



République du Sénégal

Un Peuple – Un But – Une Foi

Ministère l'Enseignement Supérieur de la Recherche et de l'Innovation

Ecole Polytechnique de Thiès

B.P.A 10 Thiès

Tel: (221) 76 223 61 74 – Fax: (221) 33 951 14 67



---

## **Rapport de projet Deep Learning**

# **Skin Disease Autoencoder Classification**

---

**Présenté par :**  
Mohamed Massamba SENE

**Professeurs :** Dr [Ndeye Fatou NGOM](#) & Dr [Michel SECK](#)

## Contents

Partie 1 : Etude de l'article choisi.....	3
A. Contexte et Problématique .....	3
B. Objectifs et contributions.....	3
C. Etat de l'art .....	4
D. Données utilisées.....	5
E. Méthodologie et Outils.....	5
F. Analyse des résultats.....	9
Partie 2: Proposition d'une architecture.....	10
A. Méthodologie.....	10
B. Implémentation .....	14
C. Analyse des résultats.....	18
D. Application Pratique .....	20
Webographie.....	25

# Partie 1 : Etude de l'article choisi

L'article choisi est A Scheme for Effective Skin Disease Classification using Optimized Region Growing Segmentation and Autoencoder based Classification avec comme auteurs:

- **Dasari Anantha Reddy**, Département de technologie d'information, NIT, Raipur, Inde
- **Sanjay Kumar**, Département de technologie d'information, NIT, Raipur, Inde
- **Rakesh Tripathi**, Département de technologie d'information, NIT, Raipur, Inde
- **Swarup Roy**, Département d'applications informatiques, Sikkim University, Gangtok, Inde

## A. Contexte et Problématique

La peau est une partie essentielle du corps humain car elle nous protège des rayonnements nocifs, de la chaleur, des blessures et des infections par les rayons UV et génère également de la vitamine D dans le corps.

Les maladies de la peau sont considérées comme un type général de maladie dont souffrent principalement les personnes âgées. Ces maladies sont principalement identifiées par l'expertise des médecins et également par les résultats des biopsies cutanées mais cela entraîne une consommation de temps élevée. De plus, la détection des troubles cutanés simplement par inspection visuelle est difficile en raison de la nature complexe et superposée des lésions malades, des textures de fond de la peau, des poils, du faible éclairage, etc.

La classification des troubles cutanés à l'aide d'une image est une tâche difficile qui repose fortement sur les caractéristiques des maladies particulières pour classer avec précision la maladie. Mais la plupart des maladies de la peau présentent des caractéristiques similaires dans la perspective visuelle, ce qui rend difficile la sélection des caractéristiques essentielles de l'image.

L'analyse précise des maladies de la peau à partir de l'image est nécessaire pour améliorer le diagnostic et accélère également le temps de diagnostic. Cela permettrait en outre d'obtenir un traitement amélioré et rentable pour les patients concernés.

## B. Objectifs et contributions

L'objectif de cet article est de fournir une nouvelle approche de détection pour améliorer le diagnostic des troubles cutanés.

Il propose une approche de bout à bout en commençant par la segmentation et l'extraction de caractéristiques des lésions malades puis l'utilisation d'un modèle de classification basé sur un encodeur automatique. Les lésions malades sont segmentées à l'aide de la croissance de région optimisée à l'aide de l'optimisation Grey Wolf (GWO). Les caractéristiques de texture sont extraites de la lésion segmentée à l'aide de la matrice de cooccurrence de niveaux de gris (GLCM) et du descripteur local Weber (WLD).

Enfin, un ensemble de fonctionnalités réduit est généré via une représentation latente à l'aide de l'auto-encodeur. Un réseau neuronal convolutionnel intégré est utilisé pour classer les lésions malades à partir d'une représentation latente.

Globalement l'article apporte les contributions suivantes:

- Un modèle de classification des troubles cutanés utilisant la segmentation et la classification basée sur un auto-encodeur
- Une segmentation efficace utilisant la croissance de région optimisée en ajustant le seuil de l'image segmentée à l'aide de l'algorithme GWO
- Utilisation des caractéristiques de texture de la lésion segmentée à l'aide de la GLCM et de WLD
- Utilisation d'un auto-encodeur pour la réduction des caractéristiques pour la classification à l'aide d'un CNN
- 

## C. Etat de l'art

Des approches informatiques utilisant les méthodes de Machine Learning et Deep Learning ont été proposées mais elles sont également limitées dans leur capacité à détecter des structures de lésions complexes.

Modèles	Utilisation de la segmentation	Modèle de base	Challenge
Deep CNN	Non	ResNet512	Difficulté à générer des cartes de densité raisonnable pour atteindre des résultats similaires aux données réelles
GP-CNN	Non	CNN	Énorme quantité de données nécessaires pour un modèle robuste

FCN	Encoder-Decoder	DenseNet	Ne tient pas compte des détails dans l'image et tend à lisser les structures détaillées
KOE-CT	Non	MobileNet-V3	Les hyperparamètres doivent être ajustés manuellement suivant le jeu de données
Cycle-GAN	Non	ImageNet	Les performances ne sont pas satisfaisantes et il ne tient pas compte des caractéristiques discriminatoires

Tableau 1 : Etat de l'art des modèles existants et les challenges qu'ils rencontrent

## D. Données utilisées

L'ensemble de données utilisées est PH<sup>2</sup> qui est un jeu de données développé à des fins de recherche et d'analyse comparaison, afin de faciliter les études comparatives sur les algorithmes de segmentation et de classification des images dermoscopiques acquises au service de dermatologie de l'hôpital Pedro Hispano, Matosinhos, Portugal.

- Résolution: 765 x 572
- Format: BMP
- Source: <https://www.fc.up.pt/addi/ph2%20database.html>

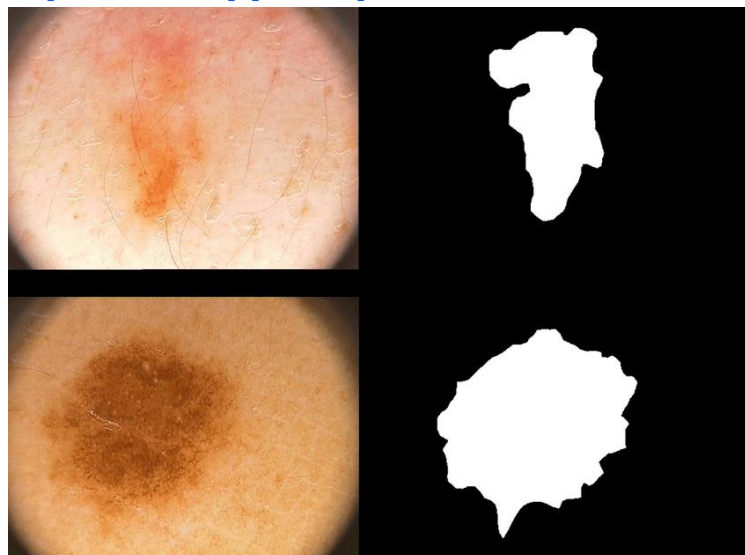


Fig 1 : Extrait d'images du PH<sup>2</sup>

## E. Méthodologie et Outils

La méthodologie proposée est représentée dans le schéma ci-dessous :

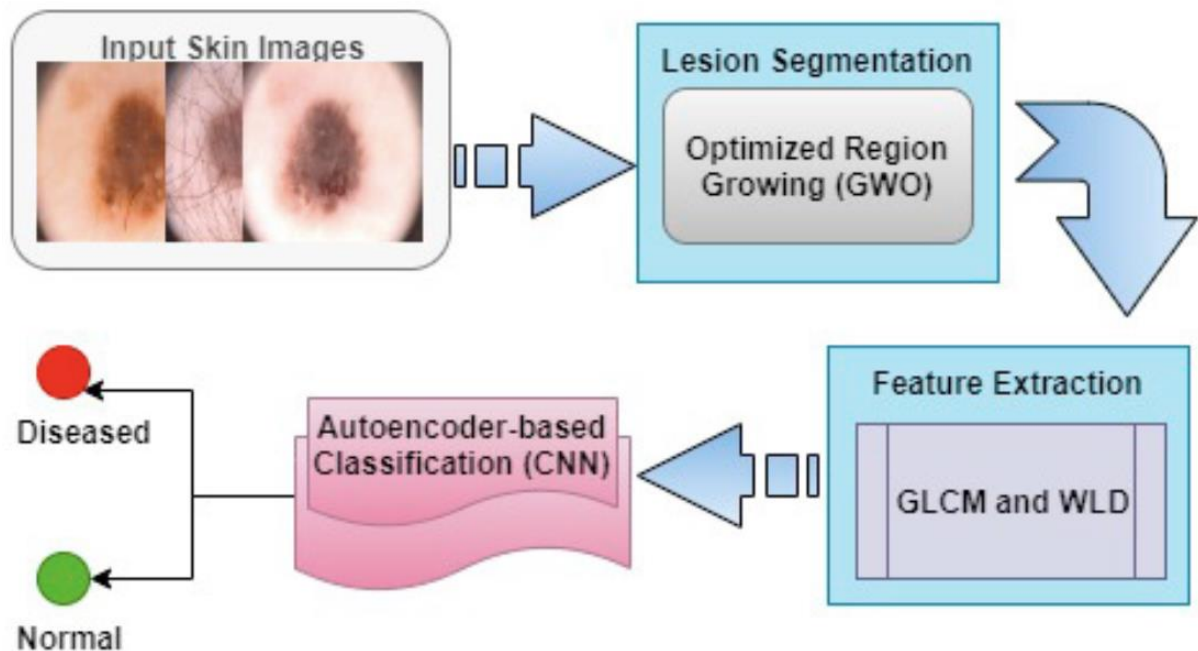


Fig 2 : Schéma de la méthodologie proposée dans l'article

Comme nous pouvons le voir, le modèle de classification a été développé en effectuant une segmentation basée sur la croissance en région et un modèle de classification basé sur les auto encoder.

