation

Diabetic Retinopathy
Relates to the ICDR* severity classification

0: No DR detected
1: Mild DR detected
2: Moderate DR detected
3: Severe DR detected
4: Proliferative DR detected

* International Clinical Diabetic Retinopathy

Annexe 2 : Entraînement du modèle U-Net avec stratégie de régularisation et sauvegarde des meilleurs poids

```
callbacks = [
    ReduceLROnPlateau(monitor="val_loss", factor=0.5, patience=3, verbose=1),
    EarlyStopping(monitor="val_loss", patience=5, restore_best_weights=True),
    ModelCheckpoint("/content/drive/MyDrive/model12042025.h5",
                    monitor="val_loss",
                    save_best_only=True,
                    save_weights_only=False,
                    verbose=1)
]

initial_learning_rate = 1e-3
batch_size = 4

train_generator = DataGenerator(train_images, train_masks, batch_size=batch_size)
valid_generator = DataGenerator(valid_images, valid_masks, batch_size=batch_size)

model.compile(optimizer=Adam(1e-4), loss=dice_loss, metrics=["accuracy"])

history = model.fit(train_generator,
                    validation_data=valid_generator,
                    epochs=50,
                    callbacks=callbacks)
```

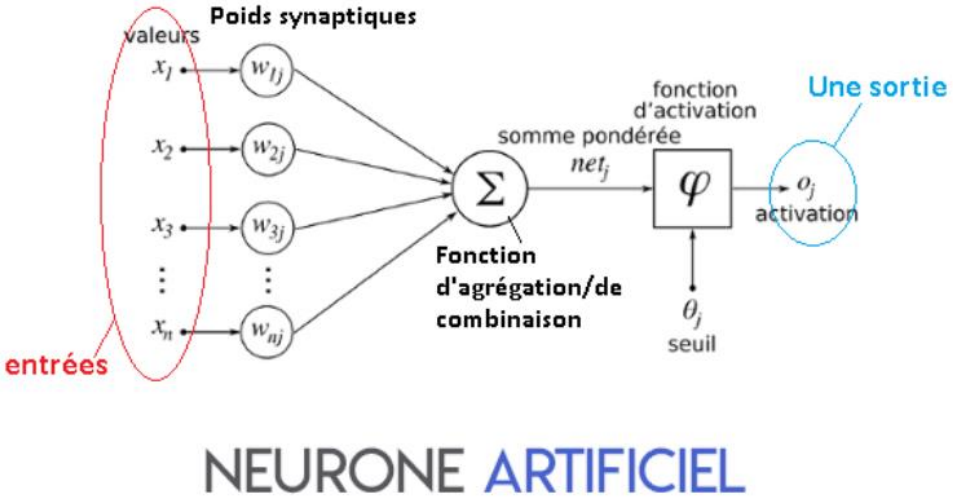
Annexe 3 : Les neurones artificiels et Le perceptron simple (monocouche)

Les neurones artificiels

Un neurone artificiel est une unité de base dans un réseau de neurones artificiels (RNA), inspirée du fonctionnement des neurones biologiques.

Il reçoit :

- Un ensemble d'entrées $(x_1, x_2, \dots, x_n)$, représentant les données ou les caractéristiques extraites du problème étudié.
- Des poids associés à chaque entrée $(w_1, w_2, \dots, w_n)$, qui déterminent l'importance de chaque entrée dans la prise de décision.
- Une somme pondérée des entrées, calculée comme suit : $S = \sum_{i=1}^{n} w_i . x_i$
  où b est le biais (bias), qui aide à ajuster la sortie du modèle.
- Une fonction d'activation (Activation Function), telle que ReLU, Sigmoid ou Tanh, qui détermine la sortie du neurone en fonction des valeurs calculées.

