

# SOMMAIRE

<i>REMERCIEMENTS</i> .....	I
<i>DEDICACE</i> .....	II
<i>DEDICACE</i> .....	III
<b>SOMMAIRE</b> .....	<b>IV</b>
<b>LISTE DES FIGURES</b> .....	<b>VI</b>
<b>INTRODUCTION GENERALE</b> .....	<b>1</b>
<b>CHAPITRE I : LES LOGICIELS DE TRAITEMENT D'IMAGES MEDICALES EXISTANT</b> .....	<b>3</b>
<b>I.1. INTRODUCTION</b> .....	<b>4</b>
<b>I.2. LOGICIELS SPECIALISES</b> .....	<b>4</b>
<i>I.2.1. Visilog</i> .....	4
<i>I.2.2. Aphélon</i> .....	5
<i>I.2.3. ImageJ</i> .....	6
<i>I.2.4. OsiriX</i> .....	7
<i>I.2.5. MediPy</i> .....	8
<b>I.3. LOGICIELS GENERALISTES</b> .....	<b>8</b>
<i>I.3.1. Matlab</i> .....	9
<i>I.3.2. Scilab</i> .....	9
<i>I.3.3. Labview</i> .....	10
<i>I.3.4. Python</i> .....	11
<b>I.4. BIBLIOTHEQUES DE PROGRAMMATION</b> .....	<b>11</b>
<i>I.4.1. Insight ToolKit (ITK)</i> .....	11
<i>I.4.2. Open CV</i> .....	12
<i>I.4.3. DIPLib</i> .....	14
<i>I.4.4. Aforge.NET</i> .....	14
<b>I.5. CONCLUSION</b> .....	<b>15</b>
<b>CHAPITRE II : SEGMENTATION &amp; ANALYSE DES TUMEURS DE LA PEAU</b> .....	<b>17</b>
<b>II.1. INTRODUCTION</b> .....	<b>18</b>
<b>II.2. MELANOME CUTANE</b> .....	<b>18</b>
<b>II.3. TYPES DU MELANOME</b> .....	<b>19</b>
1. <i>Le mélanome à extension superficielle (SSM)</i> .....	19
2. <i>Le mélanome nodulaire</i> .....	20
3. <i>Le lentigo malin</i> .....	20
4. <i>Le mélanome acrolentigineux ou mélanome acral</i> .....	21
5. <i>Tumeurs mélanocytaires (nævus)</i> .....	21
<b>II.4. LES ASPECTS CLINIQUES</b> .....	<b>22</b>
<b>II.5. EVOLUTION ET PRONOSTIC</b> .....	<b>23</b>
<b>II.6. ROLE DU TRAITEMENT D'IMAGE DANS LA MEDECINE</b> .....	<b>24</b>

1. Prétraitements.....	25
2. Segmentation .....	27
3. Détection de contour : .....	30
4. Opérateurs morphologiques.....	31
<b>II.7. EXTRACTION DES PARAMETRES.....</b>	<b>34</b>
1. La règle ABCD .....	34
2. La couleur .....	37
<b>II.8. CONCLUSION .....</b>	<b>38</b>
<b>CHAPITRE III : RESULTATS ET INTERPRETATIONS .....</b>	<b>39</b>
<b>III.1. INTRODUCTION .....</b>	<b>40</b>
<b>III.2. C SHARP.NET .....</b>	<b>40</b>
<i>Pourquoi l'orienté objet ?</i> .....	41
<i>Applications windows</i> .....	41
<i>Bibliothèque Aforge.NET</i> .....	42
<b>III.3. INTERFACE GRAPHIQUE.....</b>	<b>42</b>
1. Les variables .....	43
2. Les différentes composantes de l'interface.....	45
3. Gestion des erreurs.....	56
4. Les bulles d'info .....	57
<b>III.4. RESULTATS ET SIMULATION .....</b>	<b>57</b>
<b>III.5. EXECUTABLE, INSTALLABLE .....</b>	<b>59</b>
<b>III.6. CONCLUSION .....</b>	<b>65</b>
<b>CONCLUSION GENERALE .....</b>	<b>66</b>
<b>BIBLIOGRAPHIE.....</b>	<b>67</b>

# LISTE DES FIGURES

FIGURE I-1 INTERFACE DE VISUALISATION D'APHELION.....	6
FIGURE I-2. INTERFACE DE VISUALISATION D'IMAGEJ.....	7
FIGURE I-3. INTERFACE DE VISUALISATION D'OSIRIX.....	8
FIGURE I-4. INTERFACE DE VISUALISATION DE MATLAB.....	9
FIGURE I-5. ESPACE DE TRAVAIL DE SCILAB.....	10
FIGURE I-6. ESPACE DE TRAVAIL DE NI VISION ASSISTANT.....	11
FIGURE I-7. VUE D'ENSEMBLE D'OPENCV.....	13
FIGURE I-8. FENETRE DE VISUALISATION DE DIPLIB.....	14
FIGURE I-9. INTERFACE TRAITEMENT D'IMAGE DEVELOPPEE PAR AFORGE.IMAGING.....	15
FIGURE II-1. MELANOME SUPERFICIEL EXTENSIF (SSM) AVEC UN NODULE INVASIF.....	20
FIGURE II-2. MELANOME NODULAIRE.....	20
FIGURE II-3. MELANOME DE DUBREUILH DE L'ARETE NASALE.....	21
FIGURE II-4. MELANOME D'ACRAL LENTIGINEUX.....	21
FIGURE II-5. RESULTAT DE LA SUPPRESSION DES POILS. (A) IMAGE ORIGINALE, (B) IMAGE APRES SUPPRESSION DES POILS.....	26
FIGURE II-6. RESULTAT DE LA SEGMENTATION UTILISANT LA METHODE D'OTSU.....	31
FIGURE II-7. ÉROSION MORPHOLOGIQUE. (A) MASQUE ORIGINAL ; (B) EROSION, 3ITERATION ; (C) EROSION 6 ITERATIONS.....	32
FIGURE II-8. DILATATION MORPHOLOGIQUE. (A) MASQUE ORIGINAL ; (B) DILATATION 3ITERATION ; (C) DILATATION 6 ITERATIONS..	33
FIGURE II-9. FERMETURE MORPHOLOGIQUE.....	33
FIGURE II-10. OUVERTURE MORPHOLOGIQUE.....	34
FIGURE II-11. DETECTION D'AXES DE SYMETRIES.....	35
FIGURE II-12. EXEMPLES D'APPLICATION, A : MELANOME (COMPACITE = 2.70) ; B : TUMEUR BENIGNE (COMPACITE= 1.31).....	36
FIGURE III-1.FENETRE D'UN NOUVEAU PROJET C#.....	42
FIGURE III-2. INTERFACE REALISEE.....	44
FIGURE III-3. FENETRE DE CONTRASTE « CONTRASTFORM ».....	48
FIGURE III-4. FENETRE DE SEGMENTATION « SEGMENTATIONFORM ».....	49
FIGURE III-5. FENETRE DE LA RESOLUTION D'IMAGE.....	51
FIGURE III-6. LA FENETRE « INFORMATION DU PATIENT ».....	56
FIGURE III-7. CREER LE FICHER EXECUTABLE, ETAPE 1.....	59
FIGURE III-8. CREER LE FICHER EXECUTABLE, ETAPE 2 (CHOISIR L'EMPLACEMENT).....	60
FIGURE III-9. CREER LE FICHER EXECUTABLE, ETAPE 3 (TYPE DE FICHER).....	60
FIGURE III-10. CREER LE FICHER EXECUTABLE, ETAPE 4 (OPTION MISE A JOUR).....	61
FIGURE III-11. CREER LE FICHER EXECUTABLE, ETAPE 5 (FIN).....	62
FIGURE III-12. LE REPERTOIRE CONTENANT L'APPLICATION FINALE.....	63
FIGURE III-13. INSTALLATION DE L'APPLICATION      FIGURE III-14. APPARITION DE L'APPLICATION DANS LE MENU DEMARRER.....	63
FIGURE III-15 : INTERFACE GRAPHIQUE FINALE.....	64
FIGURE III-16. LE FICHER « PARAMETERS.TXT » CONTENANT LES PARAMETRES CALCULES.....	65

## INTRODUCTION GENERALE

Le mélanome est un cancer cutané qui se voit et que l'on repère d'après sa morphologie et son aspect extérieur. Pour favoriser l'identification clinique des mélanomes, les dermatologues ont développé une règle d'évaluation des tumeurs devant faire évoquer la possibilité d'un mélanome, la règle dite "ABCDE", "A" pour asymétrie, "B" pour Bordure, "C" pour couleur, "D" pour diamètre et "E" pour épaisseur ou évolution, selon les ouvrages.

Un mélanome est en général Asymétrique, il a des Bords irréguliers, encochés, polycycliques, une Couleur inhomogène, allant du brun clair au noir foncé, un Diamètre en général supérieur à 6 mm et il Evolue au fil du temps, changeant d'aspect, de taille, de couleur et d'épaisseur.

Seul le diagnostic au stade précoce permet de diminuer la mortalité liée à ce cancer. Grâce à un petit appareil manuel, il est possible de dépister beaucoup plus facilement le mélanome. Il est aussi plus facile de rassurer un patient au sujet d'une tache qui pouvait paraître suspecte à l'œil nu.

La dermoscopie est très utilisée en dermatologie, mais seulement si elle est pratiquée par des médecins formés spécifiquement en matière. D'autre coté, elle reste une technique manuelle, donc elle prend du temps et de l'effort. Avec la dispersion du mélanome, surtout dans la saison du soleil, il est devenu primordiale de trouver d'autre solution : un système informatisé qui allégit le travail pour le médecin, qui donne des résultats précises et qui permet l'archivage des données.

Autant qu'ingénieur on connaît plusieurs méthodes de traitement d'images, on peut les appliquer pour faciliter la tâche. Plusieurs langages de programmations peuvent être utilisés, et même quelque logiciels de traitements d'images plus généralisés. Les interfaces de traitements d'images qui existent offrent des traitements bien limités et elles ne couvrent pas les besoins du dépistage et elles sont incapable de faire les calculs nécessaires pour donner un diagnostic bien précis.

Il est très important de trouver une solution à ce propos, une interface qui est simple, efficace et bien précise pour l'utilisation en dermatologie, et qui est à la fois installable et

exécutable dans n'importe quel système sans exigence des programmes pré installé ; c'est exactement ce que nous visons dans ce projet. Pour cela nous avons utilisé le langage Visual c#.

C# (prononcé C-Sharp) est un langage de programmation orientée objet, développé par la société Microsoft, et notamment un de ses employés, Anders Hejlsberg, pour la plateforme .NET (point NET).

C# peut être utilisé pour créer des applications clientes Windows, des services Web XML, des composants distribués, des applications client-serveur et des applications de base de données.

Dans ce travail, nous avons utilisé les winforms (Windows forms) qui permet de créer des applications exécutables (avec extension .exe). Le but est de créer une interface graphique qui permet de charger une image (image dermatologique dans notre cas), d'appliquer un traitement composé d'un ensemble de filtres pour extraire uniquement la région d'intérêt (dans notre cas, la tumeur) et de calculer un ensemble des paramètres pertinents. Nous envisageons, pour la caractérisation, d'utiliser quatre critères différents (règle ABCD bien connue en dermatologie): l'Asymétrie de formes (A), les bords des structures pigmentées (B), la diversité des couleurs (C) et le diamètre des tumeurs (D). Enfin, l'image finale et les paramètres calculés sont enregistrés et archivés.

La mémoire s'articule autour de trois chapitres:

Le premier chapitre décrit quelques logiciels et bibliothèques dédiés pour le traitement des images médicales et qui peuvent être exploités en dermatologie, mais ces logiciels présentent toujours des limitations.

Le deuxième chapitre présente les méthodes de segmentation d'images, où nous avons décrit un certain nombre de méthodes classiques de segmentation (approche contour ou approche région) et nous décrivons la règle ABCD.

Le troisième chapitre est consacré à la présentation des résultats et l'interface que nous avons développée.

Nous terminons ce travail par une conclusion générale, en résumant notre contribution et en proposant les perspectives de nos travaux.

# **CHAPITRE I**

---

**LES LOGICIELS DE TRAITEMENT D'IMAGES MEDICALES  
EXISTANT**

---

## I.1. Introduction

Par le développement énorme des techniques de l'imagerie médicale, un traitement numérique est devenu nécessaire. Pour cet objet les chercheurs en utilisent une variété des logiciels de traitements, les uns sont bien spécialisés pour le domaine médical, les autres sont plus généralisés pour le traitement des images en général.

Cependant, les logiciels qui s'appliquent spécifiquement pour la dermatologie sont bien rares ou même introuvables. Les spécialistes sont toujours obligés de rester dans le manuel ou d'exploiter les logiciels existants pour répondre à leurs besoins, ce qui est pas toujours facile. D'où on a trouvé la nécessité de développer une telle application.

On essaye dans ce chapitre de voir quelques logiciels et bibliothèques dédiés pour le traitement des images médicales, et qui peuvent être exploités en dermatologie, mais qui présentent toujours des limitations.

## I.2. Logiciels spécialisés

Ils existent des logiciels spécialisés qui fournissent une variété des opérations nécessaires pour le traitement numérique d'images. Ils sont bien simples et lisibles pour des utilisateurs non experts, mais à la fois sophistiqués et efficaces pour les utilisateurs spécialistes.

### I.2.1. Visilog

Visilog [1] est l'environnement de référence pour la réalisation d'applications en traitement d'images. Il s'appuie sur une bibliothèque d'algorithmes de traitements d'images éprouvée et enrichie depuis plus de 20 ans et sur une longue expérience de réalisation d'interfaces de produits d'imagerie scientifique.

Il possède plusieurs modules à propriétés différentes : visualisation, traitement et analyse, et Développement de produits dédiés.

**Visualisation** : Il donne accès aux fonctions usuelles de zoom, scroll, choix de LUT, annotations alphanumériques ou graphiques.

**Traitement, analyse et automatisation** : (V7-Xpert) un logiciel d'imagerie professionnel dédié à l'analyse et au traitement d'images. Il met à la disposition de l'utilisateur la

richesse des algorithmes de sa bibliothèque par l'intermédiaire de commandes interactives. Ses scripts sont réalisés en Visual Basic.

**Développement de produits dédiés :** (V7-ToolKit) spécialement conçu à l'usage des développeurs qui veulent réaliser leur propre logiciel de traitement et d'analyse d'images. Il offre la possibilité d'enrichir la bibliothèque des fonctions de Visilog par des programmes de calcul spécifiques développés en C++.

Une procédure détaillée permet de réaliser l'exportation de ces fonctions dans l'environnement VBA (Visual Basic for Applications) afin de les rendre accessibles sous forme de commandes.

### **I.2.2. Aphélon**

La suite logicielle de traitement et d'analyse d'images Aphélon [2] met à la disposition de ses utilisateurs un ensemble extrêmement complet d'outils de traitement d'images 2D et 3D, incluant des centaines d'opérateurs d'imagerie, les algorithmes de morphologie mathématique les plus récents du Centre de Morphologie Mathématique ainsi que des outils de classification. Toutes ces fonctions de traitement et d'analyse sont disponibles depuis une interface utilisateur très conviviale et efficace, et sous la forme de composants .NET et de bibliothèques natives. Un choix de 4 interpréteurs de commandes (version Dev) aide l'utilisateur à développer des applications d'imagerie traitant une ou plusieurs images acquises par une caméra ou disponibles depuis le réseau

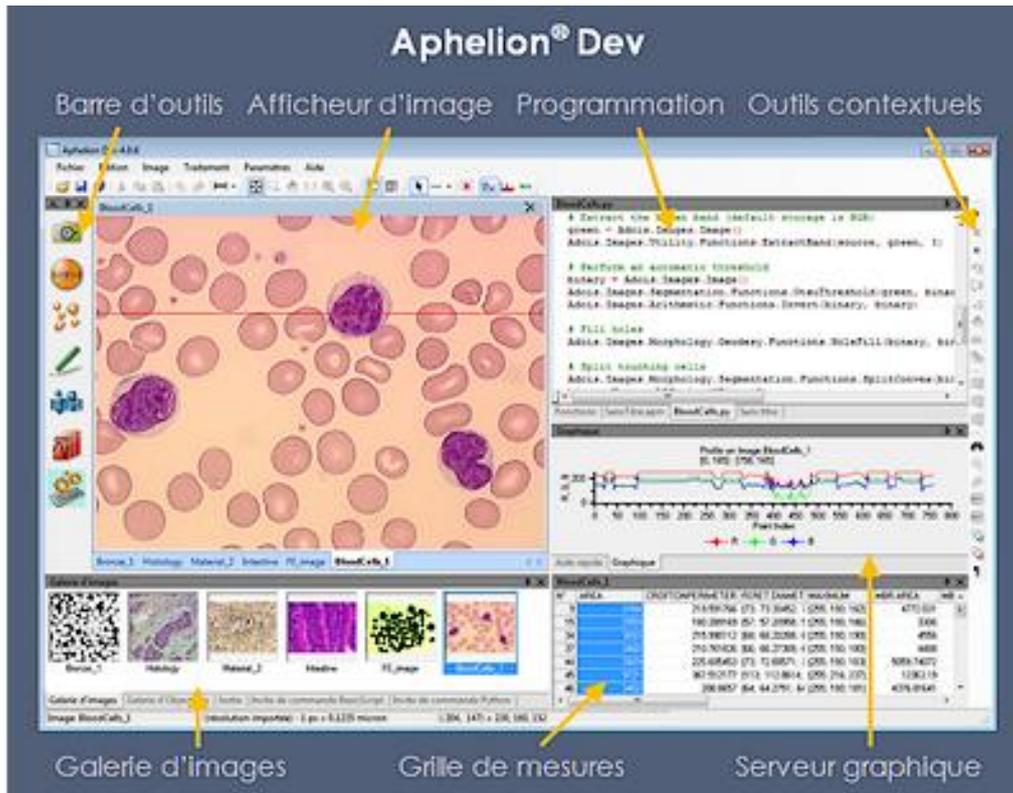


Figure I-1 interface de visualisation d'Aphelion

### I.2.3. ImageJ

ImageJ [3], [4] est un logiciel de traitement et d'analyse d'images adapté aux applications en biologie, développé par les Instituts Nationales de la Santé (NIH).

- un logiciel multiplateforme : il fonctionne en langage Java ce qui le rend accessible par les environnements Mac, Microsoft et Linux,
- logiciel open source : c'est donc un logiciel du domaine public «disponible gratuitement», non «soumis au copyright» ; il ne nécessite donc pas de licence ; le code source est en accès libre et peut être modifié.