Au début, les images sont récupérées au niveau de la base de données et assujetties à une segmentation basée sur la croissance en région. La performance de la segmentation est améliorée en optimisant la fonction de seuillage de la technique de croissance en région avec le Grey Wolf Optimization (GWO). Ensuite les caractéristiques les plus importantes sont extraites en utilisant le Gray Level Co-occurrence Matrix (GLCM) et Weber Local Descriptor (WLD). Pour finir les caractéristiques du GLCM et WLD sont concaténés et utilisés pour la classification. La classification est effectuée par un auto encodeur qui identifie si l'image contient une maladie ou non.

### **Segmentation avec la croissance en région optimisée avec GWO**

La segmentation basée sur la croissance des régions est une technique de vision par ordinateur utilisée pour partitionner une image en régions ou segments en fonction de certains critères de similarité. L'idée de base est de commencer avec un point de départ initial (ou plusieurs points de départ) et d'agrandir la région de manière itérative en incluant les pixels voisins qui répondent à un critère de similarité prédéfini.

L'algorithme est le suivant :

- Initialisation

Choisir un point de départ initial ou plusieurs points de départ dans l'image.

Définir un critère de similarité ou une mesure d'homogénéité qui détermine quand un pixel doit être ajouté à la région.

Initialiser une région vide avec le(s) point(s) de départ.

- Itération

Pour chaque pixel de la région, examinez ses voisins généralement 4 ou 8 connectivités.

Calculez la similarité entre le pixel et ses voisins en utilisant le critère de similarité choisi.

Si la similarité entre le pixel et un voisin est inférieure à un certain seuil et que le voisin ne fait pas déjà partie de la région, ajoutez-le à la région.

Répétez ce processus jusqu'à ce que plus aucun pixel ne puisse être ajouté à la région.

- Fin

Le processus de croissance de la région s'arrête lorsqu'aucun pixel ne peut plus être ajouté à la région, c'est-à-dire que tous les voisins ne répondent pas au critère de similarité.

En option, vous pouvez appliquer des étapes de post-traitement telles que la fusion de régions ou l'affinement des limites pour améliorer la qualité de la segmentation.

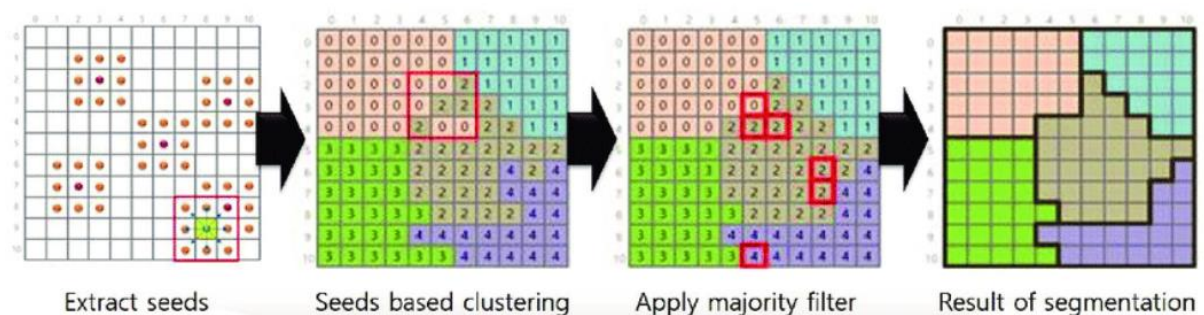


Fig 3 : Exemple du processus de croissance en région avec plusieurs points de départs

Grey Wolf Optimization est un algorithme basé sur la population inspiré du comportement de chasse des loups gris. Les trois phases principales du comportement de chasse des loups gris sont le pistage, la poursuite et la chasse. S'approcher de la proie, la poursuivre, l'encercler et la harceler jusqu'à ce qu'elle s'arrête de bouger et attaque la proie.

Il est utilisé pour optimiser la fonction de seuillage utilisée dans la segmentation en région croissante en trouvant les seuils qui permettent d'obtenir les meilleures segmentations.

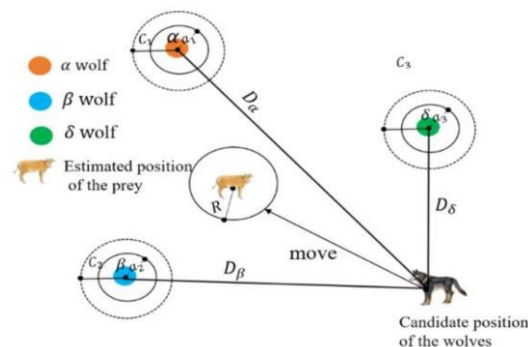


Fig 4 : Grey Wolf Optimization

En optimisant les paramètres de croissance de la région avec Grey Wolf Optimization, vous pouvez affiner l'algorithme en fonction des caractéristiques spécifiques de vos données d'image, ce qui conduit à de meilleurs résultats de segmentation.

### Extraction des caractéristiques avec WLD et GLCM

Le descripteur local de Weber (Weber Local Descriptor en anglais) est une technique de traitement d'image utilisée pour représenter localement les caractéristiques d'une région dans une image. Il est basé sur la mesure de la variation de l'intensité des pixels dans une région par rapport à un point de référence.

Il commence par choisir un point de référence au sein de la région d'intérêt. Ensuite pour chaque pixel de la région, la différence entre ce pixel et le point de référence est calculée. Les différences sont ensuite codées sous forme de vecteur ou d'histogramme.

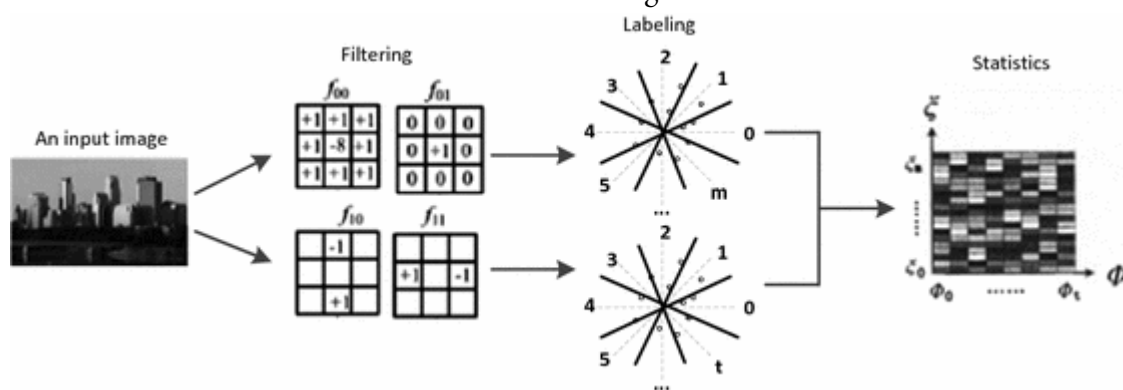


Fig 5: Weber Local Descriptor

La Matrice de Co-occurrence des Niveaux de Gris (Gray Level Co-occurrence Matrix en anglais, souvent abrégée en GLCM) est une technique d'analyse d'image utilisée pour extraire des informations sur la texture telles que le contraste, l'énergie, la dissimilarité, l'entropie et la corrélation et la structure spatiale des niveaux de gris d'une image

Tout d'abord la matrice est construite en utilisant les relations spatiales entre les niveaux de gris des pixels. Ensuite on spécifie les paramètres à considérer et à partir de mesures on extrait les caractéristiques de texture.

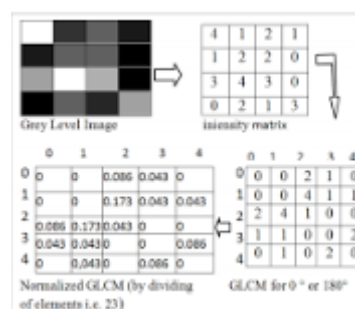


Fig 6: Gray Level Co-occurrence Matrix

### Réduction des caractéristiques par AutoEncoder



Au total 24 caractéristiques GLCM et 65 536 caractéristiques WLD concaténées pour former un grand vecteur de caractéristiques de taille 65 560. Le vecteur est réduit en utilisant la représentation latente avant d'alimenter le classificateur.

Un codeur est un processus qui se produit entre la couche d'entrée et la couche cachée. L'image est reconstruite après les fonctionnalités compressées des couches cachées sont transmises à la couche de sortie. Un décodeur est utilisé pour traiter les fonctionnalités entre les couches cachées et la sortie couches. Dans l'approche de l'auto-encodeur, la fonction coût est le paramètre le plus important, qui doit être maintenu au minimum pour obtenir une classification précise.

Le modèle d'encodeur automatique proposé peut être élaboré selon les principes du réseau neuronal à convolution (CNN), en entrée couche avec des images segmentées, alambiquée de 3 couches cachées et couche entièrement connectée avec fonction d'activation linéaire.