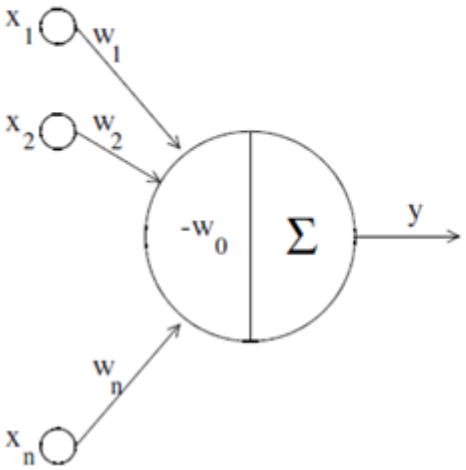


NEURONE ARTIFICIEL

Le perceptron simple (monocouche)

- Un neurone possède des entrées X_i
- Chaque entrée possède un poids W_i
- Fonctions d'activation (ou fonction de transfert) f
- La sortie est une fonction Y du poids et des entrées :
  $Y = f(W_1 * X_1 + W_2 * X_2)$
  la sortie peut avoir des valeurs 0 (excite) et 1 (inhibe)
- Le neurone a un seuil (threshold). Il s'excite si la somme pondérée des entrées > seuil

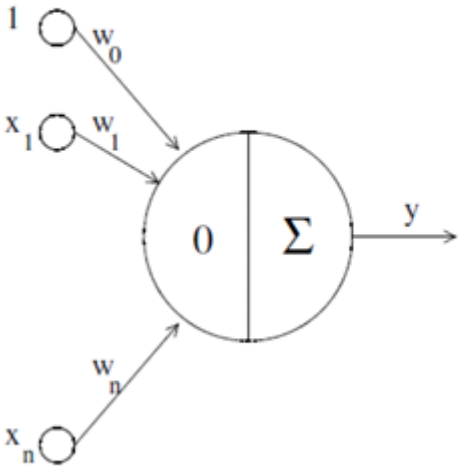
Neurones avec seuil



Neurone avec seuil $-w_0$

$$y = \begin{cases} 1 & \sum_{j=1}^{n} w_j x_j > -w_0 \\ 0 & \sum_{j=1}^{n} w_j x_j \leq -w_0 \end{cases}$$

Neurones avec biais



- Neurone avec biais w_0. Soit

$$x' = \langle 1 \mid x \rangle^T$$

- Alors

$$y = \begin{cases} 1 & \sum_{j=0}^{n} w_j x_j > 0 \\ 0 & \sum_{j=0}^{n} w_j x_j \leq 0 \end{cases}$$

- ou $\quad y = \begin{cases} 1 & w^T x' > 0 \\ 0 & w^T x' \leq 0 \end{cases}$

Classification : perceptrons

Le perceptron simple ne peut résoudre que des problèmes linéairement séparables. Pour aller plus loin, il est nécessaire d'ajouter des couches.

Annexe 4 : (Bibliothèques utilisées)

Bibliothèques	Utilité
os	Gérer les fichiers et dossiers (ouverture, enregistrement, navigation).
numpy	Manipuler les données numériques, notamment pour représenter les images en matrices.
cv2(OpenCV)	Lire, modifier et traiter les images (rognage, redimensionnement, filtres…).
glob	Rechercher des fichiers image dans un dossier selon un motif donné (*.jpg).
tqdm	Afficher une barre de progression lors de l'exécution de boucles longues.
imageio	Lire et sauvegarder des images dans différents formats.
albumentations	Appliquer des transformations d'augmentation de données pour enrichir le dataset.
tensorFlow	Construire et entraîner des modèles de deep learning pour l'analyse d'images.
keras	Créer des modèles d'intelligence artificielle facilement via TensorFlow
matplotlib	Visualiser les images, les graphes et suivre les performances du modèle

Bibliographies

[1] American Academy of Ophthalmology, *Diabetic Retinopathy Preferred Practice Pattern*, San Francisco, CA, USA: AAO, 2019.

[2] S. Majumder and N. Kehtarnavaz, "Stages of diabetic retinopathy," ResearchGate, 2021. [Online]. Available: https://www.researchgate.net/ [Consulté le : février 2025]

[3] American Academy of Ophthalmology, *Diabetic Retinopathy Preferred Practice Pattern*, San Francisco, CA, USA: AAO, 2019.