Il sert à afficher, éditer, analyser, traiter, économiser et imprimer des images de 8 bits, de 16 bits et de 32 bits. Il reconnaît plusieurs formats d'image comprenant le TIFF, le GIF, le JPEG, le BMP, le DICOM, les FITS et " données brutes ". Notons par ailleurs qu'ImageJ est très utile en complément de MATLAB ou d'ITK pour visualiser les résultats des traitements appliqués ou alors pour réaliser des tests simples de filtrage et de morphologie mathématiques.

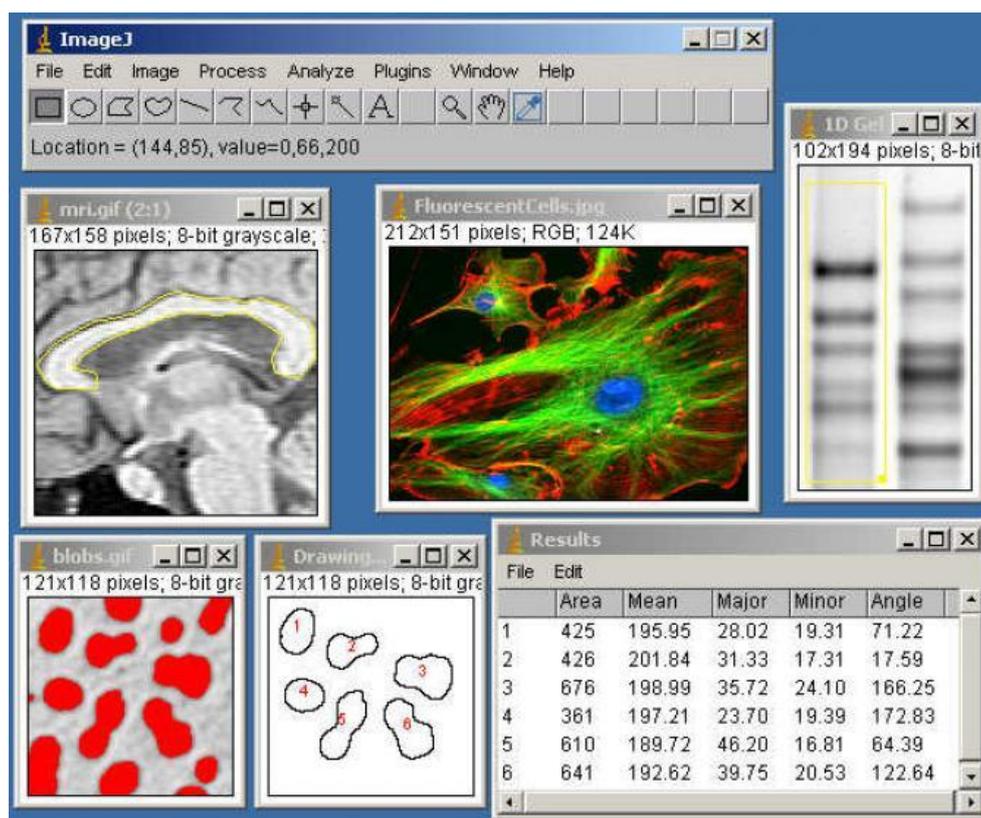


Figure I-2. Interface de visualisation d'ImageJ

### I.2.4. OsiriX

OsiriX [5] est un logiciel de traitement d'image dédié aux images DICOM (.dcm) produite par l'équipement d'imagerie (IRM, CT, PET, PET-CT, SPECT-CT, ultrasons, ...). Il est entièrement compatible avec la norme DICOM pour les formats d'images de communication et de fichiers d'images.

OsiriX a été conçu spécifiquement pour la navigation et la visualisation de la multimodalité et les images multidimensionnelles: 2D, 3D, 4D (séries 3D avec dimension temporelle) et 5D (séries 3D avec dimensions temporelle et fonctionnelle). La visionneuse 3D offre tous les modes modernes de rendu: reconstruction multiplanaire (MPR), le rendu de surface, rendu volumique et projection d'intensité maximale (MIP). Tous ces modes en charge les données 4D et sont capables de produire des images fusion entre deux séries différentes (PET-CT et SPECT-CT support d'affichage).

OsiriX est en même temps un poste de travail PACS DICOM pour l'imagerie et un logiciel de traitement d'image pour la recherche médicale (radiologie et imagerie nucléaire), l'imagerie fonctionnelle, imagerie 3D et imagerie moléculaire.

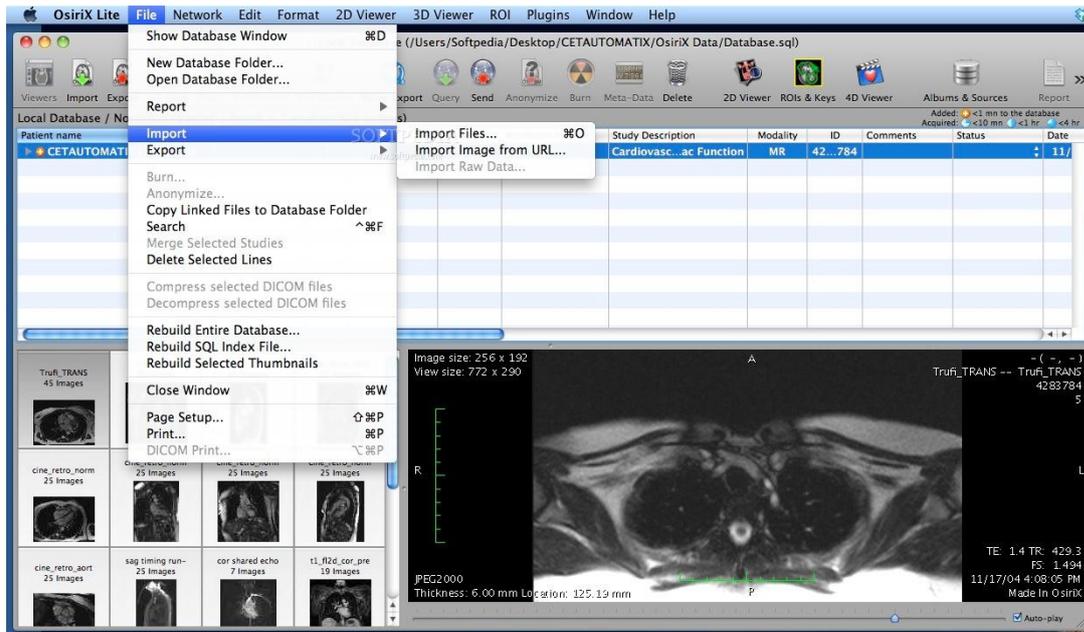


Figure I-3. Interface de visualisation d'OsiriX

### I.2.5. MediPy

MediPy [2] est une plateforme de visualisation et de traitement d'images médicales. Elle est destinée à la fois aux cliniciens et aux chercheurs en étant à la fois facile d'utilisation et ouverte à l'extension de ses fonctionnalités.

Les cliniciens pourront bénéficier des tâches préprogrammées (segmentation, recalage ou détection de lésions) et la possibilité de définir de nouvelles tâches, leur permettant d'adapter le logiciel à leurs besoins propres. De nombreux formats de fichiers peuvent être lus, et la récupération de données depuis un PACS est également possible.

Les chercheurs pourront rapidement développer de nouveaux outils en se basant sur des interfaces soit en Python pour une forte productivité, soit en C++ pour de hautes performances. Des ponts permettent un échange facile des données entre les deux langages. Des adaptateurs sont de plus prévus pour l'utilisation de bibliothèques tierces, au niveau traitement (par exemple SPM ou ITK) ou au niveau visualisation (par exemple VTK).

### I.3. Logiciels généralistes

Ces logiciels ont une vocation plus large que le traitement d'image, mais ils contiennent une boîte à outils spécialisés qui permet de faciliter la manipulation des algorithmes.

### I.3.1. Matlab

Matlab est à la base dédié au calcul matriciel. Mais en présentant l'image comme une matrice de couleur, il est facile de lui appliquer un grand nombre de traitement.

**Image Processing Toolbox** est la boîte à outil de MATLAB dédiée au traitement d'images ; Elle contient les algorithmes classiques de traitement d'images, de segmentation et elle est facilement utilisée via le GUI (Graphical User Interface) de MATLAB. [6]

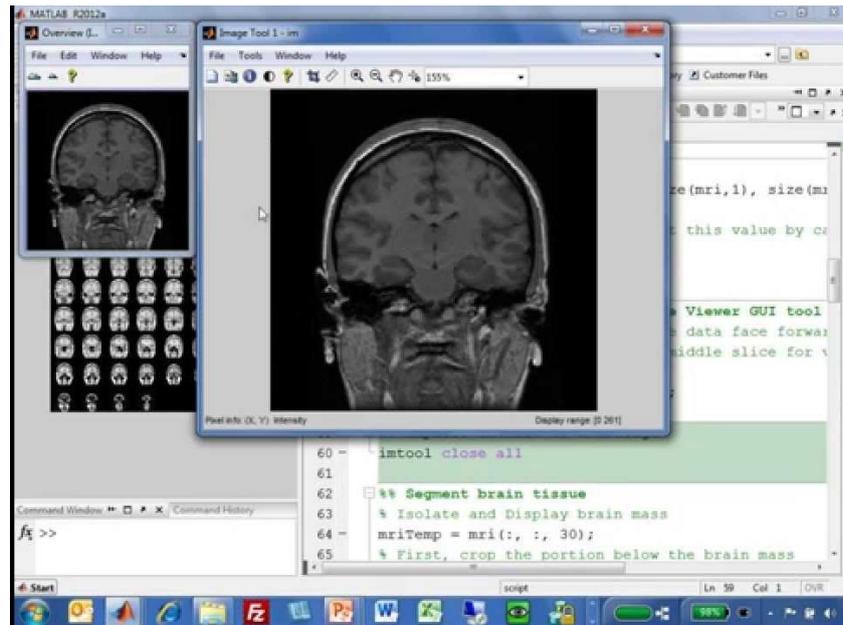


Figure I-4. interface de visualisation de Matlab

### I.3.2. Scilab

Scilab [7] est un logiciel scientifique pour des calculs numériques dans un environnement convivial. C'est un logiciel libre INRIA et CERMICS (pour L'Institut National de Recherche en Informatique et en Automatique et Centre d'Enseignement et de Recherche en Mathématiques et Calcul Scientifique). On peut dire que c'est une alternative gratuite de Matlab, à l'origine développée par l'INRIA. La syntaxe est assez semblable,

« Scilab Image Processing Toolbox » [8] SIP est la boîte à outils Scilab elle effectue des tâches d'imagerie telles que le filtrage, le flou, la détection de contour, seuillage, la manipulation d'histogramme, segmentation, morphologie mathématique, traitement des images couleurs, etc.

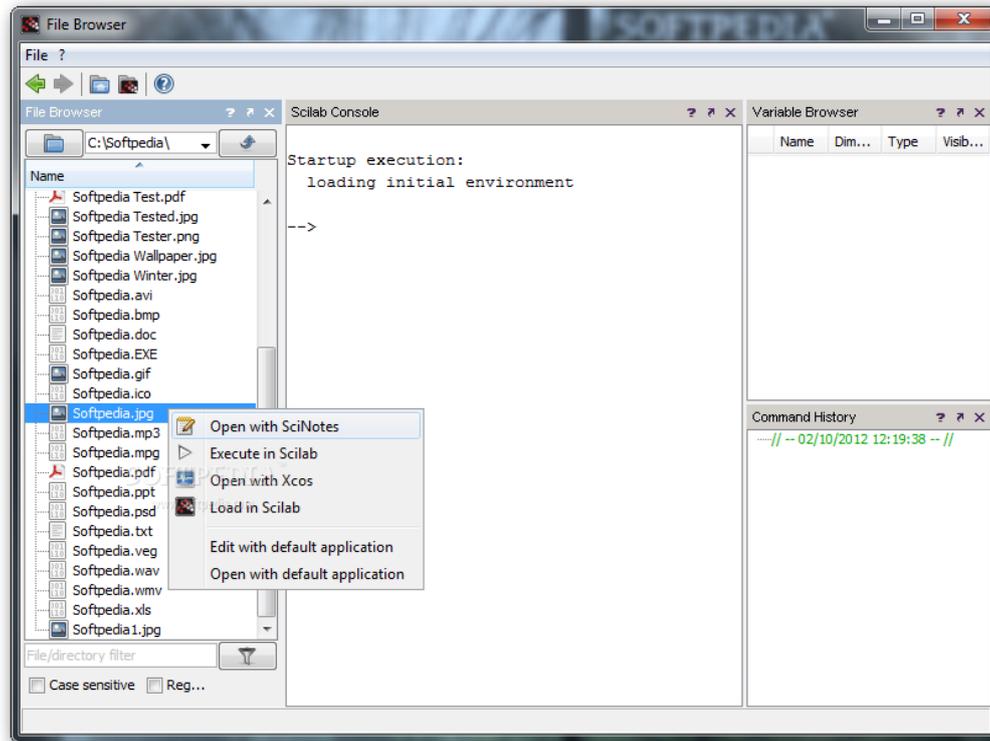


Figure I-5. Espace de travail de Scilab

### I.3.3. Labview

Labview [9] (Laboratory Virtual Instrument Engineering WorkBench) est un logiciel de développement de programmes d'application. Labview utilise un langage de programmation essentiellement graphique dédié au contrôle, à l'acquisition, l'analyse et la présentation de données. En Labview, on n'écrit pas de lignes de programme dans un langage textuel comme les langages de programmation habituelles, on manipule des objets graphiques. Ces objets graphiques représentent à la fois les variables du programme, ainsi que des fonctions qui vont réaliser des actions sur ces variables. La programmation en Labview consiste simplement à concevoir le traitement de l'information, organiser et relier les variables avec les fonctions au moyen de fils.

Le module Vision Development permet de développer des applications de vision industrielle et de traitement d'images en utilisant l'environnement de programmation graphique Labview pour Windows et les systèmes d'exploitation temps réel, ainsi que le C, le C++, ou le C# pour les systèmes Windows. Ils existent des centaines d'algorithmes de traitement d'images pour choisir à inclure entre les blocs, des filtres, des opérateurs morphologiques, ....

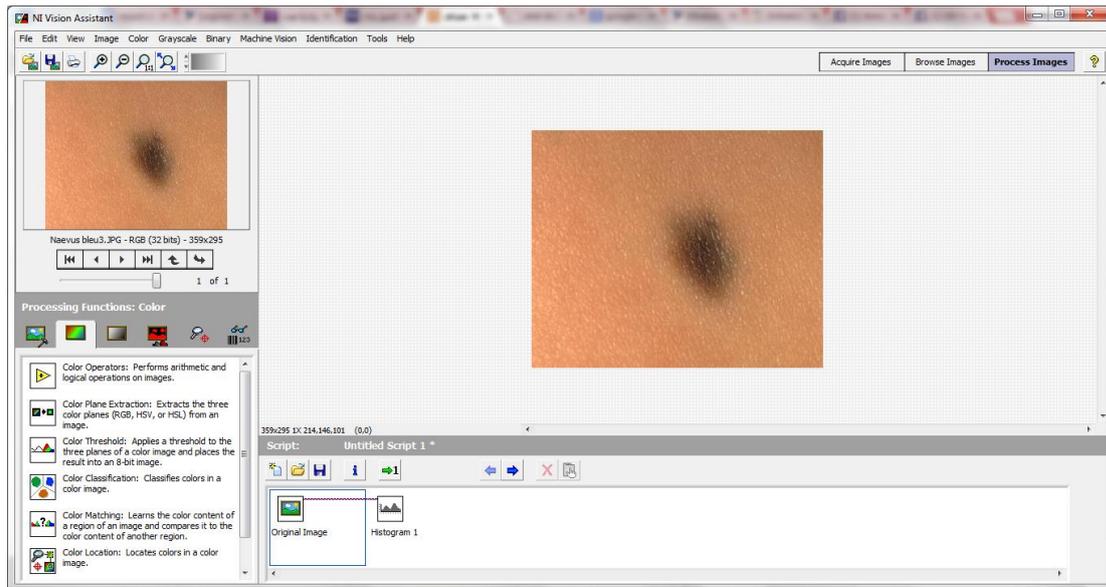


Figure I-6. Espace de travail de NI Vision Assistant

### I.3.4. Python

À la base Python [10] est plutôt un langage de programmation, mais son syntaxe et son mode d'utilisation se rapprochent de Matlab. Il existe aussi une boîte à outils image, nommée PIL (pour Python Imaging Library) [11], qui convient pour les traitements de bas niveau mais pas trop pour les traitements plus évolués (pas de morphologie mathématique apparemment). On peut aussi utiliser le langage pour interfacier ITK (Insight Segmentation and Registration Toolkit).

## I.4. Bibliothèques de programmation

Pour les langages de programmation plus classiques comme C/C++/C# ou Java, on trouve une variété des bibliothèques spécialisés. Malheureusement ils sont fréquemment incomplètes ou en développement. On cite les plus connus et utilisés.

### I.4.1. Insight ToolKit (ITK)

Insight Toolkit, ou ITK [12] est un projet américain destiné à offrir une base solide aux développements informatiques dans le cadre du traitement d'images. Les applications envisagées sont plutôt orientées vers l'imagerie médicale, mais le logiciel est suffisamment généraliste pour être utilisé dans d'autres domaines.

ITK utilise beaucoup le mécanisme de Template propre au C++. De ce fait, un algorithme une fois écrit pourra être compilé pour des images de type et de dimension arbitraire, ce

qui améliore considérablement la généralité de la bibliothèque. Par contre, le développement et la manipulation de la bibliothèque sont un peu difficile. Une alternative à la programmation en C++ avec ITK est de passer par un « wrapper » (emballage), pour appeler les fonctions d'ITK depuis un autre langage. Plusieurs langages sont supportés : Python, Java, Matlab [13]

#### **I.4.2. Open CV**

OpenCV (Open Source Computer Vision) [14] est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API (pour Application Programming Interface) pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plateformes Windows, GNU/Linux, Android et MacOS.

Initialement écrite en C il y a 10 ans par des chercheurs de la société Intel, OpenCV est aujourd'hui développée, maintenue, documentée et utilisée par une communauté de plus de 40.000 membres actifs. C'est la bibliothèque de référence pour la vision par ordinateur, aussi bien dans le monde de la recherche que celui de l'industrie.

Par ailleurs, cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

**Core** : les fonctionnalités de base.

Cette bibliothèque permet de manipuler les structures de base, réaliser des opérations sur des matrices, dessiner sur des images, sauvegarder et charger des données dans des fichiers XML.