## F. Analyse des résultats

Une analyse comparative des performances de l'architecture proposée a été faite avec des modèles comme CNN, RNN, LSTM, Cycle-GAN... en mesurant plusieurs métriques suivantes dont les plus pertinentes :

- Accuracy : permet de renseigner sur le pourcentage d'images correctement classifiés par le modèle
- Sensitivity : permet de mesurer combien d'instances positives réelles ont été correctement classées.
- Precision : permet de mesurer combien d'instances positivement classées sont réellement correctes.
- F1-Score : moyenne harmonique de précision et de rappel, qui donne une mesure équilibrée des performances d'un modèle
- Specificity: permet de mesurer la proportion de véritables prédictions négatives parmi toutes les instances négatives réelles

Comme nous pouvons le voir dans le tableau ci-dessous, le modèle proposé donne de meilleurs résultats que tous les autres modèles précédemment cités.

Measures	Cycle-GAN [5]	GP-CNN [2]	KDE-CT [4]	CNN [16]	RNN [18]	LSTM [17]	Proposed
"Accuracy"	0.910744	0.91939	0.929816	0.864	0.892	0.91	<b>0.942</b>
"F1-score"	0.629017	0.649701	0.675205	0.854077	0.888889	0.906054	<b>0.939457</b>
"FDR"	0.51755	0.494957	0.467259	0.099548	0.103734	0.07265	<b>0.038462</b>
"FNR"	0.096503	0.08951	0.078322	0.187755	0.118367	0.114286	<b>0.081633</b>
"FPR"	0.096923	0.089231	0.080839	0.086275	0.098039	0.066667	<b>0.035294</b>
"Sensitivity"	0.903497	0.91049	0.921678	0.812245	0.881633	0.885714	<b>0.918367</b>
"Specificity"	0.903077	0.910769	0.919161	0.913725	0.901961	0.933333	<b>0.964706</b>
"Precision"	0.48245	0.505043	0.532741	0.900452	0.896266	0.92735	<b>0.961538</b>
"NPV"	0.989427	0.990268	0.991551	0.913725	0.901961	0.933333	<b>0.964706</b>
"MCC"	0.61693	0.637791	0.663961	0.730758	0.783945	0.820566	<b>0.88471</b>

Tableau 2 : Analyse comparatives des performances de l'existant par rapport au modèle proposé

## Partie 2: Proposition d'une architecture

### A. Méthodologie

**Skin Input Images** — **Segmentation U-Net** — **Classification CNN**

Au début, les images sont récupérées au niveau de la base de données et assujetties à une segmentation basée en utilisant U-Net. Ce choix est motivé par les raisons suivantes:

- L'architecture de U-Net est faite pour fournir une localisation précise en capturant le contexte avec un chemin expansif symétrique. Cette conception est bien adaptée aux images biomédicales où les structures d'intérêt peuvent varier considérablement en taille et en forme.
- La segmentation utilisée dans l'analyse d'images biomédicales est une segmentation sémantique où chaque pixel est associé à une classe. U-Net excelle dans la segmentation sémantique en raison de sa nature entièrement convolutive, ce qui le rend capable de classer chaque pixel individuellement.
- Les chercheurs ont affiné les architectures DU-Net pour des applications biomédicales spécifiques, telles que la détection cellulaire, la segmentation d'organes ou la localisation de tumeurs, obtenant ainsi des résultats impressionnants.

Nous omettons l'extraction de caractéristiques avec WLD et GLCM et passons directement les images segmentées à un réseau de neurones convolutifs pour effectuer la classification. Ce choix est motivé par les raisons suivantes:

- Les CNN sont conçus pour apprendre automatiquement les caractéristiques à partir des données brutes de manière hiérarchique. Au lieu d'avoir à concevoir et à extraire manuellement des caractéristiques comme WLD et GLCM, le CNN peut apprendre les caractéristiques pertinentes à partir des données d'entrée, ce qui simplifie considérablement le processus.
- Les CNN sont souvent plus performants que les méthodes traditionnelles d'extraction de caractéristiques sur un large éventail de tâches de vision par ordinateur. Ils peuvent capturer des motifs et des caractéristiques complexes qui pourraient être manqués ou mal modélisés par des méthodes manuelles.

#### **Segmentation avec U-Net**

U-Net est une architecture de réseau de neurones convolutifs profonds (CNN) développée pour la segmentation sémantique, qui consiste à attribuer une étiquette sémantique à chaque pixel d'une

image. Cette architecture a été initialement conçue pour la segmentation d'images biomédicales, mais elle s'est depuis avérée efficace dans de nombreuses autres applications, telles que la détection d'objets, la segmentation d'images satellites, etc.

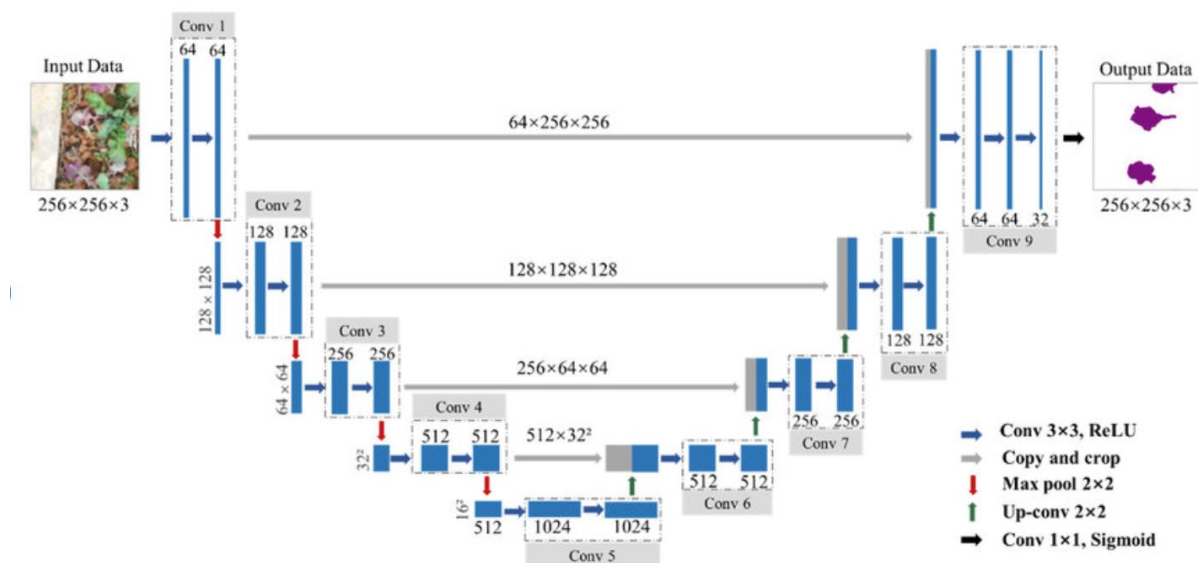


Fig 7 : Architecture U-Net

U-Net tire son nom de sa forme générale qui ressemble à la lettre "U" comme nous pouvons le voir dans la figure ci-dessus...

Elle est composée de deux parties principales : l'encodeur et le décodeur

#### - Encodeur

La partie supérieure du "U" est l'encodeur, qui est similaire à une architecture CNN classique. Il est utilisé pour extraire des caractéristiques pertinentes de l'image en réduisant progressivement sa résolution spatiale (c'est-à-dire sa taille). Cela se fait généralement en utilisant des couches de convolution, de pooling (généralement du max-pooling) et de normalisation (comme la normalisation par lots) successives. Ces couches d'encodeur réduisent la taille de l'image tout en conservant des informations importantes sur les caractéristiques visuelles.

#### - Décodeur

La partie inférieure du "U" est le décodeur, qui est utilisé pour restaurer la résolution spatiale de l'image tout en effectuant la segmentation sémantique. Le décodeur utilise des couches de déconvolution (parfois appelées couches de transposition de convolution) pour agrandir progressivement l'image. Entre ces couches de déconvolution, des informations provenant de l'encodeur sont concaténées pour permettre au modèle de combiner des informations contextuelles à différentes échelles. Cela aide à préciser la segmentation sémantique en utilisant des informations à la fois locales et globales.

Une des caractéristiques clés de l'architecture U-Net est l'utilisation de sauts de connexions entre l'encodeur et le décodeur. Ces sauts de connexions permettent au modèle de conserver des

informations détaillées provenant de l'encodeur, qui seraient autrement perdues lors de la réduction de la résolution spatiale. Ils sont généralement réalisés en concaténant les caractéristiques extraites à différentes étapes de l'encodeur aux caractéristiques correspondantes du décodeur. Cela permet au décodeur d'accéder à des informations de haute résolution tout en bénéficiant de la connaissance contextuelle du modèle.

En résumé l'architecture U-Net les sauts de connexions permettent au modèle de combiner des informations locales et globales, ce qui est essentiel pour la segmentation sémantique, où la compréhension du contexte est cruciale et la forme en "U" facilite la mise en correspondance entre les caractéristiques de l'encodeur et du décodeur, ce qui permet une meilleure correspondance spatiale et une segmentation plus précise.