[4] M. Duh, J. Sun, and T. Stitt, "Diabetic Retinopathy: Current Understanding, Mechanisms, and Treatment Strategies," *Journal of Ophthalmology*, vol. 2017, Article ID 9648653, 18 pages, 2017.

[5] iCare World, "iCare DRSplus," iCare World, 2024. [Online]. Available: https://www.icare-world.com/ [Consulté le : mars 2025]

[6] Alamy, "Banque d'images et photographies d'archives," Alamy. [Online]. Available: https://www.alamy.com/ [Consulté le : mars 2025]

[7] 123RF, "Banque d'images libres de droits," 123RF. [Online]. Available: https://www.123rf.com/ [Consulté le : mars 2025]

[8] International Council of Ophthalmology, *ICO Guidelines for Diabetic Eye Care*, 2017.

[9] IRIDEX Corporation, *Manuel d'utilisation – IQ 577 / IQ 532 Laser Systems*, 15510G_FR, 2021.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[13] Diabolocom, "Natural Language Processing (NLP): A Complete Guide," Diabolocom, Dec. 19, 2023. [Online]. Available: https://www.diabolocom.com/ [Consulté le: mars 2025]

[14] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Machine Learning (ICML)*, 2010, pp. 807–814.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[16] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vision (3DV)*, 2016, pp. 565–571.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[18] S. R. Syaukani, "Dolphin and Sharks Classification using Convolutional Neural Network (CNN)," *Medium*, Jan. 7, 2024. [Online]. Available: https://medium.com/ [Consulté le : mars 2025]

[19] A. D. Elster, "Upsampling – Questions and Answers in MRI," MRIQuestions.com, 2024. [Online]. Available: https://mriquestions.com/ [Consulté le : mars 2025]

[20] A. M. Ibrahim, M. Elbasheir, S. Badawi, and A. F. M. Alalmin, "Skin Cancer Classification Using Transfer Learning by VGG16 Architecture (Case Study on Kaggle Dataset)," ResearchGate, Jan. 2023. [Online]. Available: https://www.researchgate.net/ [Consulté le : avril 2025]

[21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint*, arXiv:1409.1556, 2014.

[22] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. doi: 10.1109/TKDE.2009.191.

[23] E. Sleightholm, "Getting Started with Google Colab: A Beginner's Guide," Marqo, Jun. 10, 2024. [Online]. Available: https://www.marqo.ai/ [Consulté le : mars 2025]

[24] Evosens, "Digital Micromirror Device : les micromiroirs au service de l'optique," Evosens. [Online]. Available: https://www.evosens.fr/ [Consulté le : avril 2025]

[25] Texas Instruments, *DLP7000 DLP® 0.7 XGA 2x LVDS Type A DMD*, Rev. B, 2024. [Online]. Available: https://www.ti.com/ [Consulté le : avril 2025]

[26] J. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Springer, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[27] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[28] A. Agarwal, "Understand Confusion Matrix in AI," Innovatiana, Nov. 17, 2020. [Online]. Available: https://www.innovatiana.com/post/understand-confusion-matrix-in-ai [Consulté le : mars 2025]

[29] COSS Ophtalmologie, Pathologies de la rétine et du vitré. [En ligne]. Disponible sur : https://www.coss-ophtalmologie.paris/pathologies/pathologies-de-la-retine-et-du-vitre/ [Consulté le : mars 2025]

[30] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, et N. Killoran, « Transfer learning in hybrid classical-quantum neural networks », Quantum, vol. 4, p. 340, oct. 2020. Disponible sur: https://doi.org/10.22331/q-2020-10-09-340 [Consulté le: avril 2025]

[31] Oculista Falchi Paolo Siracusa, *« LASER ARGON ET JAUNE »*, [en ligne]. Disponible sur : https://www.oculistafalchipaolo.it/laser-argon-e-giallo/. [Consulté le: juin 2025]