**Imgproc** : traitement d'image.

Les fonctions et structures de ce module ont trait aux transformations d'images, au filtrage, à la détection de contours, et détection des régions d'intérêt.

**features2d** : descripteurs.

Ce module concerne principalement l'extraction de descripteurs nécessaires dans la caractérisation d'images.

**Objdetect** : détection d'objets.

Cette bibliothèque permet de faire de la reconnaissance d'objets dans une image au moyen de l'algorithme Adaboost [15].

**video** : traitement de flux vidéo.

Ces fonctions servent à segmenter et suivre les objets en mouvement dans une vidéo.

**Highgui** : entrées-sorties et interface utilisateur.

OpenCV intègre sa propre bibliothèque haut-niveau pour ouvrir, enregistrer et afficher des images et des flux vidéo. Celle-ci contient aussi un certain nombre de fonctions permettant de réaliser des interfaces graphiques très simples, mais largement suffisantes pour tester nos programmes.

**calib3d** : calibration, estimation de pose et stéréovision.

Ce module contient des fonctions permettant de reconstruire une scène en 3D à partir d'images acquises avec plusieurs caméras simultanément.

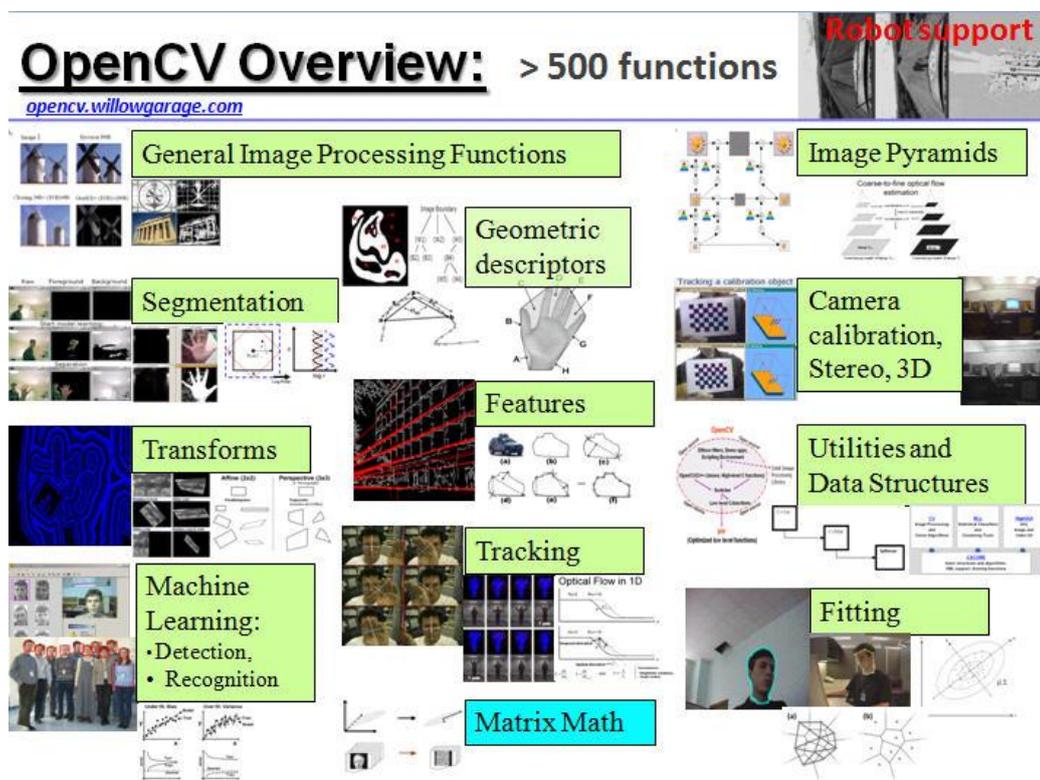


Figure I-7. Vue d'ensemble d'OpenCV.

### I.4.3. DIPLib

DIPLib [16] est une bibliothèque de traitement d'images écrite en C, dont le gros intérêt est de fournir une interface directe sous Matlab. Il contient un grand nombre de fonctions pour traiter et analyser des données d'image multidimensionnelles. La bibliothèque fournit aussi des fonctions pour les opérations de filtrage, la génération d'objet, et l'analyse statistique des images.

Cela permet de manipuler les images encore plus simplement qu'avec la boîte à outils image processing, et de bénéficier de quelques fonctionnalités supplémentaires.

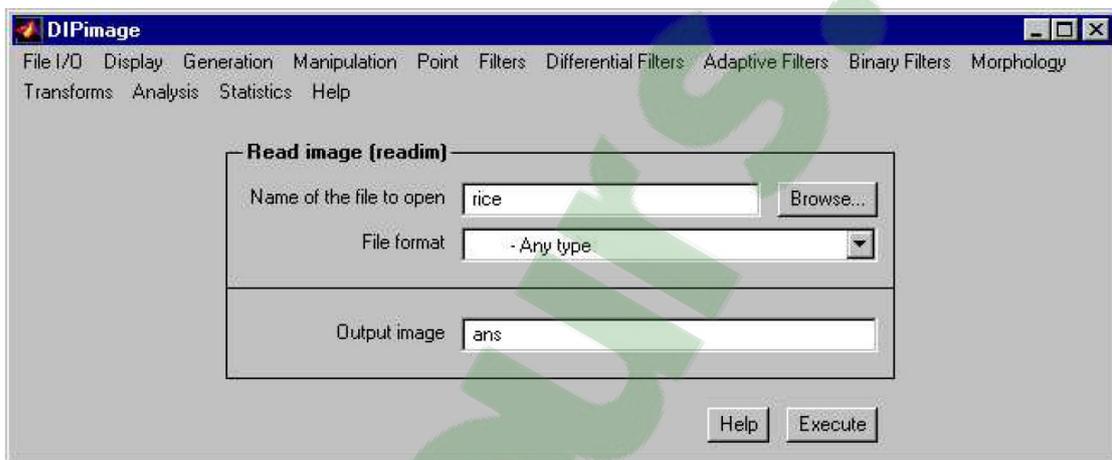


Figure I-8. Fenêtre de visualisation de DIPLib

### I.4.4. AForge.NET

AForge.NET [17] est une bibliothèque open source développée en C# conçu pour les développeurs et les chercheurs dans les domaines de la vision par ordinateur, Intelligence Artificielle, traitement de l'image, réseaux de neurones, algorithmes génétiques, la logique floue, l'apprentissage machine, robotique, etc.

Le cadre est constitué par l'ensemble des bibliothèques et des exemples d'applications qui démontrent leurs caractéristiques:

- AForge.Imaging : bibliothèque routines de traitement d'image et des filtres;
- AForge.Vision : Bibliothèque de vision par ordinateur;
- AForge.Video : ensemble de bibliothèques pour le traitement vidéo;
- AForge.Neuro : réseaux de neurones de la bibliothèque de calcul;
- AForge.Genetic : Bibliothèque de programmation de l'évolution;
- AForge.Fuzzy : bibliothèque de calculs floue;

- AForge.Robotics : bibliothèque soutien de certains kits robotiques;
- AForge.MachineLearning : bibliothèque d'apprentissage machine;

La plus part du travail que nous avons réalisé dans ce projet est basé sur la bibliothèque Aforge.Imaging, qui a facilité la programmation en offrant une collection d'interfaces et de classes, qui fournissent différents filtres de traitement d'image. Les Classes de cet espace permettent d'appliquer différente transformation sur l'image source.

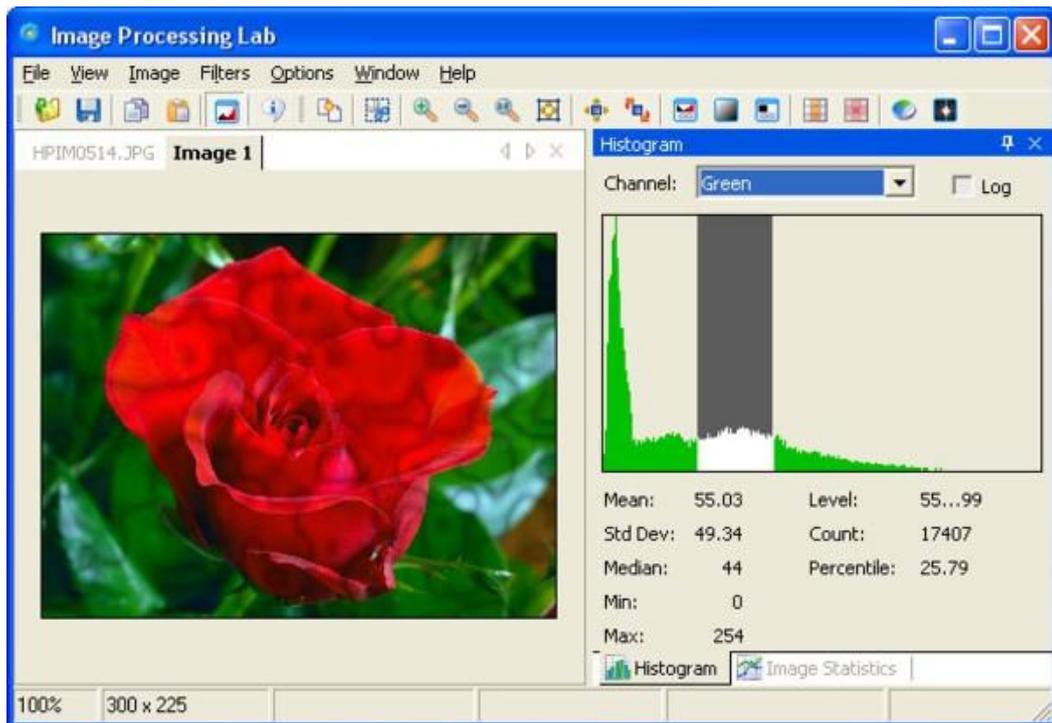


Figure I-9. Interface traitement d'image développée par Aforge.Imaging

## I.5. CONCLUSION

Toutes les solutions mentionnées dans ce chapitre peuvent s'adapter à notre problème qui est le dépistage du mélanome, mais aucune d'eux ne peut être considérée entièrement efficace.

Les logiciels spécialisés de traitement des images médicales sont bien équipés avec tout type de filtres nécessaires dans le domaine, mais ils ne possèdent pas les techniques de calcul nécessaire pour détecter les aspects cliniques pertinents qui doivent être appliqués en dermatologie. Egalement, ils ne peuvent pas être altérés pour s'adapter à l'exigence d'utilisateur.

En revanche, les logiciels généralisés peuvent bien s'adapter au besoin, il suffit de programmer des interfaces convenables, mais ce type d'interface exige la présence du logiciel utilisé en programmation pour qu'il fonctionne correctement.

Le système que nous avons proposé dans ce travail présente une solution plus efficace, qui est à la fois spécialisée pour les besoins des dermatologues et installable sur n'importe quel système.

# CHAPITRE II

---

SEGMENTATION & ANALYSE DES TUMEURS DE LA PEAU

---

## **II.1. Introduction**

La peau est l'organe le plus étendu et le plus lourd du corps humain. Ses fonctions sont multiples comme la défense immunitaire, la sécrétion, l'excrétion, la protection contre les agents physiques et chimiques et la protection contre les rayons ultraviolets ; cette dernière tâche est très importante, elle est assurée par la mélanine qui est un pigment de couleur variant du brun clair au noir. Le déficit en mélanine acquis ou congénital, est à l'origine de nombreuses maladies cutanées dont le mélanome malin. [18]

Le mélanome est une tumeur maligne des cellules de la peau qui fabriquent la mélanine. L'histoire naturelle et l'augmentation de fréquence du mélanome sont intimement liées à l'exposition solaire. [19]

La mortalité du mélanome est corrélée avec son épaisseur. Il est donc très important de dépister les mélanomes à un stade précoce : le mélanome, comme tout cancer, doit être pris en charge le plus tôt possible : détecté et traité tôt. Le mélanome peut être dépisté avec des surveillances régulières de la peau et des grains de beauté par le médecin. Pour faciliter cette tâche nous proposons un système qui fait un dépistage automatique, pour établir un diagnostic précoce. [20]

## **II.2. Mélanome cutané**

Le mélanome est un cancer de la peau ou des muqueuses, se développant à partir des mélanocytes (les cellules responsables de la pigmentation de la peau). Dans plus de 90% des cas, le mélanome est cutané, il peut aussi se développer plus rarement sur les muqueuses, sous les ongles, ou dans les yeux. Sur le plan cutané, le mélanome peut prendre des formes très différentes (tache marron, bouton...). Seul un médecin peut diagnostiquer un mélanome.

Le mélanome représente 3 % de l'ensemble des cancers humains, mais surtout, c'est le cancer humain dont la fréquence augmente le plus : son incidence (nombre de nouveaux cas par an) augmente de 10 % par an depuis 50 ans ! [20]

Le mélanome peut être dépisté lors de surveillances régulières de la peau et des grains de beauté. Mais il peut en effet passer pour un simple grain de beauté ou une tache brune et

c'est le médecin qui, suspectant un mélanome, décide de retirer la lésion et de la faire analyser par un laboratoire d'anatomopathologie. Si le laboratoire conclut à un mélanome, un traitement complémentaire est généralement proposé au patient, ainsi qu'une surveillance régulière et un dépistage du mélanome chez les autres membres de la famille (ascendants, descendants et fratrie).

La détection d'un mélanome doit être très précoce car le pronostic des lésions de stade I est favorable à traiter: le taux de survie relative à 5 ans est supérieur à 90 % lorsque la taille de la tumeur est inférieure à 1 mm (index de Breslow). [21]

Alors le problème qui se pose est la possibilité que le médecin suspecte un mélanome et demande les analyses nécessaires. Pour améliorer ces chances, il est conseillé d'utiliser un système informatique pour le dépistage, pour que le médecin analyse toute lésion, sans suspecté sa malignité. Toute tâche qui s'étend sur la peau peut donc faire l'objet d'une analyse médicale car on ne répètera jamais assez l'importance du dépistage précoce

### **II.3. Types du mélanome**

Il existe de nombreux types de mélanomes. Parmi ceux-ci on peut citer :

#### **1. Le mélanome à extension superficielle (SSM)**

Le SSM (Superficial Spreading Melanoma) représente environ 70% des mélanomes. Il est caractérisé par son extension horizontale durant plusieurs mois ou années, avant son extension verticale : le mélanome s'étend progressivement puisqu'il revêt tout d'abord un aspect de tache de contour irrégulier, de couleurs inhomogènes (noir, marron, rouge, blanc), non palpable au début de son évolution où il est intra-épidermique, puis devenant palpable lorsqu'il aborde sa croissance verticale : les cellules malignes sont alors susceptibles d'avoir franchi la jonction dermo-épidermique et de se rapprocher des vaisseaux sanguins et lymphatiques, et donc pouvant possiblement métastaser. [18]



Figure II-1. Mélanome superficiel extensif (SSM) avec un nodule invasif

## 2. Le mélanome nodulaire

Le mélanome nodulaire représente environ 15 % des mélanomes et son diagnostic précoce est difficile car il n'a pas de phase d'extension horizontale intra-épidermique. La lésion est souvent identifiée alors qu'elle a déjà une épaisseur notable. Il s'agit le plus souvent d'une lésion bleue noire entourée d'un halo inflammatoire (Figure II-2). Elle doit être distinguée des nævus bleus, qui sont des nævus pigmentaires bénins, dermiques, dont la couleur bleue noire et la bordure régulière sont le plus souvent caractéristiques. [19]



Figure II-2. Mélanome nodulaire

## 3. Le lentigo malin

C'est la forme la plus bénigne, elle représente 15- 20% des cas de mélanome, la lésion est noire. Il s'observe surtout chez les sujets âgés (entre 60-70 ans), le plus souvent au visage et plus rarement (10%) au dos des mains ou aux jambes. Cette lésion d'évolution clinique prolongée est considérée comme un précurseur de mélanome acral lentigineux. [18]



Figure II-3. Mélanome de Dubreuilh de l'arête nasale

#### 4. Le mélanome acrolentigineux ou mélanome acral

Il s'agit d'un mélanome survenant sur les extrémités (paumes et plantes). Il s'agit de la forme prédominante de mélanome sur les peaux noires et asiatiques. Le mélanome acral peut prendre plusieurs formes : tache brune, nodule couleur chair ou pouvant même ressembler à une verrue ou une tumeur saignante.



Figure II-4. Mélanome d'acral lentigineux

#### 5. Tumeurs mélanocytaire (nævus)

Les nævus pigmentaires congénitaux sont présents à la naissance chez 1 % des nouveaux nés. Leurs tailles sont très variables. Ils ont été classés en 3 groupes selon leur diamètre :

- petits (inférieurs à 1,5 cm),
- moyens (de 1,6 à 20 cm)
- géants (supérieurs à 20 cm).

Ils sont souvent pileux. Le risque de transformation en mélanome croît avec la taille du nævus. Il varie, pour l'ensemble de ces nævus, de moins de 1 % pour les petits à 5 %

environ pour les géants. L'exérèse à titre préventif n'est pas admise par tous les spécialistes et certains, arguant du faible risque évolutif, recommandent une surveillance. La survenue du mélanome est marquée par un épaissement nodulaire, et/ou un changement de coloration. Dans les nævus congénitaux géants, il peut s'agir de l'apparition d'un nodule dermique sous le nævus, décelable seulement par la palpation. Le traitement consiste en l'exérèse large du mélanome : elle n'est associée à l'exérèse complète du nævus pigmentaire congénital que si celle-ci est techniquement possible. [19]

#### **II.4. Les aspects cliniques**

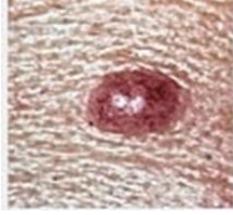
Le mélanome est une tumeur cutanée accessible à la vue dans la grande majorité des cas. Les patients consultent pour une lésion pigmentée de la peau d'apparition récente ou pour une modification d'un nævus ancien.

Le diagnostic de mélanome est évoqué par l'aspect clinique et confirmé par un examen pathologique. Toute lésion pigmentée, asymétrique, irrégulière, de couleur variable et de diamètre supérieur à 6mm a de forte chance d'être un mélanome, même si les signes ne sont pas complets par une biopsie. Ces critères peuvent être énumérés à l'aide de la règle alphabétique graduée **A, B, C, D, E**.