### Réseau de neurones convolutifs

Les réseaux de neurones convolutifs (CNN), également connus sous le nom de ConvNets, sont une classe de modèles de Deep Learning spécialement conçus pour résoudre des problèmes de vision par ordinateur, tels que la classification d'images, la détection d'objets et la segmentation d'images. Ils sont largement utilisés dans de nombreuses applications, y compris la reconnaissance faciale, la vision par ordinateur autonome (pour les voitures autonomes), la surveillance vidéo, etc. L'architecture des CNN est la suivante :

#### - Couche de convolution

Les CNN commencent par une couche de convolution. Cette couche est la pierre angulaire de l'architecture CNN. Elle utilise des filtres (aussi appelés noyaux) pour effectuer des opérations de convolution sur l'image d'entrée.

Chaque filtre parcourt l'image d'entrée et calcule la somme pondérée des valeurs des pixels sous le filtre. Le résultat est stocké dans une nouvelle matrice, appelée carte de caractéristiques.

Les filtres sont déplacés sur toute l'image d'entrée en effectuant des opérations de convolution pour extraire des caractéristiques importantes telles que les bords, les textures, les formes, etc. Après chaque opération de convolution, une fonction d'activation est appliquée sur la carte de caractéristiques résultante. La fonction d'activation la plus couramment utilisée est la fonction ReLU, qui remplace toutes les valeurs négatives par zéro. L'application de la fonction d'activation introduit de la non-linéarité dans le modèle, ce qui lui permet de capturer des relations plus complexes entre les caractéristiques.

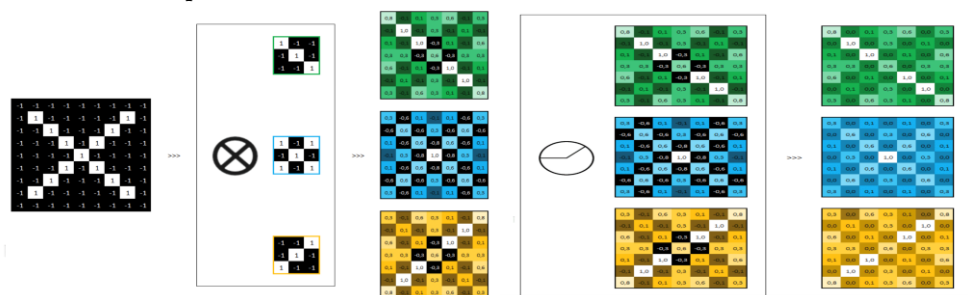


Fig 8 : Convolution et utilisation de ReLu sur les features map

- **Couche de regroupement (Pooling Layer)**

Les couches de regroupement sont utilisées pour réduire la dimension de la carte de caractéristiques tout en préservant les informations essentielles.

La couche de regroupement prend généralement la forme de la max-pooling, où la carte de caractéristiques est divisée en régions et la valeur maximale de chaque région est conservée.

Le pooling réduit la complexité du modèle et rend les caractéristiques invariantes aux petites variations de l'emplacement dans l'image.

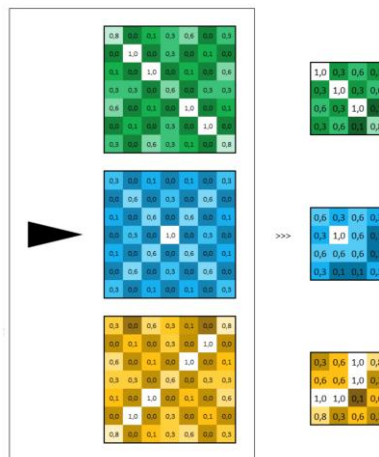


Fig 9 : Résultat de l'application du Pooling à la sortie de la convolution

- **Couche entièrement connectée**

Après plusieurs couches de convolution et de regroupement, une ou plusieurs couches entièrement connectées sont utilisées pour effectuer la classification.

Ces couches sont similaires aux couches d'un réseau neuronal classique et sont utilisées pour combiner les caractéristiques extraites précédemment pour effectuer la classification finale.

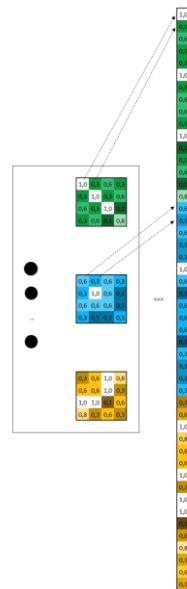


Fig 10 : Sortie du passage des cartes issues du pooling au niveau de la couche entièrement connectée

Voici une synthèse des couches précédemment décrites formant un réseau de neurones

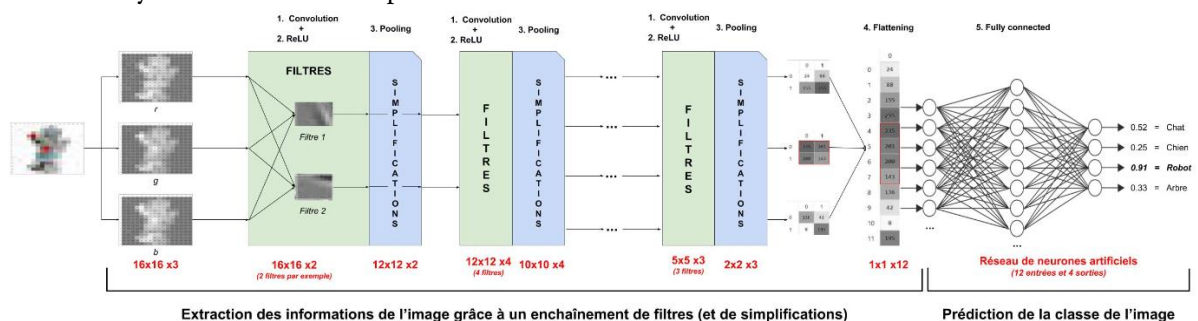


Fig 11 : Exemple d'un CNN

Ainsi les CNN sont une architecture de Deep Learning spécialement conçue pour résoudre des problèmes de classification d'images en automatisant l'extraction de caractéristiques pertinentes à partir des images.

## B. Implémentation

Pour implémenter notre solution, nous utilisons Python avec les bibliothèques suivantes :

OpenCv, Tensorflow, Numpy, Pandas... et procédons en trois étapes.

D'abord nous entraînons le modèle de segmentation basé sur l'architecture U-Net, ensuite nous entraînons le modèle de classification et pour finir nous regroupons les deux modèles dans une même fonction qui fait les traitements afin de prédire directement les maladies à partir de l'image RGB.

### Segmentation U-Net

Pour créer ce modèle nous utilisons le jeu de données PH<sup>2</sup> sur lequel après avoir chargé les images nous effectuons du prétraitement en redimensionnant les images vers une résolution spatiale (224, 224) et en les normalisant.

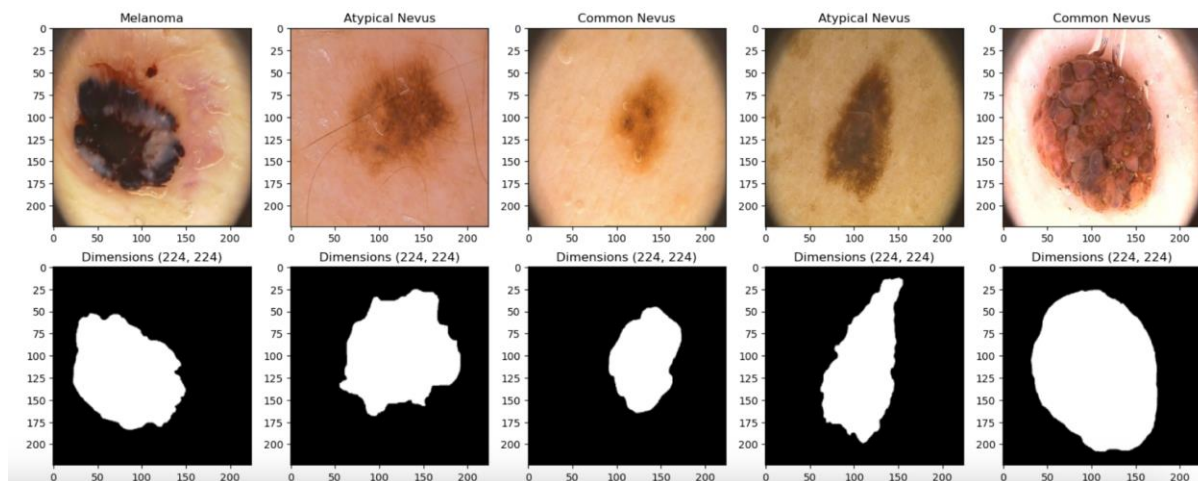


Fig 12 : Images après redimensionnement et normalisation

Ensuite étant donné que le jeu de données est assez petit avec 200 images, nous effectuons du ‘data augmentation’ en appliquant de la rotation et retournement à chaque image ce qui nous donne un jeu de données de 600 images que nous séparons ensuite en données d’entraînement, de validation et de test.

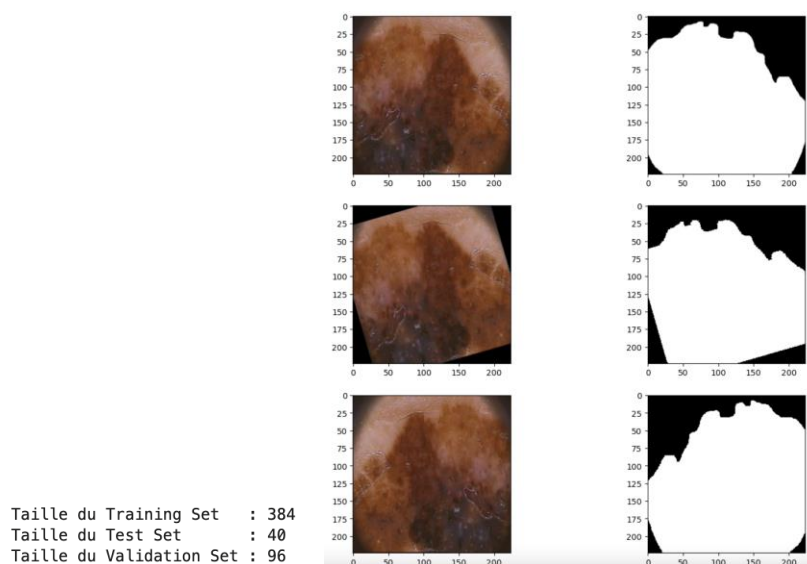


Fig 13 : Data augmentation et taille du jeu final

Nous définissons ensuite le modèle en utilisant comme fonction coût la fonction de Jaccard qui est une mesure de similarité entre deux ensembles. Elle est souvent utilisée pour évaluer la similarité entre deux ensembles de données.

On procède ensuite à l’entraînement et on obtient les résultats suivants :



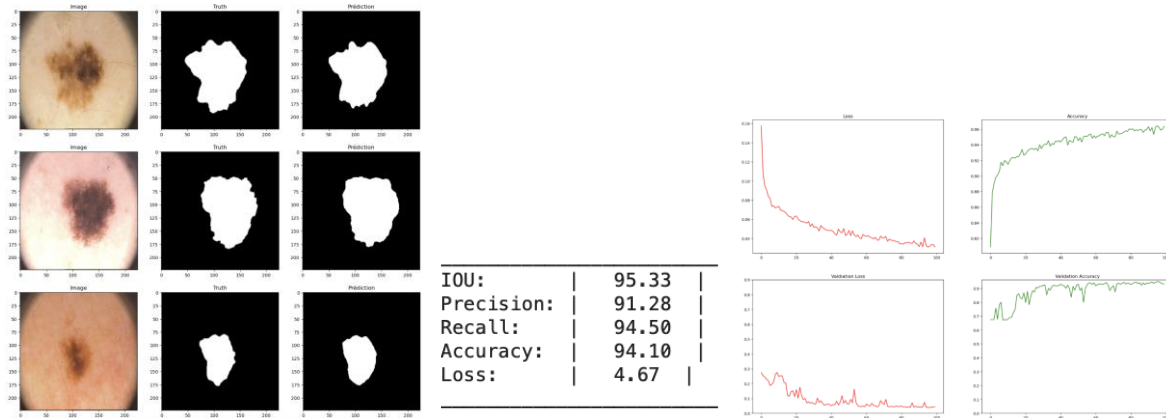


Fig 14 : Résultats du modèle de segmentation

Pour finir on enregistre le modèle

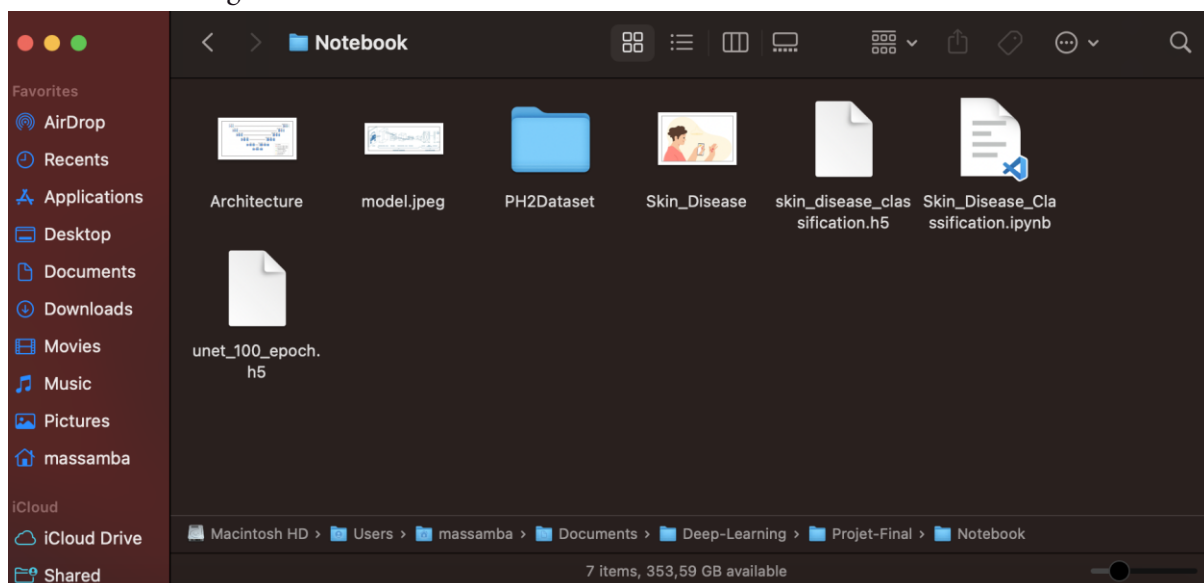


Fig 15 : Modèle de segmentation sauvegardé

## Classification

Pour ce modèle nous avons opté de faire de la classification multi classe plutôt qu'une classification binaire. Pour cela nous avons utilisé le jeu de données HAM10000 qui est une collection de plus de 10000 images biomédicales de lésions cutanées réparties en 7 classes.

Nous avons commencé par charger le jeu de données, rechercher et effacer les duplicatas. Le jeu de données étant déséquilibré avec une classe possédant plus de 7000 des images, nous avons procédé à de l'oversampling pour équilibrer le jeu de données. Pour finir nous l'avons séparé en jeu d'entraînement, de validation et de test que nous avons utilisé pour charger les images au niveau de chaque jeu en effectuant le prétraitement (nécessaire redimensionnement et normalisation) avant de générer les masques qui seront utilisés pour entraîner le modèle à l'aide du modèle entraîné précédemment.



	4	5822	Melanocytic nevi	5822
	5	1067	Dermatofibroma	5350
	2	1011	Melanoma	5335
	1	479	Vascular lesions	5160
	0	297	Benign keratosis-like lesions	5055
	6	129	Basal cell carcinoma	4790
	3	107	Actinic keratoses	4455

unduplicated 5514  
 duplicated 4501

Dimensions du Training Set : (32370, 224, 224, 3)  
 Dimensions du Test Set : (3597, 224, 224, 3)  
 Dimensions du Validation Set : (1103, 224, 224, 3)

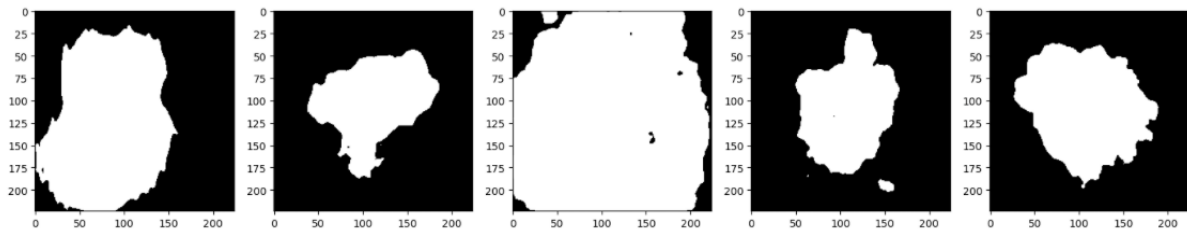


Fig 16 : Prétraitement effectué

Nous créons ensuite le modèle en utilisant une architecture CNN avec Random Search de keras-tuner pour trouver les meilleurs paramètres. Nous obtenons les résultats suivants :

Trial 5 Complete [00h 04m 29s]  
 val\_accuracy: 0.8005439639091492

Best val\_accuracy So Far: 0.8014506101608276 113/113 [=====] -  
 Total elapsed time: 00h 24m 45s [0.06516091525554657, 0.9808173775672913]  
 INFO:tensorflow:Oracle triggered exit

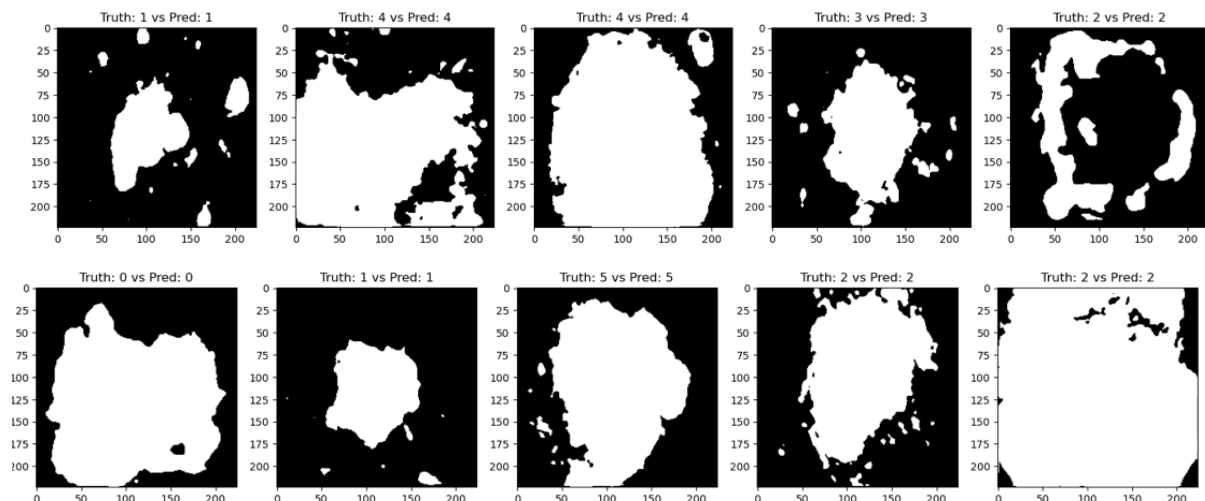


Fig 17 : Entraînement et Résultats du modèle de classification

Pour finir nous enregistrons le modèle.