- **A = Asymétrie** : forme non circulaire, avec deux moitiés qui ne se ressemblent pas.
- **B = Bords irréguliers**, encochés, polycycliques, mal délimités.
- **C = Couleur non homogène** : présence de plusieurs couleurs (noir, marron, rouge, blanc ou bleu).
- **D = Diamètre** : diamètre en augmentation, en général supérieur à 6 mm.
- **E = Évolution** : toute tache pigmentée qui change d'aspect rapidement (forme, taille, épaisseur, couleur) est un signe d'alerte. [18]

Les aspects du mélanome cutané sont nombreux et souvent trompeurs. Une exérèse cutanée doit donc être facilement pratiquée devant toute lésion récemment modifiée ou d'aspect inhabituel. Le Tableau II-1 présente des exemples de situations cliniques pour chaque critère.

Tableau II-1. Illustration des aspects cliniques pour des cas normaux et des cas suspects

	Cas normal	Cas suspect
<b>A : Asymétrie</b>		
<b>B : Bords irréguliers</b>		
<b>C : Couleur</b>		
<b>D : Diamètre</b>		

## II.5. Evolution et pronostic

L'évolution spontanée d'un mélanome malin est très variable dans le temps. Certaines tumeurs progressent rapidement alors que d'autres évoluent sur plusieurs années avec des régressions et plus rarement des guérisons spontanées. L'évolution spontanée habituelle est marquée par un envahissement local avec extension possible à la peau adjacente ou à distance, aux ganglions régionaux et évolués en métastases, habituellement multiples.

Le pronostic est en règle très mauvais. Les patients avec métastase à distance cutanée ou ganglionnaire sans atteinte viscérale, ou avec un site métastatique viscéral unique (poumons) ont en règle une survie plus prolongée, tandis que ceux qui développent rapidement de multiples métastases viscérales, ou ont une survie très courte quel que soit le traitement. [18]

D'après la classification conjointe de l'AJCC (American Joint Committee on Cancer) et de l'UICC (Union Internationale Contre le Cancer), la plus utilisée pour les pays anglo-saxons et la plus fréquente dans la littérature. Il existe 4 stades :

- **Stade 0** : Il s'agit du stade le plus fréquent. La tumeur est alors limitée à l'épiderme de façon superficielle et elle ne s'est pas propagée. La chirurgie consiste à enlever le mélanome et une portion de la peau qui l'entoure, ce qui complète le traitement. Le pronostic est excellent.
- **Stade 1** : Ces mélanomes en sont encore à un stade précoce de développement. Ils ont moins de 2 mm d'épaisseur. Dépendant de la sévérité, la technique du ganglion sentinelle est parfois offerte.
- **Stade 2** : La tumeur à plus de 2 mm d'épaisseur. La technique du ganglion sentinelle est habituellement offerte. Après la chirurgie, le risque de réapparition du mélanome ou de propagation du cancer est modéré, à cause de la profondeur de la tumeur.
- **Stade 3** : Le mélanome a atteint les ganglions lymphatiques voisins, qui devront être enlevés.
- **Stade 4** : Il s'agit d'un stade avancé du mélanome. Le cancer s'est répandu à une autre partie du corps comme les poumons, le foie, le cerveau ou l'abdomen. Cette situation est rare.

## II.6. Rôle du traitement d'image dans la médecine

Il a été démontré qu'un diagnostic précis des lésions de peau pigmentée n'est achevé qu'en présence d'un consensus de trois experts cliniciens [22], compte tenu de l'étude réalisée par Koh et al. [23] qui prouve qu'un dermatologue expérimenté n'est capable de détecter que 75% des lésions cancéreuses. De là, les dermatologues ont préconisé le développement des systèmes informatiques d'aide au diagnostic pour la détection et l'identification anticipée du mélanome. Ainsi, les chercheurs sont de plus en plus intéressés par un

diagnostic automatique des lésions en se basant sur des techniques de traitement d'images. [20]

Le traitement d'images joue un rôle très important dans la médecine. Il est utilisé surtout pour le diagnostic (localiser, comprendre et quantifier les pathologies), mais ce n'est pas toujours évident d'utiliser les images telles qu'elles sont, des traitements d'images sont d'abord requis. On utilise souvent la chaîne de traitement suivante :



## 1. Prétraitements

Le terme de prétraitement est réservé aux processus qui améliorent la qualité de l'image acquise pour favoriser les analyses ultérieures. Généralement, cette étape consiste à augmenter les informations pertinentes et à diminuer les bruits parasites. Les prétraitements les plus couramment utilisés sont le débruitage, la convolution, la restauration, le rehaussement du contraste et l'homogénéisation du fond. Dans le cadre de notre application, le prétraitement se réduit à une amélioration de contraste, segmentation et quelques filtres morphologiques. Cet ensemble de traitement sont utilisés pour faciliter l'extraction des paramètres.

### 1.1. Suppression des poils

Beaucoup d'images de la peau contiennent des poils ayant une teinte de couleur semblable à celle de la lésion ce qui peut dégrader la segmentation. Evidemment, le fait de raser les poils avant l'acquisition des images est une solution possible.

Cependant, cette solution est peu pratique surtout dans le cas de plusieurs lésions réparties sur tout le corps. De là, on utilise souvent l'approche "DULLRAZOR" permettant de supprimer les poils épais dans des images dermatologiques.

La technique du Dullrazor est réalisée en 3 étapes :

- identifier l'emplacement des poils par un opérateur de fermeture morphologique ;
- vérifier les poils par leurs formes pour les distinguer du bruit causé par le processus d'acquisition ;

- remplacer les pixels poils par une interpolation bilinéaire des pixels non-poils voisins.



Figure II-5. Résultat de la suppression des poils. (a) Image originale, (b) Image après suppression des poils.

### 1.2.Filtre moyen

Le principe est très simple: un pixel est remplacé par la moyenne de lui-même et de ses voisins. C'est dans la définition du voisinage en connexité 4 ou 8, ou même encore plus large. Le filtre moyen est un filtre passe-bas permettant ainsi d'éliminer les hautes fréquences, correspondants aux bruits. Son inconvénient est qu'il élimine également les hautes fréquences correspondant aux détails de l'image : il rend ainsi l'image moins bruitée mais plus floue.

Soit  $A[i, j]$  une image numérique. L'image traitée (filtrée) est donnée par la formule suivante:

$$B[i, j]_{mn} = h(m, n). A[i - m, j - n] \quad \text{II-1}$$

**Avec :**

B : l'image filtrée

A : l'image originale

h : le masque du filtre

### 1.3.Filtre médian

Le filtre médian est une amélioration du filtre conservatif. Son principe consiste à remplacer un pixel par la médiane de ses voisins. Ainsi, même si plusieurs pixels voisins sont bruités, on peut corriger le pixel courant. Ce filtre induit cependant un lissage puisque même des pixels corrects peuvent être modifiés. De plus, ce filtrage est plus coûteux. On pensera donc, lors de l'implémentation, à utiliser un algorithme de tri rapide tel que le

filtre médian permet d'obtenir de bons résultats sur du bruit poivre et sel mais est aussi peu performant que le filtre conservatif pour le bruit gaussien.

## 2. Segmentation

La segmentation d'image joue un rôle prépondérant dans le traitement d'image et dans la vision par ordinateur. Cette étape se situe entre d'une part l'acquisition de l'image et son amélioration et d'autre part sa description et la prise de décision finale conséquente.

La segmentation est un traitement de bas-niveau qui consiste à effectuer une partition de l'image en régions homogènes par rapport à un ou plusieurs critères. Les régions obtenues se distinguent les unes des autres par des différences significatives selon ces mêmes critères. La segmentation consiste à extraire des points, des lignes ou des régions. Le choix d'une technique de la segmentation est liée à plusieurs facteurs comme : La nature de l'image, les conditions d'acquisition (bruit), les primitives à extraire (contour, texture). Il existe de nombreuses méthodes de segmentation que l'on peut regrouper en trois principales méthodes:

- Segmentation par seuillage.
- Segmentation basée sur les contours.
- Segmentation basée sur les régions.

### 2.1.Segmentation par seuillage

Le seuillage a pour objectif de segmenter une image en plusieurs classes en n'utilisant que l'histogramme. Il existe pour cela de très nombreuses méthodes dont la plupart s'appliquent correctement si l'histogramme contient réellement des pics séparés. De plus, ces méthodes ont très souvent été développées pour traiter le cas particulier de la segmentation en deux classes (passage à une image binaire) et leur généralité face aux cas multi-classes n'est que très rarement garantie. L'approche par seuillage se base sur deux principales méthodes : seuillage manuel et seuillage automatique.

### 2.2.Seuillage global

Le principe du seuillage global utilise une valeur seuil à partir de laquelle on peut choisir à quelle classe le pixel appartient. La transformée peut s'écrire sous la forme :

$$\forall i, j \in N \times M I(i, j) = \begin{cases} 1 & \text{si } f(i, j) > S \\ 0 & \text{sinon} \end{cases} \quad \text{II-2}$$

Avec :

- $N \times M$ : nombre de colonnes et de lignes de l'image.
- $f(i, j)$ : image binarisée.
- $S$  : seuil de binairisation.

### 2.3. Seuillage manuel

En pratique, on effectue des seuillages à plusieurs niveaux (extraire les pixels dont la luminosité est comprise entre 2 valeurs SB et SH).

#### Algorithme

- Pour tous les pixels de l'image
- Soit  $g = A(i, j)$  le niveau de gris d'un pixel quelconque
- Si  $SB < g < SH$  Alors  $g = 1$  (blanc).
- Sinon  $g = 0$  (noir).

Pour cela on part de l'analyse de l'histogramme sur laquelle on travaille (en général le NG) et on cherche alors des modes, puisqu'une région est considérée comme un ensemble de pixels ayant des niveaux de gris voisins.

### 2.4. Seuillage local

Le principe du seuillage local est de déterminer quel seuil il faut utiliser. Pour réaliser cette étude locale, les techniques utilisent une fenêtre d'étude centrée sur le pixel à étudier.

Mathématiquement, le calcul du seuil peut s'écrire ainsi :

$$S(i, j) = \frac{\max(f(i, j)) + \min(f(i, j))}{2} \quad \text{II-3}$$

Avec :

- $S(i, j)$  : seuil à appliquer pour le point  $i, j$  ;
- $\text{Max}(f(i, j))$  : valeur du niveau de gris maximale dans une fenêtre centrée en  $(i, j)$  de taille  $[N \times M]$  ;
- $\text{Min}(f(i, j))$  : valeur du niveau de gris minimale dans une fenêtre centrée en  $(i, j)$  de taille  $[N \times M]$  ;

Cependant, ce filtre est très sensible au bruit du fond. À cause de la prise en compte du maximum et du minimum uniquement, dans le cas où la fenêtre est uniquement sur du fond, le bruit sera interprété comme objet, car le seuil sera bas. On propose une méthode

similaire sur le principe, mais prenant en compte d'autres paramètres. Dans sa méthode, le seuil est calculé ainsi :

$$S(i, j) = \mu(i, j) + k \cdot h(i, j) \quad \text{II-4}$$

Avec :

- $S(i, j)$  : seuil à appliquer pour le point  $i, j$  ;
- $h(i, j)$  : valeur de l'écart type dans une fenêtre centré en  $i, j$  de taille  $N \times M$  ;
- $\mu(i, j)$  : valeur moyenne des niveaux de gris dans la même fenêtre ;
- $k$  : constante fixée le plus généralement à 0, 2

### 2.5. Méthode d'Otsu

La méthode développée par Otsu ne s'applique que dans le cas de la binarisation ; segmentation d'image en deux catégories (le fonds et les objets). L'idée est de chercher un seuil permettant d'obtenir les deux classes en minimisant une fonction de coût. On va alors essayer diverses valeurs de seuil et choisir celui qui sépare l'histogramme de façon optimale en deux classes (qui minimise la variance intra-classe). En supposant que le nombre de niveaux de gris est 256 et que l'histogramme est noté  $h(i)$ . Alors on peut définir une mesure de variance intra-classe par [24]:

$$\sigma_{intra}^2(s) = t_1(s) \cdot \sigma_1^2(s) + t_2(s) \cdot \sigma_2^2(s) \quad \text{II-5}$$

$$\text{Avec : } t_1(s) = \sum_{i=0}^{s-1} h(i) \quad \text{et} \quad t_2(s) = \sum_{i=s}^{255} h(i)$$

$\sigma_1^2(s)$  : variance des pixels dont le NG est  $<$  seuil  $S$

$\sigma_2^2(s)$  : variance des pixels dont le NG est  $\geq$  seuil  $S$

Il vient alors :

$$\sigma_{intra}^2 = \sum_{i=0}^{s-1} h(i) \cdot (i - \mu_1)^2 + \sum_{i=s}^{255} h(i) \cdot (i - \mu_2)^2 \quad \text{II-6}$$

$\mu_1$  et  $\mu_2$  sont les moyennes des NG de chaque classe. On peut alors essayer toutes les valeurs du seuil  $S$  possibles et on garde celui qui rend  $\sigma_{intra}^2(s)$  minimum.

Cette méthode donne de bons résultats dans le cas d'une image bimodale.

### 2.6.Segmentation en régions

La croissance des régions est une technique de segmentation qui regroupe itérativement des points connexes en des larges régions selon des critères d'homogénéité. Les mesures statistiques apparaissent comme des outils fiables pour définir l'homogénéité des régions. Le succès d'une segmentation est ainsi directement associé au choix des mesures et d'un seuil d'arrêt.

#### Algorithme

- Initialisation avec quelques pixels (germes)
- Chaque germe fusionne avec un 1<sup>er</sup> pixel pour former une région. Le processus est itératif.
- Le pixel est ajouté à une région spécifique, selon une règle donnée.

### 3. Détection de contour :

Un contour est la frontière entre deux régions homogènes, qui sont dans notre cas le fonds (la peau) et l'objet (la lésion). La détection de contour est une opération fondamentale de la segmentation des images médicales.

Les algorithmes de détection de contour consistent à identifier les contours d'une image en repérant les forts changements d'intensité lumineuse de l'image. Alors ils basent sur la mesure de la discontinuité.

Il existe pour cela toute une gamme classique d'opération de filtrage par convolution. Il y a aussi les filtres différentiels qui donnent des valeurs élevées aux points où la variation des niveaux de gris est différente.



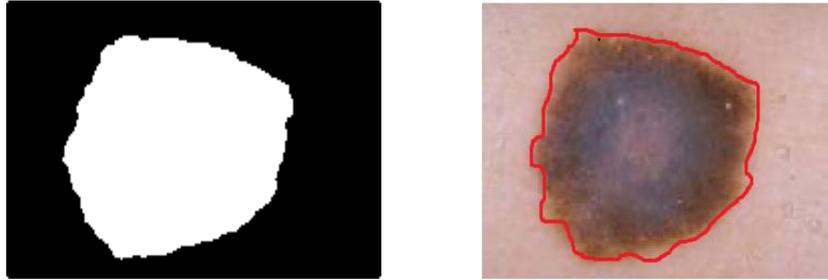


Figure II-6. Résultat de la segmentation utilisant la méthode d'Otsu.

### 3.1. Méthodes de convolution

Ils utilisent une variation des masques : horizontaux, verticaux et diagonaux.

$$\begin{array}{ccc}
 -1 & -1 & -1 & -1 & -1 & 2 & -1 & 2 & -1 \\
 2 & 2 & 2 & -1 & 2 & -1 & -1 & 2 & -1 \\
 -1 & -1 & -1 & 2 & -1 & -1 & -1 & 2 & -1
 \end{array}$$

### 3.2. Méthodes de dérivation

- Gradient : soit une image A (i, j).

$$|\nabla A(i, j)| = \max \left\{ \left| \frac{\Delta A}{\Delta j} \right|, \left| \frac{\Delta A}{\Delta i} \right| \right\} \quad \text{II-7}$$

Avec

$$\frac{\Delta A}{\Delta j} = A(i, j+1) - A(i, j) \text{ et } \frac{\Delta A}{\Delta i} = A(i+1, j) - A(i, j)$$

#### Algorithme

##### Début

- pour tous les pixels de l'image, calculer
- Grad (j) = A (i+1, j+1) - A (i, j)
- Grad (i) = A (i+1, j) - A (i, j+1)
- B(I, j) = sup (Grad(j) , Grad(i) )

##### Fin

## 4. Opérateurs morphologiques

La morphologie mathématique est une théorie de traitement non linéaire de l'information apparue en France dans les années 60 [25], et qui est aujourd'hui largement utilisée en analyse d'images. Contrairement au traitement linéaire des images, la morphologie mathématique ne s'appuie pas sur le traitement du signal, mais repose sur la théorie des

ensembles, ce qui en fait une discipline relativement « auto-contenue » et formant un tout cohérent.

Le principe de base de la morphologie mathématique est de comparer l'image à analyser par rapport à un ensemble de géométrie connue appelé **élément structurant** que l'on déplace de façon à ce que son origine passe par toutes les positions de l'image, pour mettre en évidence certaines caractéristiques de l'image.

L'élément structurant est une forme qui définit une opération morphologique. On peut le voir comme un masque. Il y a deux opérations morphologiques de base [26] :

- **Erosion :**

L'érosion est une opération qui va avoir tendance à réduire la masse des objets. On considère que dans une image binaire tous les pixels objets ont une valeur de 1 et les pixels du fond ont une valeur de 0. Pour chaque pixel considéré individuellement, on regarde sa valeur et la valeur des 4 pixels immédiatement voisins (c'est-à-dire ceux partageant une face avec le pixel considéré). Parmi ses valeurs, on recherche la plus petite. Puis, on remplace dans l'image résultat la valeur du pixel considéré par cette valeur minimum. Pour une image binaire, le résultat de ce traitement revient à transformer en pixels du fond les pixels objets en contact direct avec des pixels du fond. Cela revient à "raboter" d'une couche de pixels les masques des objets.



Figure II-7. Érosion morphologique. (a) masque original ; (b) érosion, 3itération ; (c) érosion 6 itérations

- **Dilatation :**

La dilatation est l'opération complémentaire de l'érosion. Dans ce cas, on recherche la valeur la plus grande dans un voisinage. Pour une image binaire, le résultat de ce traitement revient à transformer en pixels objets, les pixels du fond en contact direct avec

des pixels objets. Cela revient donc à épaissir d'une couche de pixels les masques des objets.



Figure II-8. Dilatation morphologique. (a) masque original ; (b) dilatation 3 itérations ; (c) dilatation 6 itérations

- **Fermeture**

La fermeture consiste à appliquer à l'image binaire une dilatation d'une profondeur donnée suivie d'une érosion de profondeur identique. La dilatation va permettre de boucher les trous des masques, la dilatation comble tous les trous dans le masque du noyau. Cependant on constate que le noyau a maintenant une forme empâtée et surdimensionnée. L'érosion qui suit permet de compenser cet épaississement. On constate que le masque du noyau a retrouvé ses dimensions originales alors que son contour apparaît moins échancré. La fermeture a aussi effectué un lissage du contour.