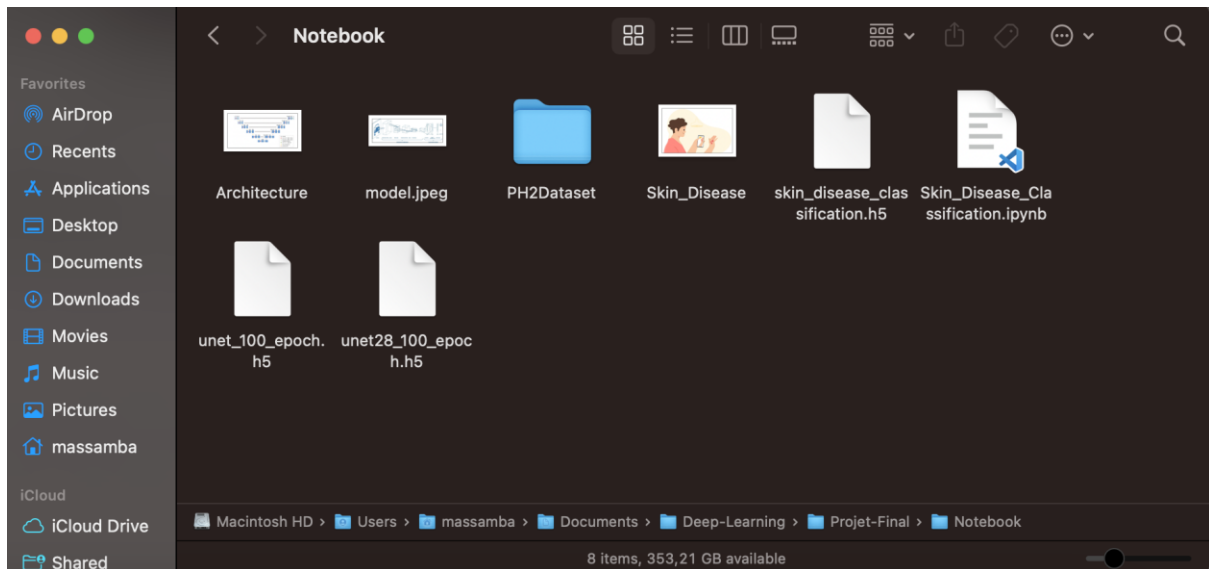


Fig 18 : Modèle de classification sauvegardé

## Regroupement

Comme nous l'avons vu précédemment, le modèle de segmentation crée les masques à partir de l'image RGB du patient et ce sont ces masques qui sont ensuite utilisés par le modèle de classification pour effectuer la prédiction.

Nous avons donc regroupé les deux modèles à l'aide d'une nouvelle fonction de prédiction et voici les résultats de quelques prédictions faites :

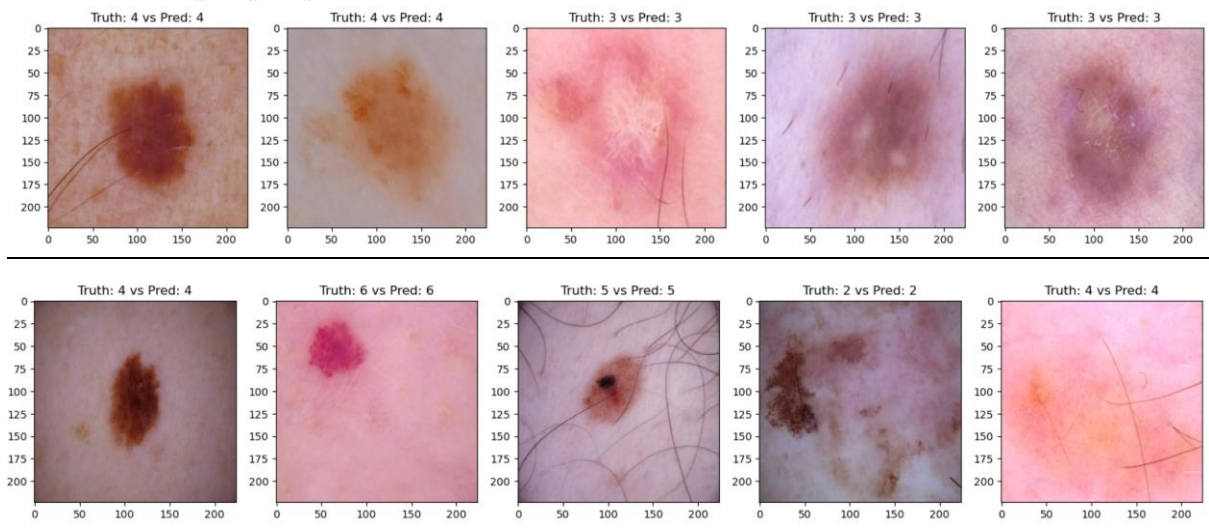


Fig 19 : Prédictions faites en utilisant l'architecture proposé

Le notebook a été envoyé pour voir les étapes de l'implémentation en détail.

## C. Analyse des résultats

Après entraînement et évaluation des performances du modèle nous obtenons les résultats suivants :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	403
1	0.99	1.00	0.99	486
2	0.97	1.00	0.98	505
3	1.00	1.00	1.00	525
4	1.00	0.89	0.94	615
5	0.93	1.00	0.96	550
6	1.00	1.00	1.00	513
accuracy			0.98	3597
macro avg	0.98	0.98	0.98	3597
weighted avg	0.98	0.98	0.98	3597

Tableau 3 : Performances de l'architecture proposée

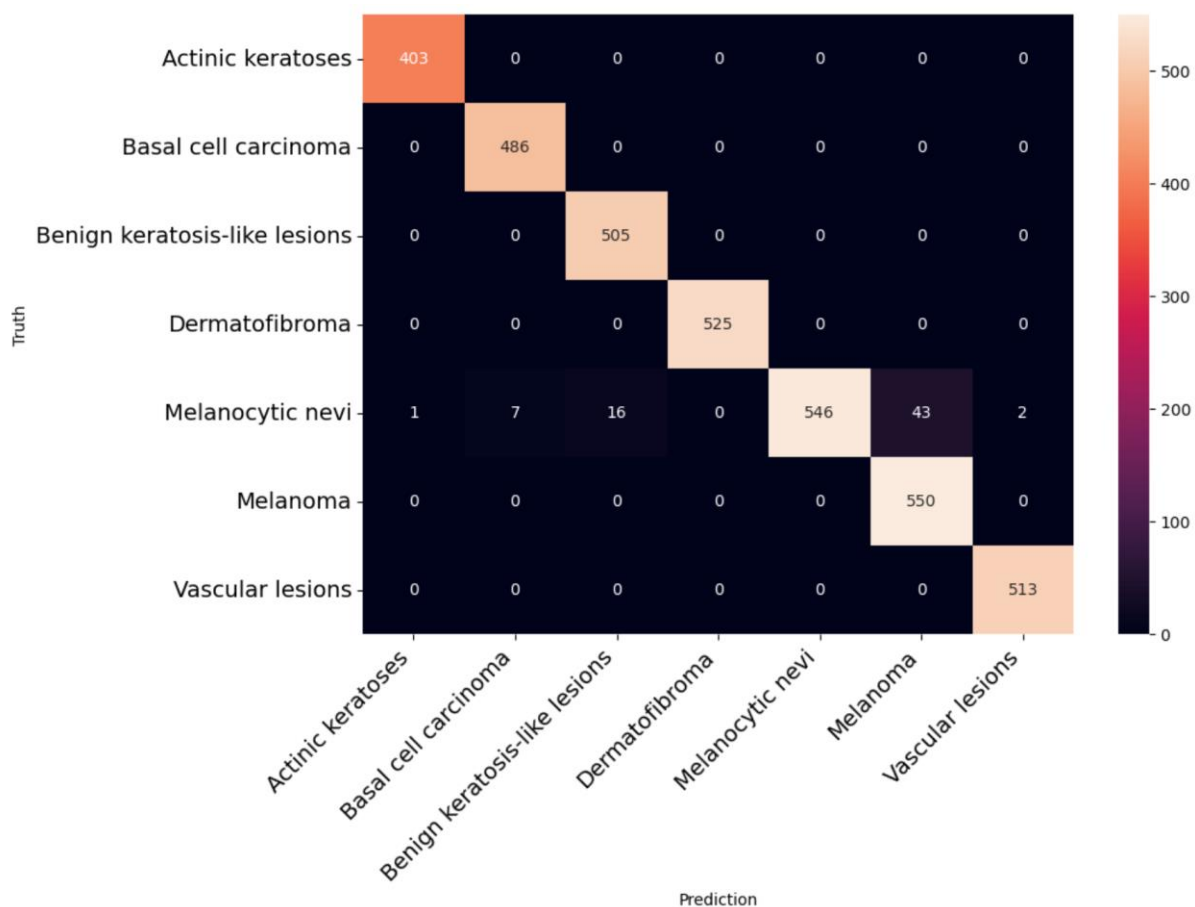


Fig 20 : Matrice de confusion de l'architecture proposée

Dans le tableau ci-dessous nous pouvons voir une comparaison des performances du modèle proposé par rapport au modèle défini dans l'article

Métriques	Modèle proposée (U-Net + CNN)	Modèle de l'article (Segmentation GWO + Feature extraction + AutoEncoder)
Accuracy	0.98	0.942
Precision	0.984	0.9615
Recall	0.984	0.918
F1-Score	0.9814	0.9339

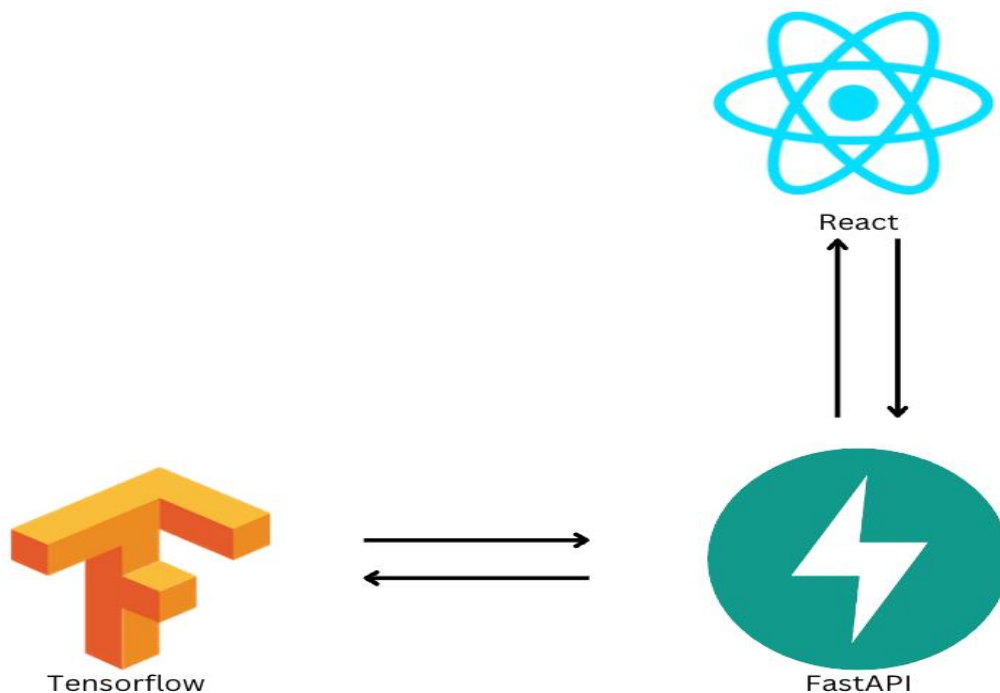
Tableau 4 : Analyse comparative de l'architecture proposée et celle de l'article

Nous voyons donc que le modèle proposé présente de meilleurs résultats que ceux de l'article. Cependant il faut noter que le modèle proposé dans l'article effectue une classification binaire tandis que celui que nous avons proposé effectue une classification multi classe.

## D. Application Pratique

Comme application pratique nous avons comme objectif de créer une application permettant de charger une image d'une lésion cutanée et qui affiche à l'utilisateur l'image segmentée montrant la région de la lésion ainsi que la maladie identifiée avec un niveau de certitude.

Pour cela nous adoptons l'architecture suivante :



L'application React est utilisée comme le frontend de notre application où nous avons deux pages :

- Une page d'accueil avec un message de bienvenue ainsi que des informations sur les différentes maladies identifiables avec notre modèle

Les maladies de la peau, également appelées troubles dermatologiques ou dermatoses, englobent un large éventail d'affections qui affectent la peau, le plus grand organe du corps. Ces troubles peuvent résulter de diverses causes, notamment des facteurs génétiques, des infections, des allergies, des réactions auto-immunes, des facteurs environnementaux, etc. Avec notre application, vous pourrez charger votre image et obtenir une tentative de diagnostic pour vous orienter.

**A noter qu'il faudra vous rapprocher d'un médecin par la suite, ce Diagnostic n'est pas celui d'un professionnel!**

**Actinic keratoses**

La kératose actinique touche généralement les personnes âgées. Diminuer l'exposition au soleil peut aider à réduire le risque. La maladie touche le plus souvent le visage, les lèvres, les oreilles, le dos des mains, les avant-bras, le cuir chevelu et le cou. La plaque squameuse rugueuse croît lentement et ne provoque généralement pas d'autres signes ou symptômes. Le développement d'une lésion peut prendre des années. En raison du risque de cancer, une ablation préventive est généralement réalisée.

**Basal cell carcinoma**

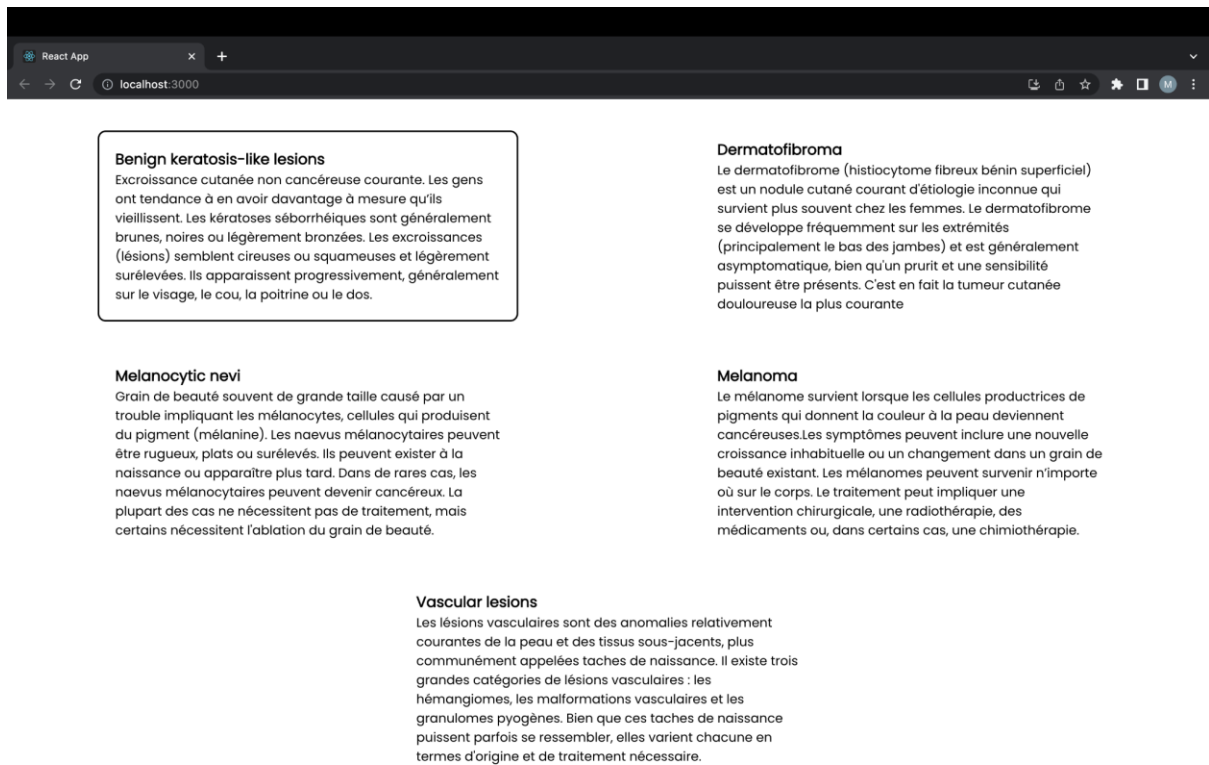
Les cellules basales produisent de nouvelles cellules cutanées à mesure que les anciennes meurent. Limiter l'exposition au soleil peut aider à empêcher ces cellules de devenir cancéreuses. Ce cancer apparaît généralement sous la forme d'une masse blanche et cireuse ou d'une tache brune et squameuse sur les zones exposées au soleil, comme le visage et le cou. Les traitements comprennent des crèmes sur ordonnance ou une intervention chirurgicale pour éliminer le cancer.