Figure II-9. Fermeture morphologique

- **Ouverture**

L'ouverture consiste à appliquer à l'image une érosion d'une profondeur donnée suivie d'une dilatation de profondeur identique. L'érosion va permettre d'éliminer les petites

structures isolées. Par exemple (Figure II-10), l'érosion permet de gommer les petits artefacts et ne laisser subsister que le masque du noyau. La dilatation permet de retrouver les dimensions originales du noyau.

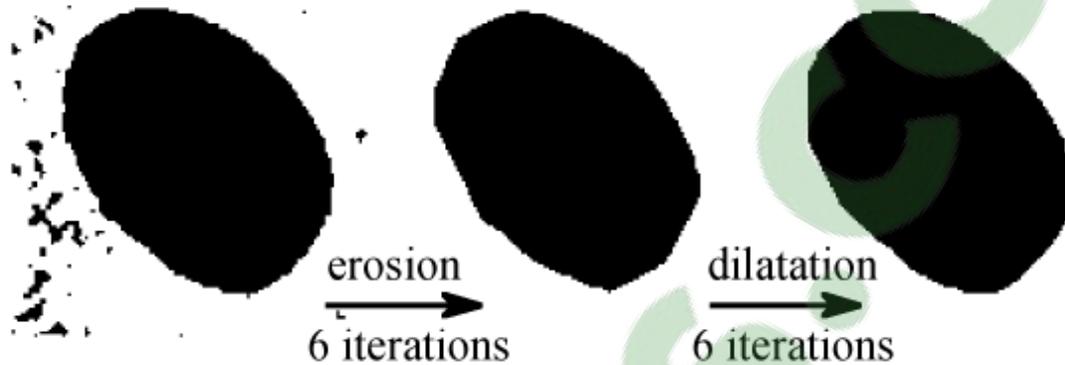


Figure II-10. Ouverture morphologique

## II.7. Extraction des paramètres

C'est la partie la plus importante dans notre travail. Après qu'on fait les traitements nécessaires sur l'image, l'étape de la caractérisation est nécessaire pour faire le diagnostic. Cette étape vise la quantification des caractéristiques de la lésion en produisant un vecteur descriptif de la lésion (VDL). Ces caractéristiques modélisent les signes cliniques utilisés par les dermatologues dans le diagnostic médical afin de décrire la malignité de la lésion. Ils peuvent être regroupés selon différentes formulations universelles (ABCD, ABCDE, 3C, Seven checklist, ...). Nous allons discuter la formulation ABCD modélisant la malignité d'une lésion par l'Asymétrie de la forme, l'irrégularité du Bord, l'hétérogénéité de la Couleur et le Diamètre de la forme. Notre sélection des paramètres appropriés est guidée par la littérature et par des recherches similaires réalisées auparavant.

### 1. La règle ABCD

- **Asymétrie**

L'asymétrie est une caractéristique essentielle dans la différenciation des tumeurs malignes et des lésions bénignes. Elle est l'un des critères de la règle ABCD (A : Asymétrie). Elle est généralement évaluée par les dermatologues grâce à l'observation de la lésion par comparaison des deux moitiés de la lésion selon l'axe principal. Pour élaborer un algorithme visant à calculer un indice d'asymétrie, il doit utiliser les axes principaux de la

lésion pour quantifier l'indice de symétrie. Un indice est calculé à partir de la plus petite différence d'aire entre l'image de la lésion et l'image de la lésion réfléchiée à partir de l'axe principal. Cette valeur est rapportée à l'aire de la lésion, ce qui permet de fournir un pourcentage d'asymétrie.

Cette méthode, évaluée sur 86 tumeurs expertisées au préalable par deux dermatologues, a conclu que la concordance entre la prédiction du système et l'avis des médecins est plutôt bonne. Ils ont pu en outre conclure, au vu du diagnostic, que 63% des lésions malignes ont été évaluées comme étant asymétriques tandis que 28% des lésions bénignes étaient symétriques.

Une autre méthode calcule l'indice d'asymétrie à partir des différences entre les aires (aire de l'image initiale et l'aire de l'image réfléchiée de  $180^\circ$  par rapport au centre de gravité de la lésion).

- **Centre de gravité**

Pour mesurer l'indice d'asymétrie, nous avons besoin de déterminer les axes qui doivent passer par le centre de gravité de la tumeur. Dans cette section, nous nous intéressons à la détermination du centre de gravité. A partir du masque de la lésion, nous pouvons considérer que l'image s'apparente à un ensemble de points de coordonnées  $(x, y)$ . Le centre de gravité est alors calculé comme étant la moyenne des coordonnées en  $x$ , et la moyenne des coordonnées en  $y$ .

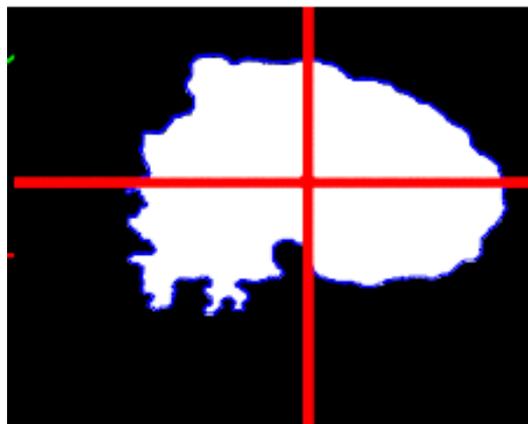


Figure II-11. Détection d'axes de symétries

- **Bordure**

L'irrégularité est un critère de suspicion de la malignité d'une lésion, qui fait partie des critères pris en compte lors de l'examen clinique. C'est un critère global, difficile à

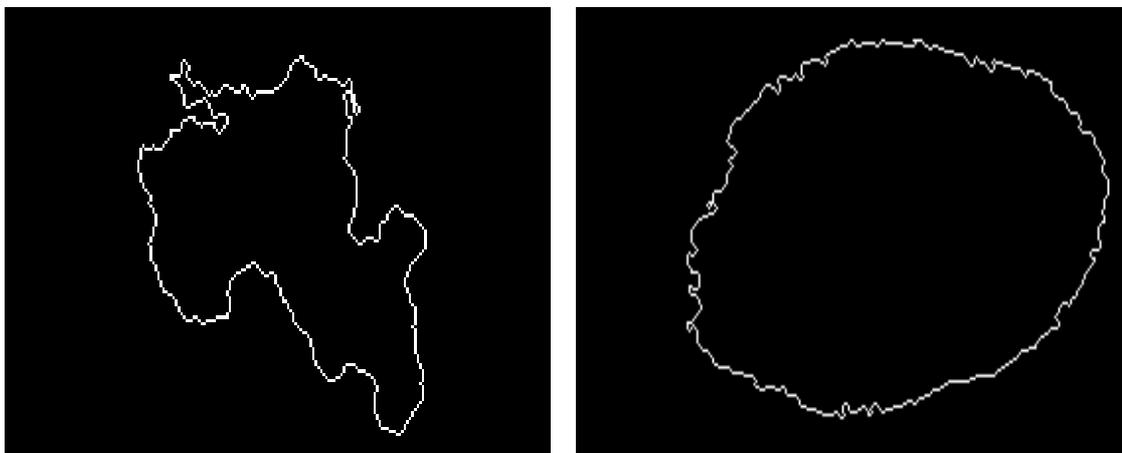
formaliser. Par conséquent, les dermatologues ont parfois du mal à s'accorder sur ce critère.

- **Compacité**

La compacité est définie par l'équation suivante :

$$C = \frac{p^2}{4\pi a} \quad \text{II-8}$$

Avec **a** : aire de la tumeur, **p** : son périmètre.



(a) Compacite= 2.70

(b) Compacite= 1.31

Figure II-12. Exemples d'application, a : mélanome (compacité = 2.70) ; b : tumeur bénigne (Compacité= 1.31)

- **Périmètre**

L'aire et le périmètre sont calculés à partir du masque obtenu par l'opération de segmentation en termes de nombres de pixels appartenant au contour et nombres de pixels définissant la surface

- **Diamètre**

L'axe le plus long de la lésion, Il est l'un des critères de la règle ABCD (D : diamètre inférieur à 6 mm). Le diamètre est une caractéristique qui aide à la différenciation des tumeurs malignes et des lésions bénignes.

La distance polaire (distance entre le centre de gravité et le bord) exprimée par le rapport entre le minimum et maximum.

## 2. La couleur

L'homogénéité de la couleur est évaluée en comptant le nombre de transitions clair-sombre et sombre-clair en parcourant la lésion horizontalement et verticalement. Ce nombre est faible pour les lésions bénignes et élevé pour les lésions malignes à cause de leurs couleurs hétérogènes.

Les paramètres de texture que nous avons choisis mesurent d'une manière locale la répartition spatiale des couleurs et l'homogénéité de la texture des lésions. Dans notre cas, nous avons utilisés les paramètres de texture suivants :

- La moyenne des composantes couleurs.

$$\text{Moy} = \frac{1}{M \times N} \sum_{i=0}^{Ng} \sum_{j=0}^{Ng} f(i, j) \quad \text{II-9}$$

- La variance : constitue une estimation du degré de dispersion des niveaux des composantes couleurs autour de leur moyenne.

$$Va^2 = \frac{1}{M \times N} \sum_{i=0}^{Ng} \sum_{j=0}^{Ng} (f(i, j) - \text{Moy})^2 \quad \text{II-10}$$

- La dissymétrie : mesure le degré d'asymétrie des niveaux des composantes couleurs par rapport à leur moyenne.

$$\text{Dissymétrie} = \frac{1}{M \times N} \cdot \frac{\sum_{i=0}^{Ng} \sum_{j=0}^{Ng} (f(i, j) - \text{moy})^3}{Va^3} \quad \text{II-11}$$

- Le second moment angulaire (ou énergie) : l'énergie mesure l'uniformité de la texture de l'image. Plus l'image est uniforme, moins il y a de transitions de niveaux de gris et donc plus la somme des carré des éléments de la matrice est faible. Ce paramètre a une valeur faible lorsque les pixels ont des valeurs très proches.

$$\text{Energie} = \sum_{i=0}^{Ng} \sum_{j=0}^{Ng} [fc(i, j)]^2 \quad \text{II-12}$$

- Le contraste : dans la mesure où chaque terme de la matrice est pondère par sa distance à la diagonale. Ce paramètre a une valeur numérique importante si les pixels sont concentres hors diagonale.

$$\text{Contraste} = \sum_{i=0}^{Ng} \sum_{j=0}^{Ng} [(i-j)^2] fc(i,j) \quad \text{II-13}$$

- La corrélation est un indice qui mesure la dépendance linéaire des niveaux de gris dans l'image. Ce paramètre a une grande valeur quand les valeurs sont uniformément distribuées dans la matrice de cooccurrence et une faible valeur dans le cas contraire.

$$\text{Corrélation} = \sum_{i=0}^{Ng} \sum_{j=0}^{Ng} \frac{(i-u_i)(j-u_j)}{Va2} fc(i,j) \quad \text{II-14}$$

## II.8. Conclusion

Comme tous types de cancers, le mélanome ne peut être traité que lors qu'il est découvert dans ces premiers stades. Un tel diagnostic précoce est difficile pour le médecin, il est important d'utiliser un système automatique pour le dépistage précoce.

Les systèmes d'aide au diagnostic génèrent un dépistage en basant sur des images de la lésion. Ces images doivent être convenablement traités et analysés pour obtenir une bonne précision.

Dans ce chapitre, nous avons présenté les différents traitements nécessaires pour l'aide au diagnostic des tumeurs de la peau.

# CHAPITRE III

---

**RESULTATS ET INTERPRETATIONS**

---

### **III.1. Introduction**

Les techniques d'imagerie occupent une place croissante dans l'exercice de la médecine. Les images peuvent intervenir aussi bien dans le processus de décision diagnostique que pronostique ou thérapeutique.

Qu'elle soit faite d'emblée ou secondairement, la numérisation des images facilite leur traitement, leur stockage et leur transmission, à distance. Ces traitements se sont considérablement développés avec l'apparition des techniques efficaces de compression et de stations de travail présentant les capacités de stockage et la puissance de calcul nécessaire à la mise en œuvre locale des algorithmes de traitement.

Dans le domaine de la dermatologie, les techniques de manipulation des images ne sont pas assez développées. Il existe toujours un manque de logiciels dédiés pour l'aide au diagnostic. Même en développant des applications assurant cette fonction, le problème de lisibilité et d'accessibilité se pose toujours.

Afin de faciliter l'interaction entre l'utilisateur et la machine, nous avons développé une interface graphique sous Visual C#. Celle-ci permet à l'utilisateur d'effectuer plusieurs opérations parmi lesquelles on peut citer le chargement du fichier image (dermatologique), segmentation, traitement et extraction des paramètres requises par le dermatologue.

Dans ce chapitre nous allons introduire l'environnement .NET et le langage C Sharp, et expliquer notre travail étape par étape, en détaillant chaque fonction utilisée. Finalement, nous allons voir le rendement, et discuter les résultats obtenus.

### **III.2. C Sharp.NET**

.NET est une technologie inaugurée par Microsoft au début de l'année 2002. Il s'agit de proposer aux développeurs un environnement de programmation puissant et versatile. .NET regroupe un ensemble de langages, une bibliothèque de classes riche et complète et une machine virtuelle.

Les langages disponibles, tout à objets, répondent à des spécifications précises de Microsoft, les *Common Language Specifications* ou CLS. Les langages qui suivent ces exigences sont dits « managés ». Beaucoup de langages dits classiques ont été adaptés à

.NET et sont donc devenus managés. Il existe le Visual Basic .NET : il s'agit du C# (prononcer « C Sharp ») [27].

Le langage C# (C Sharp) est un langage objet créé spécialement pour l'environnement Microsoft.NET. L'équipe qui a créé ce langage a été dirigée par Anders Hejlsberg, un informaticien danois qui avait également été à l'origine de la conception du langage Delphi pour la société Borland (évolution objet du langage Pascal) [28].

### **Pourquoi l'orienté objet ?**

L'orienté objet a été introduit par le langage SmallTalk (développé par la société Xerox) en 1972. L'orienté objet est un ensemble de règles, de concepts, d'outils du langage de programmation pour rendre le développement plus aisé grâce au groupement d'éléments. Les objets sont donc des entités qui permettent de grouper ensemble des fonctions et des variables en facilitant leurs gestions par groupe et en ouvrant de multiples possibilités. Définir un objet peut paraître comme ajouter un outil, Le framework .NET apporte lui un set imposant d'objets existants. Un langage Orienté objet n'est pas les objets, mais une syntaxe et des mots clefs pour les manipuler et les créer [28].

### **Applications windows**

Les applications Windows sont basées sur une ou plusieurs fenêtres constituant l'interface entre l'utilisateur et l'application. Pour développer ce type d'application, nous avons à notre disposition dans le Framework. NET un ensemble de classes permettant la conception de l'interface de l'application. Ces éléments sont fréquemment regroupés sous le terme Technologie Windows Forms. Une application basée sur les Windows Forms utilisera un ou plusieurs formulaires pour construire l'interface utilisateur de l'application. [29]

En créant un nouveau projet, on doit choisir « windows forms application »

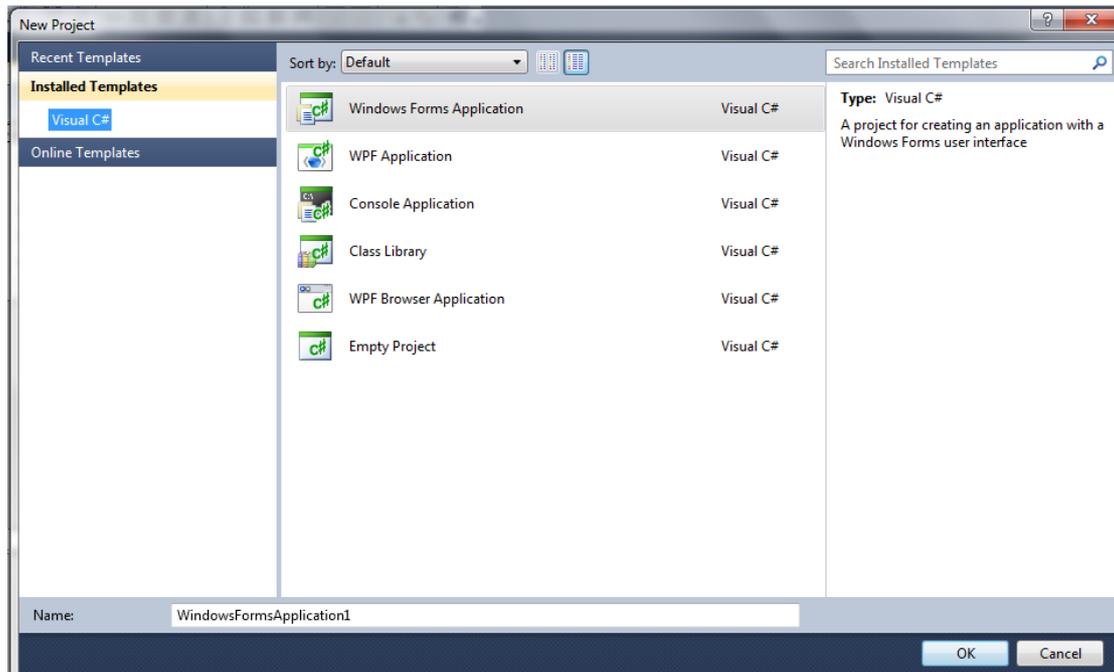


Figure III-1.Fenêtre d'un nouveau projet c#

Lors de sa création, notre projet contient, par défaut, une Windows Form (une fenêtre vide) appelée **Form1.cs**.

### Bibliothèque Aforge.NET

AForge.NET Framework est un environnement C# conçu pour les développeurs et les chercheurs dans les domaines de la vision par ordinateur et Intelligence Artificielle, traitement de l'image, les réseaux de neurones, algorithmes génétiques, l'apprentissage automatique, et la robotique. Le cadre est constitué par l'ensemble des bibliothèques et de l'échantillon applications, qui démontrent leurs caractéristiques.

C'est une bibliothèque externe, qui n'est pas inclus par défaut dans le Framework .NET, pour l'utiliser on doit tout d'abords l'ajouter comme référence [3].

### III.3. Interface graphique

La qualité de l'interface est un facteur important d'acceptabilité. Les médecins sont peu attirés par l'utilisation de terminaux d'ordinateurs. Les interfaces doivent faire l'objet d'une ergonomie particulièrement étudiée permettant d'utiliser le système dans les conditions de travail habituelles. L'utilisation doit être simple et la phase d'apprentissage réduite au maximum.