**Benign keratosis-like lesions**

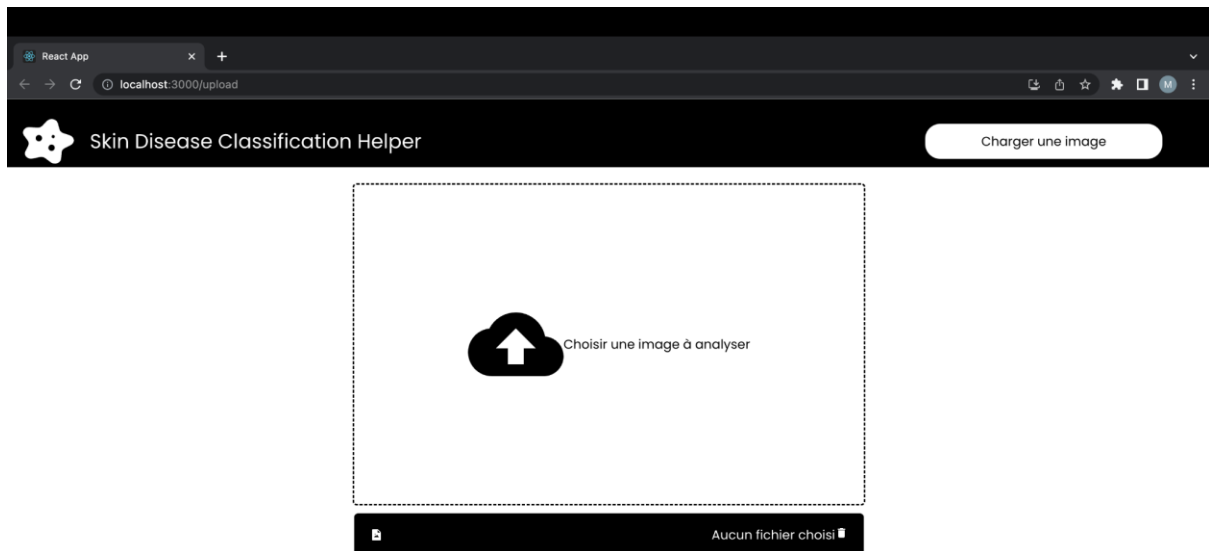
Excroissance cutanée non cancéreuse courante. Les gens ont tendance à en avoir davantage à mesure qu'ils vieillissent. Les kératoses séborrhéiques sont généralement brunes, noires ou légèrement bronzées. Les excroissances

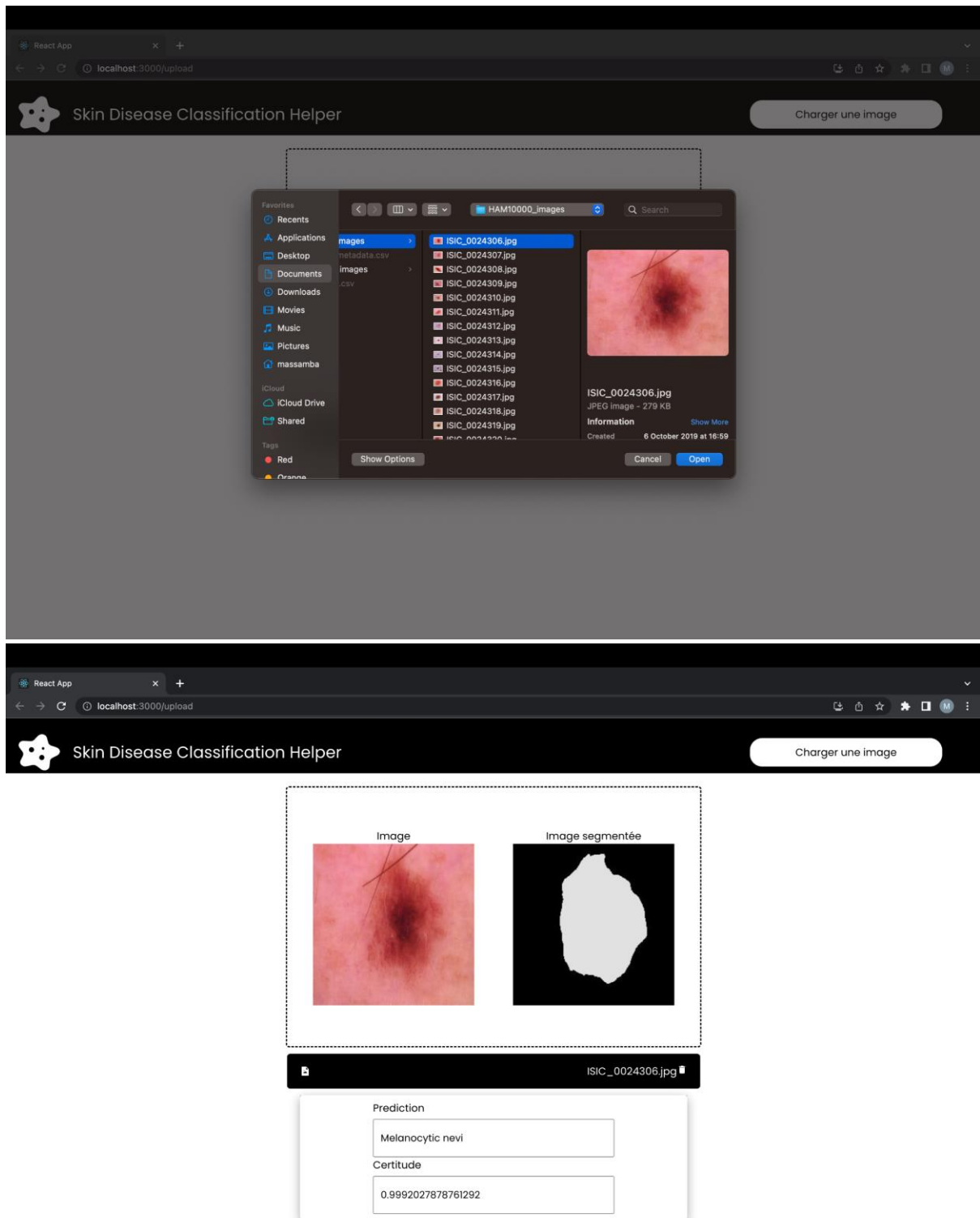
**Dermatofibroma**

Le dermatofibrome (histiocytome fibreux bénin superficiel) est un nodule cutané courant d'étiologie inconnue qui survient plus souvent chez les femmes. Le dermatofibrome se développe fréquemment sur les extrémités



- Une page où l'utilisateur peut déposer son image et recevoir les résultats de la segmentation et de la classification





L'api que nous avons développée avec FastAPI utilise les modèles que nous avons créé et enregistré au format HDF5 pour effectuer les prédictions qu'il renvoie comme réponse aux requêtes effectuées au niveau de l'endpoint `http://localhost:5000/predict`

The image shows a Visual Studio Code editor window with a Python FastAPI application. The Explorer sidebar on the left shows a project structure with folders like `API`, `data`, `models`, `utils`, `classification`, and `segmentation`. The main editor displays `app.py` with the following code:

```
1 from fastapi import FastAPI, HTTPException, Request
2 from fastapi.middleware.cors import CORSMiddleware
3 from utils.classification.cl_utils import PREDS_DICT, preprocess_image, load_classification_model, enhance
4 from utils.segmentation.seg_utils import *
5 import uvicorn
6 from PIL import Image
7 import io
8 import numpy as np
9
10 app = FastAPI()
11
12 app.add_middleware(
13     CORSMiddleware,
14     allow_origins=['*'],
15     allow_credentials=True,
16     allow_methods=['*'],
17     allow_headers=['*']
18 )
19
20 cl_model = load_classification_model()
21 seg_model = load_segmentation_model()
22
23 def predict(image):
24     mask = seg_model.predict(image.reshape(1,224,224,3), verbose=0)
25     mask = enhance(mask)
26     mask = np.concatenate((mask, mask, mask), axis=2)
```

The terminal at the bottom shows the command to run the application and its output:

```
conda activate tf_env
/Users/massamba/anaconda3/envs/tf_env/bin/python /Users/massamba/Documents/Deep-Learning/Projet-Final/App/API/app.py
(base) massambamohameds-mbp API % conda activate tf_env
(tf_env) massambamohameds-mbp API % /Users/massamba/anaconda3/envs/tf_env/bin/python /Users/massamba/Documents/Deep-Learning/Projet-Final/App/API/app.py
Metal device set to: Apple M2 Pro

systemMemory: 16.00 GB
maxCacheSize: 5.33 GB

WARNING:tensorflow:Error in loading the saved optimizer state. As a result, your model is starting with a freshly initialized optimizer.
INFO: Started server process [11579]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://127.0.0.1:5000 (Press CTRL+C to quit)
2023-09-08 20:22:47.537879: W tensorflow/tsl/platform/profile_utils/cpu_utils.cc:128] Failed to get CPU frequency: 0 Hz
INFO: 127.0.0.1:55158 - "POST /predict HTTP/1.1" 200 OK
```



# Webographie

1. DigitalSreeni. (2021, May 27). *219 - Understanding U-Net architecture and building it from scratch* [Video]. YouTube. <https://www.youtube.com/watch?v=GAYJ81M58y8>
2. O'Sullivan, C. (2023, March 9). U-Net Explained: Understanding its Image Segmentation Architecture. *Medium*. <https://towardsdatascience.com/u-net-explained-understanding-its-image-segmentation-architecture-56e4842e313a>
3. *Qu'est-ce qu'un réseau de neurones convolutifs ?* | IBM. (n.d.). <https://www.ibm.com/fr-fr/topics/convolutional-neural-networks>
4. Vancappel, K., & Vancappel, K. (2023). Deep Learning : le Réseau neuronal convolutif (CNN). *Business & Decision*. <https://fr.blog.businessdecision.com/tutoriel-deep-learning-le-reseau-neuronal-convolutif-cnn/>
5. Defend Intelligence. (2021, February 7). *Coder un réseau de neurones convolutifs de classification d'image avec Python et Tensorflow*. [Video]. YouTube. <https://www.youtube.com/watch?v=6FHtTyZxS5s>
6. Riva, M., & Riva, M. (2023). Batch normalization in convolutional neural networks | Baeldung on Computer Science. *Baeldung on Computer Science*. <https://www.baeldung.com/cs/batch-normalization-cnn>
7. Thales Sehn Körting. (2014, July 31). *How region growing image segmentation works* [Video]. YouTube. <https://www.youtube.com/watch?v=VaR21S8ewCQ>