Nous avons choisi comme un nom pour notre application « Dermatology Wizard » qui signifie « l'assistant de dermatologie », c'est un nom qui convient bien la fonction de notre application. Le logo est un petit ruban noir, qui est par convention le symbole du mélanome cutané.

Lorsque on crée le WinForm, les bibliothèques suivantes sont incluses par défaut

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;
```

Avant de programmer l'interface, nous devons d'abord préparer et initialiser toutes les variables et les bibliothèques nécessaires pour accomplir les tâches.

Commençant à introduire la bibliothèque aforge.NET et ses accessoires,

```
using AForge;  
using AForge.Imaging;  
using AForge.Imaging.Filters;  
using AForge.Math;
```

Et par la suite nous aurons besoin de l'espace entrée/sortie, alors il faut l'ajouter par l'instruction suivante :

```
using System.IO;
```

## 1. Les variables

Nous avons dit que le C Sharp est un langage orienté objet, qui est basé sur le principe des objets. C'est-à-dire chaque bloc est un objet indépendant du reste du programme, alors si on introduit une variable dans un bloc, elle ne peut pas être reconnue dehors. Les blocs sont limités par des accolades {}. Tandis qu'il est indispensable d'utiliser la variable dans plusieurs endroit.

Pour résoudre ce problème nous avons initialisé les variables au début du programme, pour qu'ils soient inclut dans tous les objets.

```
private Bitmap image;
private Bitmap image_copie;
private Bitmap image_gray;
private Bitmap image_BW;
private Bitmap image_ROI;
private Bitmap image_edge;
double surface;
double perimeter;
int X; int Y;
double mean_R; double mean_G; double mean_B;
double ecart_R ; double ecart_G ; double ecart_B ;
int diameter;
double compacity;
int channel_index;
```

Ces variables n'étaient pas introduites au début du travail, car il est impossible de prévenir nos besoins. Elles sont initialisées au fur et à mesure au cours du travail.

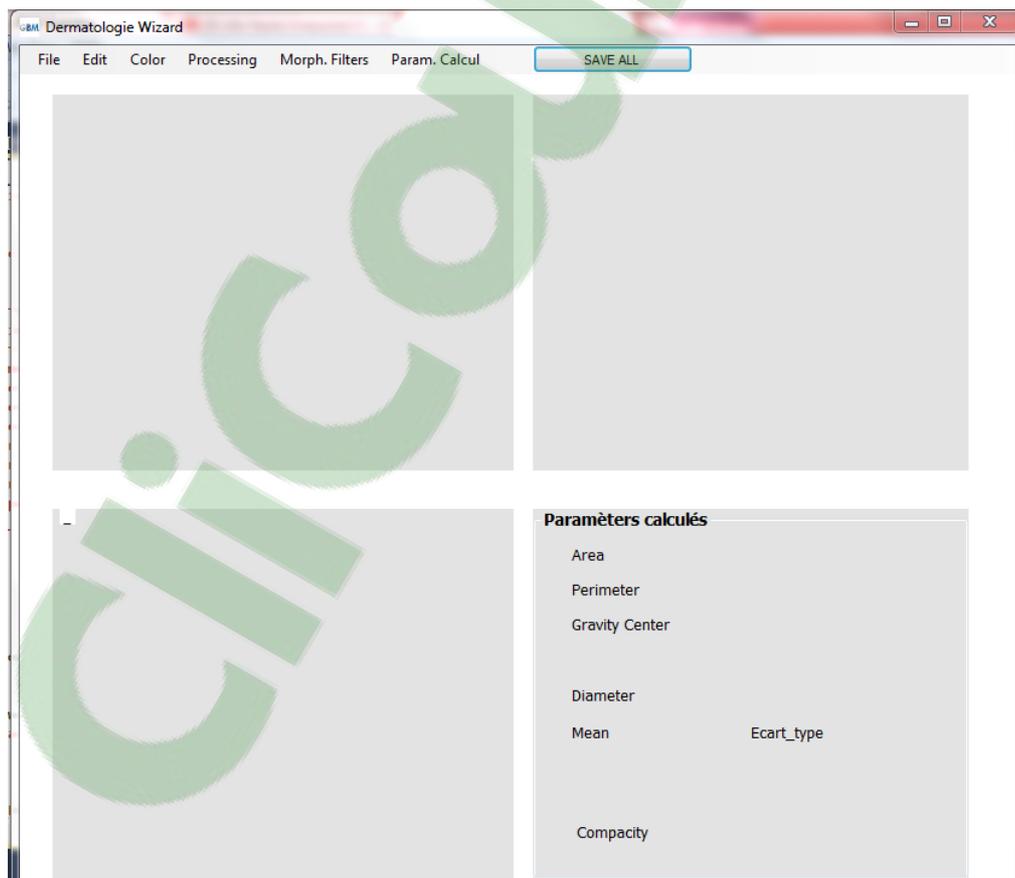
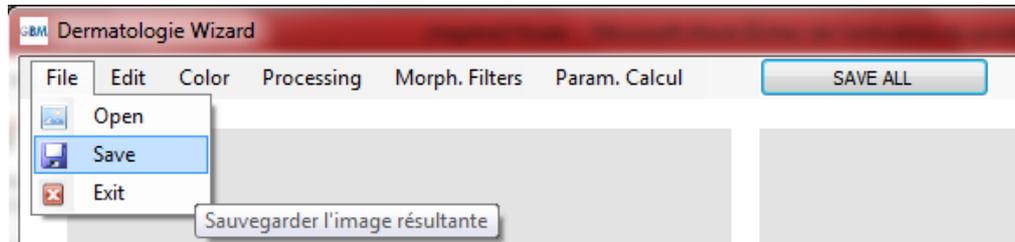


Figure III-2. Interface réalisée

## 2. Les différentes composantes de l'interface

### 2.1. La fenêtre « file » : fichier

Cet onglet permet le lancement de travail, l'ouverture et l'enregistrement de l'image.

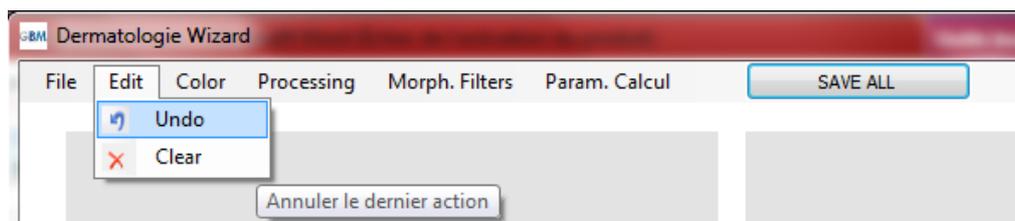


- « open » Ouvrir une image  
Cette fonction permet de lire et de charger les images. Elle donne à l'utilisateur la main pour choisir une image de n'importe quel répertoire sur l'ordinateur.  
Nous avons utilisé la classe « open file dialog » qui est générée par l'environnement framework.NET.
- « Save » Sauvegarder une image  
Cette fonction permet de sauvegarder l'image dans un répertoire choisi par l'utilisateur, et sous un nom de fichier fourni par l'utilisateur.  
Pour ce but, nous avons utilisé la classe « Save file dialog », qui est aussi fourni par l'environnement .NET.
- « exit » : quitter l'interface  
Cette fonction permet de quitter l'interface. L'opération se fait par une seule instruction

```
this.Close();
```

### 2.2. La fenêtre « edit » ou éditer

Cet onglet regroupe les fonctions de modification



- « Undo » : annuler

En traitement d'image, les résultats des filtres et des opérations sont parfois imprédictible. Très souvent on fait des traitements qui nous ne plaisent pas, et on doit revenir en arrière pour corriger l'erreur.

Cette fenêtre permet d'annuler la dernière modification. Pour cela nous avons introduit une variable appelée « image\_copie » : à chaque fois qu'on exécute une fonction, on commence par sauvegarder l'image courante (avant le changement) dans la variable « image\_copie », puis on applique notre tâche.

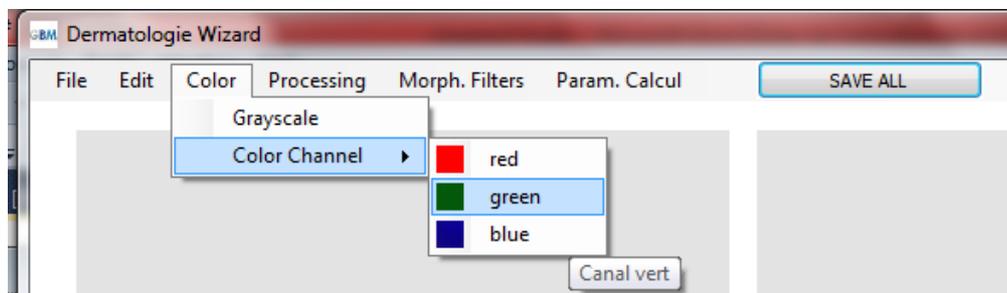
Le bouton annuler sert à récupérer l'image avant le changement, qui est tout simplement « image\_copie ».

- « Clear » : supprimer

Cette fonction permet de faire une remise à zéro du système.

### 2.3. La fenêtre « color » : couleur

Cet onglet permet de travailler sur la couleur. Elle reçoit une image couleur, et redonne à la sortie une image en niveau de gris.



- « Grayscale » : niveau de gris

Cette fonction permet de rendre l'image en niveau de gris. Pour cela nous avons utilisé une classe de la bibliothèque AForge.Imaging.Filters qui est appelée « Grayscale ». Elle est utilisée comme suit :

```
Grayscale filter = new Grayscale(0.2125, 0.7154, 0.0721);
image_gray = filter.Apply(image);
```

- « color channel » ou canal couleur

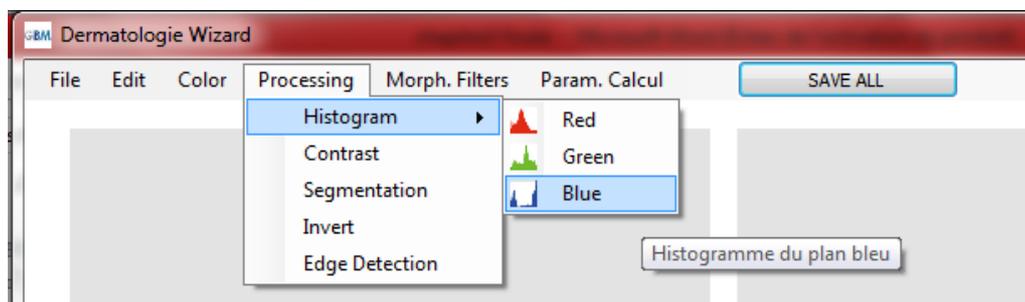
Cette fonction permet d'extraire les plans couleur. Pour cela nous avons utilisé la classe « ExtractChannel » de la bibliothèque AForge.Imaging.Filters.

on doit l'initialiser en spécifiant à chaque fois la couleur désirée, voilà un exemple du canal rouge :

```
ExtractChannel extractFilter = new ExtractChannel(RGB.R);
//Bitmap channel = extractFilter.Apply(image);
image_gray = extractFilter.Apply(image_copie);
```

#### 2.4. La fenêtre « processing » : traitement

Le plus important dans l'interface que nous avons réalisée est l'étape du traitement. Cet onglet regroupe l'ensemble des traitements appliqués sur l'image en niveau de gris.



- « Histogram » ou histogramme

Le calcul de l'histogramme d'une image permet de représenter le nombre de pixels pour chaque niveau de gris (densité) de l'image.

Cette fonction permet de calculer et ploter l'histogramme de chaque plan couleur.

- « Contrast » : contraste

Cette fonction permet d'améliorer le contraste de l'image, elle est appliquée sur une image en niveau de gris. Nous avons utilisé une classe de l'espace AForge.Imaging.Filters appelée « ContrastCorrection », en fixant la valeur de contraste dans le programme. mais pour donner plus de contrôle à l'utilisateur et obtenir une amélioration plus précise nous avons créé une nouvelle forme, qui permet de saisir une valeur et l'utiliser comme facteur de contraste dans notre fonction.

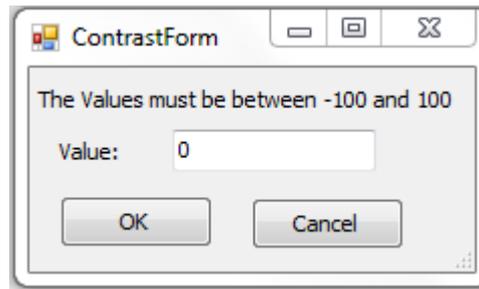


Figure III-3. fenêtre de contraste « ContrastForm »

### Algorithme

- Créer une nouvelle forme : Project>> Add Windows Form
- Personnaliser la forme, pour qu'elle soit lisible et claire
- Dans le programme, ContrastForm.cs , on va initialiser les boutons « ok » et « cancel »

```
public ContrastForm()
{
    InitializeComponent();
    btnOK.DialogResult = DialogResult.OK;
    btnCancel.DialogResult = DialogResult.Cancel;
}
```

- Initialiser la valeur que l'utilisateur va rentrer, comme variable pour l'utiliser dans notre programme principale

```
public byte ContrastValue
{
    get
    {
        if (string.IsNullOrEmpty(txtContrastValue.Text))
            txtContrastValue.Text = "0";
        return Convert.ToByte(txtContrastValue.Text);
    }
    set { txtContrastValue.Text = value.ToString(); }
}
```

- Revenant au programme initial, on va utiliser notre variable dans la classe

```
ContrastForm cFrm = new ContrastForm();
cFrm.ContrastValue = 0;
ContrastCorrection filter = new
ContrastCorrection(cFrm.ContrastValue);
image_gray = filter.Apply(image_gray);
```

il est important de respecter le type de variable créée dans la nouvelle fenêtre, elle doit être compatible avec le type demandé de la classe « ContrastCorrection ».

- « Segmentation »

La phase de segmentation consiste à isoler l'objet d'intérêt (la lésion) dans l'image, en introduisant un simple seuillage. Tous les pixels d'intensité supérieur ou égal au seuil sont converti en blanc, les pixels d'intensité inférieur sont converti en noir. Le seuil est suggéré par l'utilisateur. Le retenu est une image binaire, avec un objet blanc sur un fonds noir.

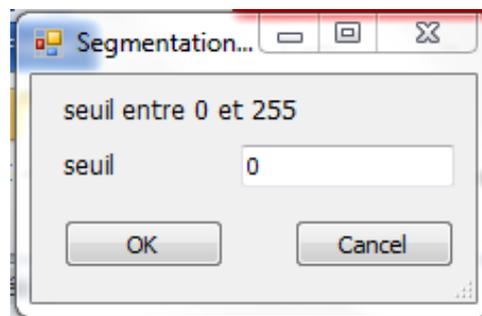


Figure III-4. fenêtre de segmentation « SegmentationForm »

- « invert » ou inversion

C'est une fonction qui permet l'inversion des pixels de l'image.

Elle est réalisée par la classe « invert » de l'espace AForge.Imaging.Filters.

Cette opération ne fonctionne que sur les images binaires.

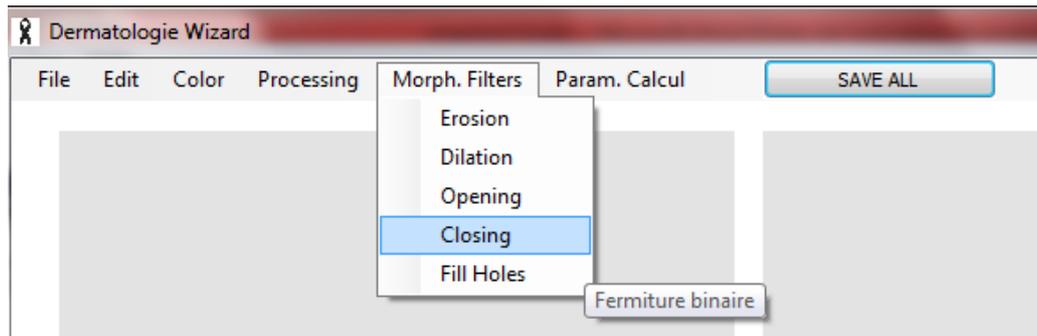
- « edge » ou détection de contour

Cette fonction est fait pour la détection du contour de la tumeur. Pour cela nous avons utilisé la classe « edges » de l'espace AForge.Imaging.Filters, qui est à la base une simple convolution de l'image avec le masque suivant :

$$\begin{array}{ccc} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{array}$$

## 2.5. La fenêtre « morphological filters » : les filtres morphologiques

Cet onglet regroupe un ensemble des opérateurs morphologiques dédiés à l'amélioration du masque de l'image.

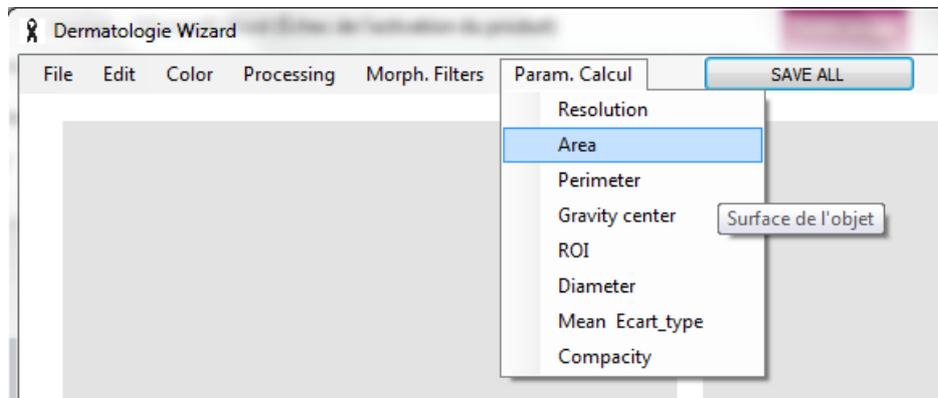


- « érosion » : érosion morphologique  
 Cette fonction permet d'appliquer une érosion morphologique qui sert à éliminer des points artéfacts. Ce filtre se trouve sous le nom « érosion » sous l'espace AForge.Imaging.Filter, il sert à enlever les Pixels, qui ne sont pas entourés par la quantité indiquée de voisins. Il donne la capacité d'enlever les Pixels bruyants (Pixels autonomes) ou de rétrécir des objets.
- « dilatation » : dilatation morphologique  
 Permet d'appliquer une dilatation morphologique sur le masque de l'image, pour lisser l'intérieur de l'objet. Elle se trouve dans Aforge sous le nom « dilatation »
- « Opening » : ouverture morphologique  
 Une ouverture morphologique est une érosion suivie par une dilatation. Ce filtre permet d'éliminer les petits objets. Il prend le nom « opening » sous aforge.
- « Closing » : fermeture morphologique  
 Une fermeture morphologique est une dilatation suivie par une érosion. Ce filtre est utilisé pour connecter ou remplir les objets. Il prend le nom « closing » sous aforge.
- « fill holes » : remplissage de trous  
 Ce filtre permet de remplir les trous dans une image binaire. Il est possible de changer la taille des trous à remplir, dans ce programme nous avons pris une longueur maximale de 50 pixel, et une largeur de 50 pixels.  
 Il se trouve dans aforge sous le nom « FillHoles ».

## 2.6. La fenêtre « Parameter Calcul » : calcul des paramètres

Cet onglet est le plus important dans notre travail : calculer les paramètres utiles pour les dermatologues. Ces paramètres sont le cœur de notre système d'aide au diagnostic. Il faut

calculer l'image binaire avant d'utiliser ces fonctions. Ces images doivent avoir un objet blanc sur un fonds noir.



- « resolution » : la résolution

Lorsqu'on fait des calculs les résultats sont en pixels. Mais pour un médecin c'est difficile à interpréter, il est plus intéressant d'avoir ces mesure en mm. Pour faire cette conversion il faut connaître la résolution de l'appareil utilisé.

La résolution est prise par défaut 0.047.

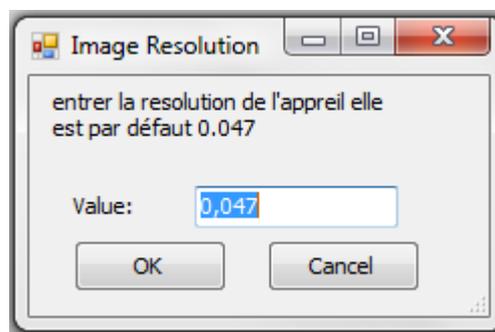


Figure III-5. Fenêtre de la résolution d'image

- « area » : surface

Cette fonction permet de calculer la surface de l'objet en pixel. Elle utilise un algorithme simple qui compte les pixels blancs du masque binaire.

#### Algorithme

##### **Début**

- Initialiser la surface à zéro
- Initialiser une variable couleur c

- Pour tout pixel de l'image faire
  - Tester la couleur du pixel
  - Incrémenter la surface si pixel est blanc

**Fin**

- « perimeter » : périmètre

Le périmètre est le nombre de pixels appartenant au contour.. Il fait compter les pixels blancs de l'image de contour.

#### Algorithme

**Début**

- Initialiser le périmètre à zéro
- Initialiser une variable couleur c
- Pour tout pixel de l'image faire
  - Tester la couleur du pixel
  - Incrémenter le périmètre si pixel est blanc

**Fin**

C'est le même algorithme utilisé pour calculer la surface, mais il est appliqué sur l'image du contour.

- « Gravity Center » : centre de gravité

Cette fonction comporte deux taches : calculer le centre de gravité, et ploter le sur l'image.

Pour calculer le centre de gravité, nous avons utilisé la classe « BlobCounter » de l'espace AForge.Imaging.

BlobCounter est une classe qui s'intéresse par les objets de l'image, et leurs caractéristiques, il peut compter le nombre d'objets qui existent et les extraire.

« blobCounter.GetObjectsInformation » est une méthode qui rend un vecteur des propriétés. On s'intéresse par le centre de gravité :

```

BlobCounter blobCounter = new BlobCounter();
blobCounter.ProcessImage(image_BW);
Blob[] blobs = blobCounter.GetObjectsInformation();
AForge.Point Center = new AForge.Point();
if (blobs.Length > 0)
{
    Center.X = blobs.Average(c => c.CenterOfGravity.X);

```

```

        Center.Y = blobs.Average(c => c.CenterOfGravity.Y);
    }

```

Nous avons calculé les coordonnées du centre de gravité, maintenant pour le ploter, nous avons utilisé la classe « Graphics » offerte par l'environnement .NET. Cette classe encapsule une surface de dessin GDI+, et permet de dessiner des formes géométriques sur l'image. Puisque dessiner un seul pixel n'est pas vraiment donné, nous avons choisi de dessiner un rectangle de centre (X,Y) et de longueur et largeur de 1pixel

```

Graphics g = Graphics.FromImage(pictureBox1.Image);
g.FillRectangle(Brushes.Red, new Rectangle(X, Y, 1, 1));
pictureBox2.Refresh();

```

- « Diameter » : diamètre

Le diamètre est l'axe le plus long de la lésion, Il est l'un des critères de la règle ABCD (D : diamètre inférieur à 6 mm). Le diamètre est une caractéristique de différenciation des tumeurs malignes et des lésions bénignes.

Dans ce programme nous avons simplifié un peu, et nous avons calculé un diamètre « horizontal » et un diamètre « vertical » et on prend la valeur maximale des deux.

#### Algorithme diamètre horizontal

##### **Début**

- initialiser une variable couleur c
- on fixe l'abscisse i=X (abscisse du centre de gravité)
- pour tous les pixels de coordonnées (i=X, j=1 : longueur d'image)

Compter les pixels blancs

##### **Fin**

#### Algorithme diamètre vertical

##### **Début**

- on fixe l'ordonnée j=Y (ordonnée du centre de gravité)
- pour tous les pixels de coordonnées (i=1 : largeur d'image, j=Y)

Compter les pixels blancs

##### **Fin**

- « mean » : moyenne des composantes couleurs

La moyenne est une des paramètres de texture, elle mesure l'homogénéité de la couleur dans l'image. Elle est définie par :

$$moy = \frac{1}{m * n} * \sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i,j) \quad \text{III-1}$$

Algorithme**Début**

- Pour tous les pixels de l'image faire
- Appliquer l'équation III-1

**Fin**

- « Ecart-type »

C'est aussi un paramètre de texture. Il mesure la dispersion des intensités des composantes couleurs. Elle est définie par :

$$Ecart = \frac{1}{m * n} \sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} (p(i,j) - Moy)^2 \quad \text{III-2}$$

Algorithme**Début**

- Pour tous les pixels de l'image faire
- Appliquer l'équation III-2

**Fin**

- « ROI » : région d'intérêt

Cette fonction fait apparaître la tumeur en niveaux de gris sur un fonds noir. L'approche la plus simple est de faire une multiplication entre l'image originale et son masque, mais le C Sharp ne permet pas une multiplication entre deux variables couleur, donc on a développé un algorithme différent.

Algorithme**Début**

- Initialiser une variable couleur c1 pour l'image couleur
- Initialiser une variable couleur c2 pour l'image binaire
- Pour tous les pixels de l'image binaire

Si  $c2 = 255$  (couleur blanche)

Remplacer le pixel par  $c1$

**Fin**

- « compactness » : compacité

La compacité est un critère très important de suspicion de la malignité. L'index d'irrégularité est défini par ;

$$C = \frac{p^2}{4\pi a} \quad \text{III-3}$$

Avec **a** : aire de la tumeur, **p** : son périmètre.

Algorithme

**Début**

- Pour tous les pixels de l'image faire
- implémenter l'équation III-3

**Fin**

## 2.7. Le bouton « Save all »

La possibilité d'enregistrer les paramètres calculés est le plus important dans ce travail. Pour un médecin l'archivage des résultats est une nécessité, puisque il permet le suivi de la progression du patient, il permet d'évaluer un paramètre aussi important pour le diagnostic qui est E (Evolution).

Dans ce travail, nous avons créé une nouvelle forme qui demande à l'utilisateur d'entrer le numéro du patient, son nom et son âge. Ces informations seront enregistrées dans le même fichier avec les paramètres que nous avons calculé.

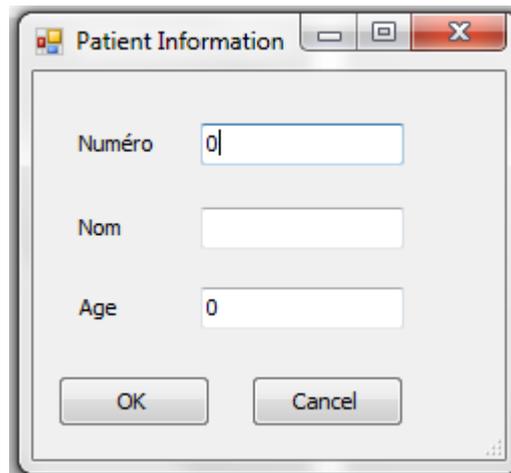


Figure III-6. La fenêtre « information du patient »

Ce bouton sert à enregistrer tous les paramètres calculés, accompagnés par le numéro du patient, son nom, son âge, et la date correspondante au jour du traitement. Toutes informations sont insérées par le médecin, sauf la date qui est automatiquement enregistrée selon la date du jour.

### 3. Gestion des erreurs

Pour éviter l'arrêt du programme lors de l'exécution, il faut connaître les erreurs qui peuvent se générer et les gérer d'avance.

Les erreurs les plus communes sont les problèmes de compatibilité.

Le premier cas est lorsqu'on applique sur un type d'image, alors qu'on doit appliquer sur un autre.

Exemple : la segmentation se fait sur une image en niveau de gris, si on l'applique sur une image couleur, une erreur se produit. Il faut alors tester le type de l'image avant d'appliquer le traitement.

Algorithme :

**Début**

```
Si image_NG = null
Afficher message erreur !
Sinon
Appliquer ( algorithme )
Fin si
```

**Fin**

Un deuxième cas : si le seuil choisi dépasse l'intervalle autorisé.

Exemple : la segmentation autorise une plage des seuils entre [0 - 255], si on dépasse cet intervalle, le système ne fonctionne pas.

Algorithme :

**Début**

Si seuil<0 ou seuil>255  
Afficher message erreur !  
Sinon  
Appliquer (algorithme)  
Fin si

**Fin**

#### 4. Les bulles d'info

L'interface est en Anglais, nous avons choisi l'anglais puisque elle est plus utilisée dans les logiciels et les interfaces qui existent. Mais pour s'approcher aux utilisateurs non experts ou qui ne maîtrisent pas la langue, nous avons ajouté des bulles d'informations qui génèrent une petite description du bouton à chaque fois qu'on passe par le curseur.

Cette fonction est réalisée par une propriété appelée « ToolTip », comme suit :

- Ajouter un événement « tooltip » de la boîte d'outil
- Pour chaque bouton >> propriété >> ToolTip
- On définit le texte qu'on veut voir

#### III.4. Résultats et simulation

L'interface que nous avons développée est utilisée en dermatologie. Nous allons étudier son efficacité en comparant les résultats obtenus pour deux types d'images : une présentant un cas normal (tumeur bénin) et deux images présentant deux cas pathologiques (mélanome).

Tableau III-1. Comparaison des résultats

	Cas sein	Cas pathologique 1	Cas pathologique 2
Image originale			
Surface (mm <sup>2</sup> )	26,305	62,528	171,248
Périmètre (mm)	18,753	36,566	66,505
Diamètre (mm)	5,499	8,084	12,878
Moyenne	>77	>43	>70
Ecart_type	>27000	>12000	>30000
Compacité	1,064	1,702	2,055

### Discussion des résultats :

#### 1. Surface :

Pour un cas normal, la taille de la lésion est petite, dans notre cas 26 mm<sup>2</sup>. Pour un mélanome, la surface de la lésion est considérablement large.

#### 2. Périmètre :

La même chose comme la surface, le périmètre d'une lésion maligne (mélanome) est plus élevé que celui d'une lésion normale.

#### 3. Diamètre :

C'est un paramètre très pertinent, par convention un diamètre supérieur à 6 mm propose une grande probabilité de malignité de la lésion.

Le cas normal présente un diamètre de 5.499 mm qui est considéré normal.

Le deuxième cas présente un diamètre de 8.084 qui propose que c'est cas malin

Le troisième cas présente un diamètre très élevé de 12,878 mm.

Selon les diamètres mesurés et la règle utilisée, le premier cas est classé comme un cas normal et les deux autres cas proposent une grande probabilité de malignité .

#### 4. Compacité : c'est un paramètre qui traduit la régularité du contour de la lésion. Pour des cas normaux, sa valeur s'approche de 1 et s'éloigne pour les cas malins.

Dans notre exemple, le premier cas présente une compacité égale à 1,064 qui indique un contour régulier.

Le deuxième cas avec une compacité égale à 1,702 et le troisième cas avec une compacité égale à 2,055, sont considérés comme des contours irréguliers, qui indique une probabilité de malignité des lésions.

### III.5. Exécutable, installable

Le problème qui se pose pour la majorité des systèmes d'aide au diagnostic qui existent est la possibilité de les installer et les exécuter sur n'importe quel système Windows. C'est bien l'objet de cette partie.

Le C Sharp offre une méthode pour créer une version installable du programme développé.

- Dans la fenêtre projet nous allons au bouton « publier ».

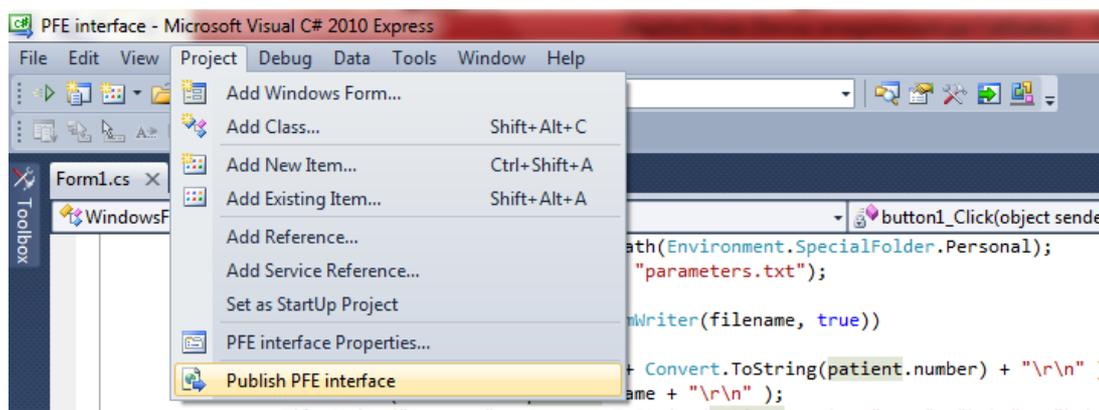


Figure III-7. Créer le fichier exécutable, ETAPE 1

- Une fenêtre de publication s'apparait et nous proposons le chemin désiré pour publier notre application

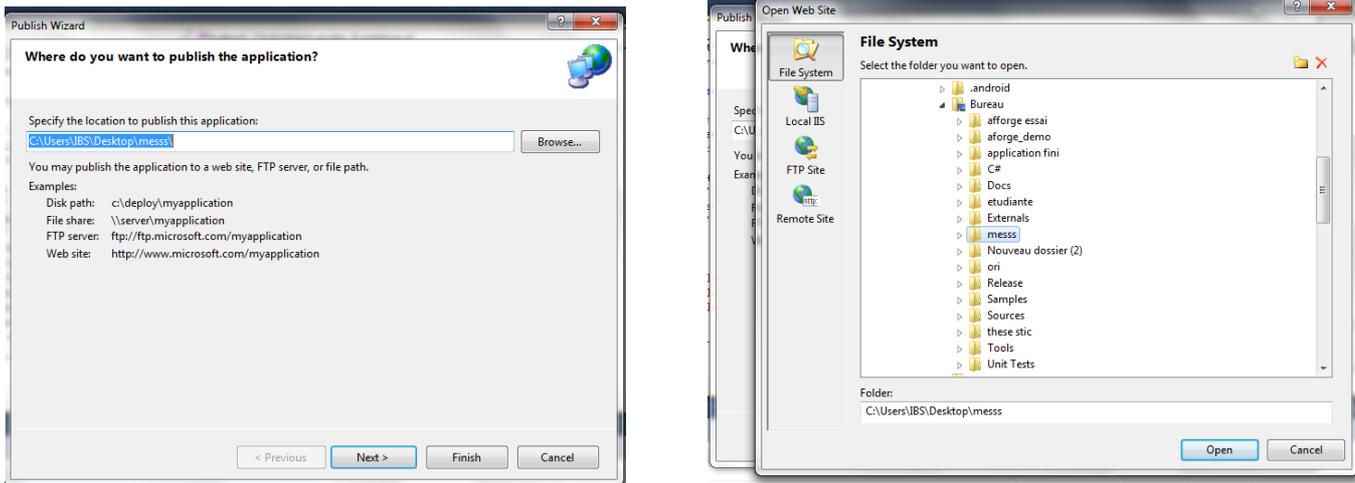


Figure III-8. Créer le fichier exécutable, ETAPE 2 (choisir l'emplacement)

- Comment peut l'utilisateur installer l'application ? il nous propose 3 possibilité : soit depuis un URL (site Web), ou bien un chemin UNC, ou d'après un CD/DVD-ROM. Pour créer un installable \*.exe nous avons utilisé le dernier choix.

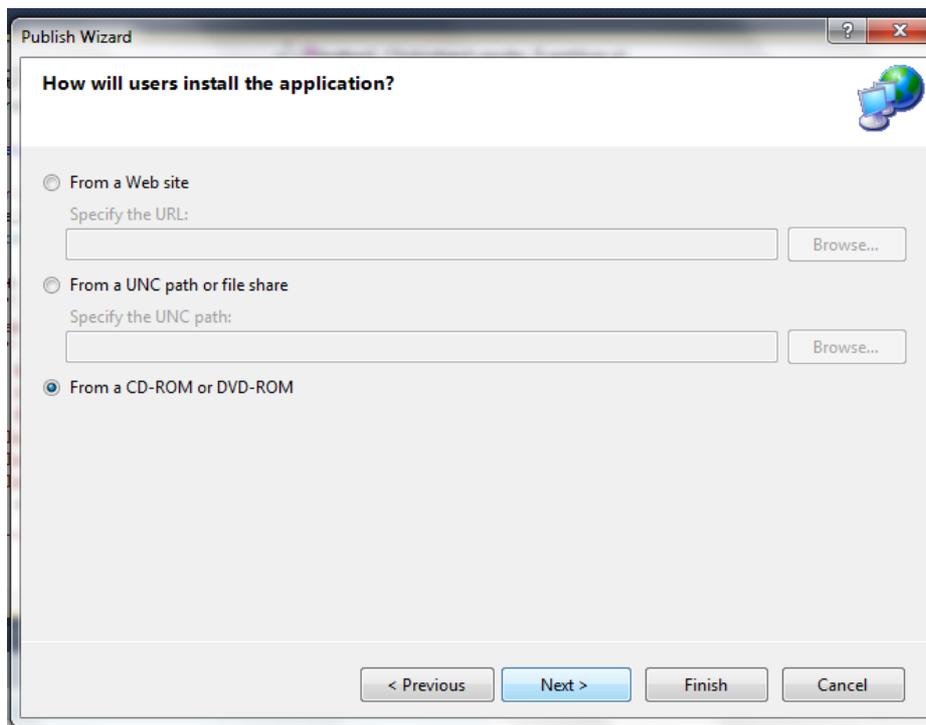


Figure III-9. Créer le fichier exécutable, ETAPE 3 (type de fichier)

- L'information médicale n'existe que dans un cadre interprétatif qui doit être constamment mis à jour pour éviter des erreurs diagnostiques ou thérapeutiques. Dans l'espace Csharp .Net, nous avons la possibilité d'introduire un emplacement qui sera proposé pour installer des mises à jour à chaque fois qu'on exécute le programme.

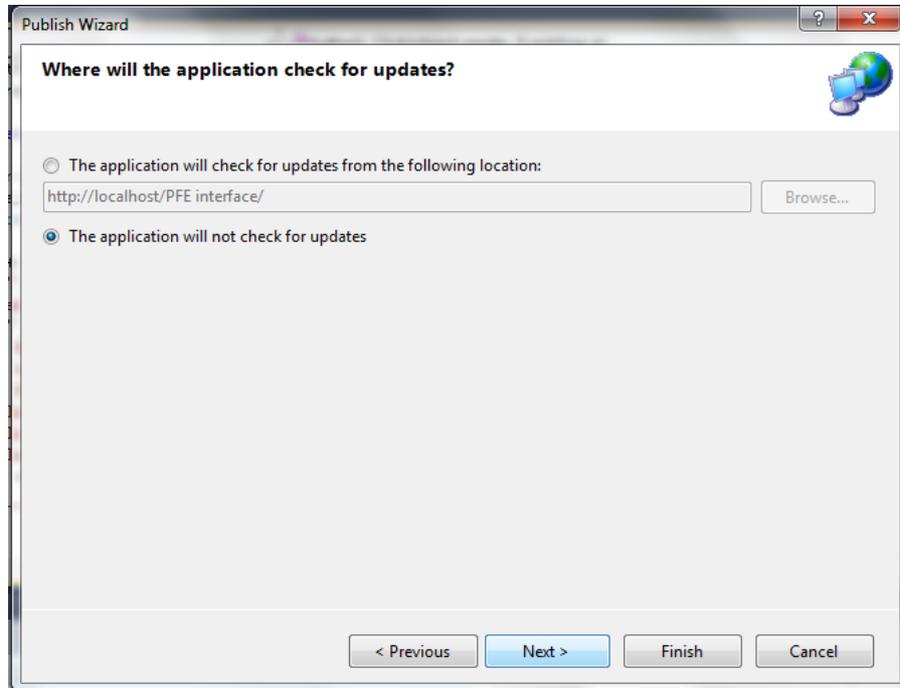


Figure III-10. Créer le fichier exécutable, ETAPE 4 (option mise à jour)

- Notre programme est maintenant prêt pour être publié.

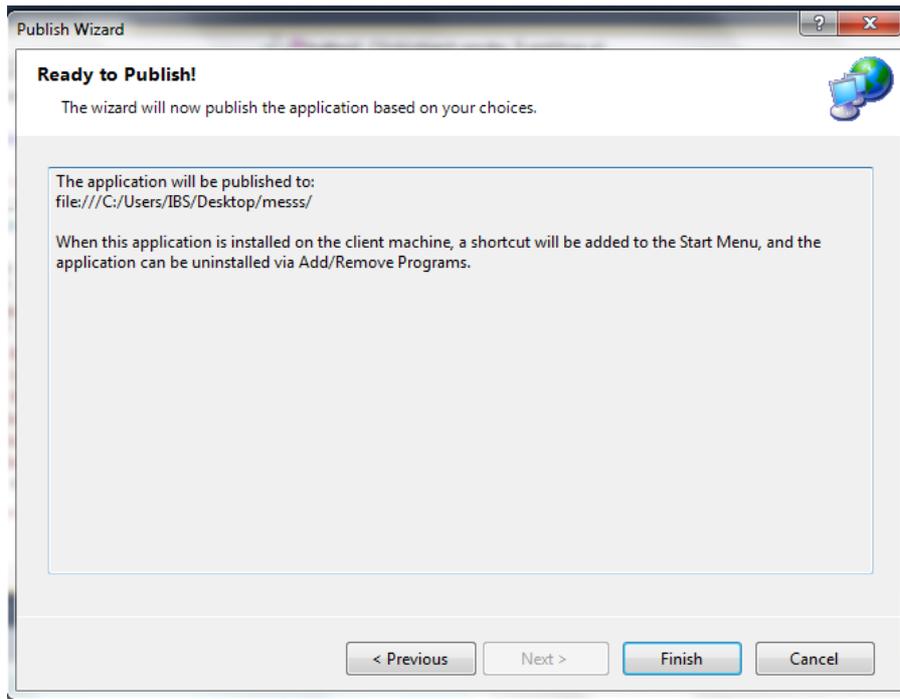


Figure III-11. Créer le fichier exécutable, ETAPE 5 (fin)

- Dans le répertoire que nous avons choisi (figure 6), se trouve :

PFE interface : l'application sous la forme d'un exécutable, c'est-à-dire on peut l'exécuter sans être obligé de l'installer sur l'ordinateur.

Setup : l'application sous la forme d'un installable, c'est-à-dire elle va créer un répertoire sur l'ordinateur, et un raccourci dans le menu démarrer.

Application Files : répertoire contenant tous les bibliothèques nécessaire pour le bon fonctionnement de l'application.

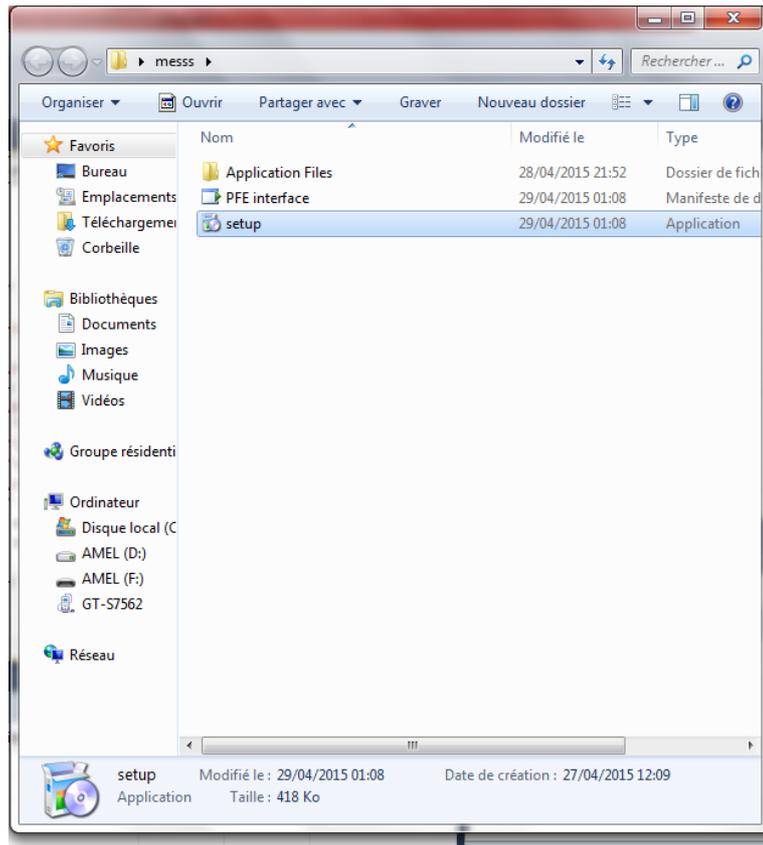


Figure III-12. Le répertoire contenant l'application finale

- On installe l'application, on suit les instructions indiquées par le guide d'installation.

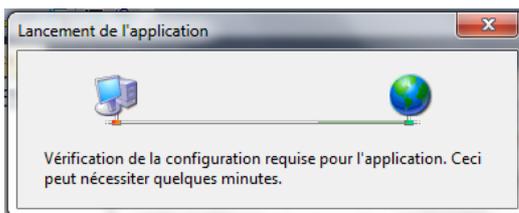


Figure III-13. Installation de l'application

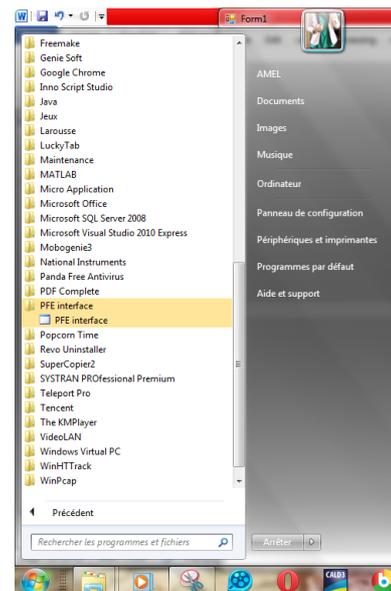


Figure III-14. Apparition de l'application dans le menu démarrer

- Le produit final se présente dans la figure suivante :

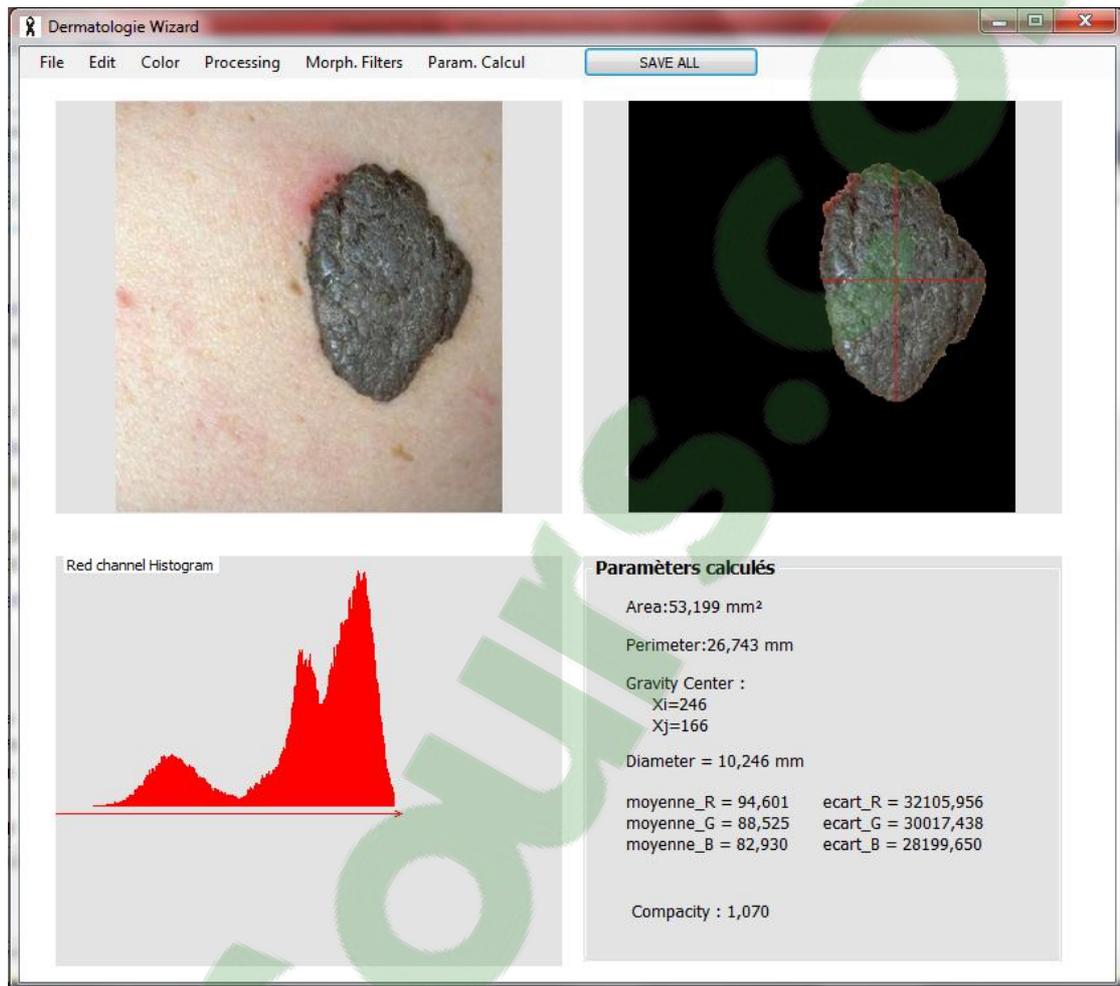
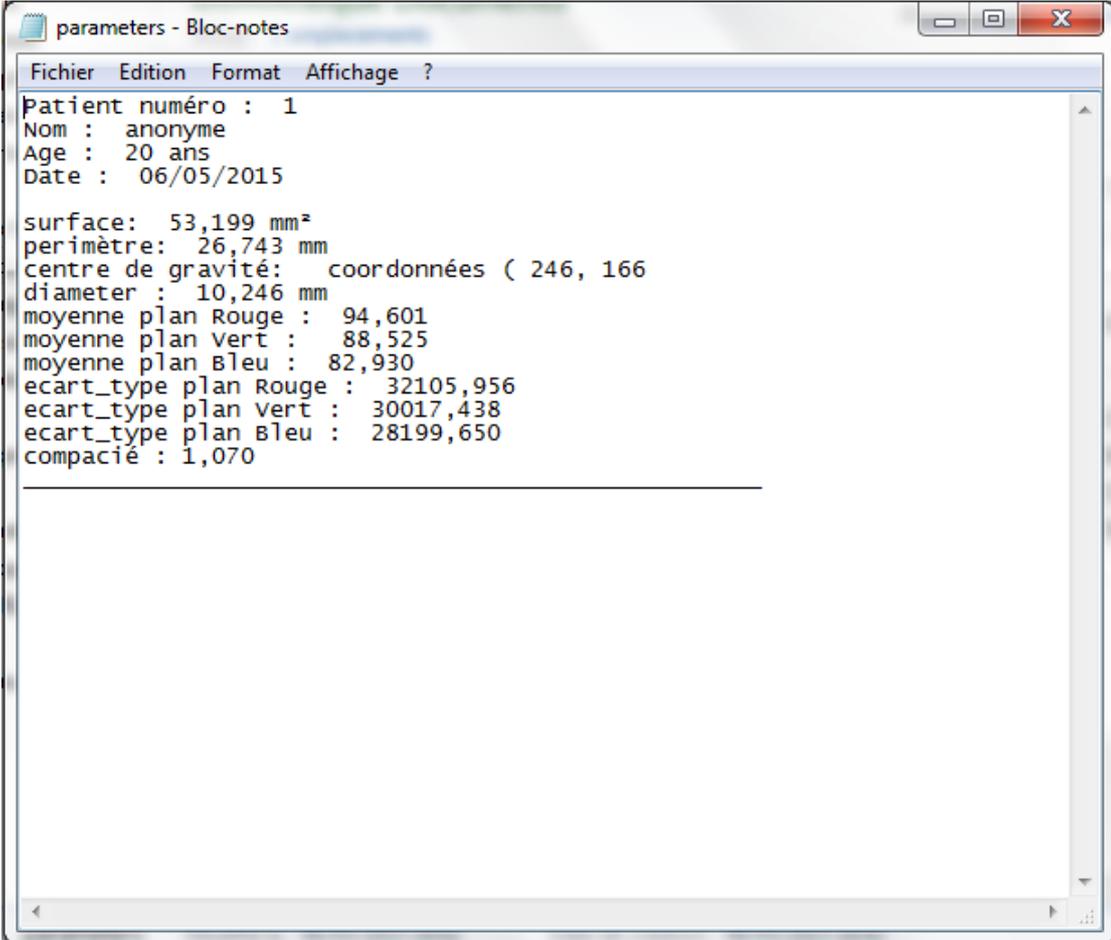


Figure III-15 : Interface graphique finale

- Les paramètres sauvegardés dans un fichier \*.txt se présente dans la figure suivante :



```
parameters - Bloc-notes
Fichier Edition Format Affichage ?
Patient numéro : 1
Nom : anonyme
Age : 20 ans
Date : 06/05/2015

surface: 53,199 mm²
perimètre: 26,743 mm
centre de gravité: coordonnées ( 246, 166
diameter : 10,246 mm
moyenne plan Rouge : 94,601
moyenne plan vert : 88,525
moyenne plan Bleu : 82,930
ecart_type plan Rouge : 32105,956
ecart_type plan vert : 30017,438
ecart_type plan Bleu : 28199,650
compacité : 1,070
```

Figure III-16. Le fichier « parameters.txt » contenant les paramètres calculés

### III.6. Conclusion

Dans ce chapitre, nous avons présenté l'interface que nous avons réalisée. Notre but était de développer une application d'aide au diagnostic en dermatologie. Il est très important qu'elle soit efficace, et en même temps claire et facile à utiliser.

Le plus important, qu'elle soit installable sur n'importe quelle système Windows. Cette caractéristique est assurée par l'environnement .NET.

Finalement, le résultat obtenu est très satisfaisant et l'interface réalisée répond aux besoins exigés.

## CONCLUSION GENERALE

Dans ce travail, nous avons démontré la possibilité de développer un système d'aide à la décision médicale qui est à la fois installable et exécutable sur n'importe quel système Windows.

Nous avons pris comme un titre d'exemple le cas du mélanome en dermatologie, mais cette solution peut s'appliquer sur n'importe quel problème, il suffit de se familiariser avec l'environnement C#.NET pour traduire les algorithmes nécessaires.

Le résultat obtenu est très satisfaisant, il réponds à la problématique que nous avons posé, mais il peut être amélioré en ajoutant un système de classification pour générer une décision finale pour le médecin, en même temps, on peut le relier à une base de données pour importer les images et exporter les paramètres calculés.

Comme autre perspective, on peut créer une version exécutable sur le système Android (\*.apk) pour utiliser l'application sur un Smartphone.

## BIBLIOGRAPHIE

- [1] «Noesis,» [En ligne]. Available: <http://www.tnpc.fr/>. [Accès le 03 Avril 2015].
- [2] «ADCIS,» [En ligne]. Available: <http://www.adcis.net/fr/Produits/Suite-Logicielle-Aphelion-Traitement-Et-Analyse-D%27Image.html>. [Accès le 15 Avril 2015].
- [3] U. o. L. H. a. C. Dr. Michael D. Abràmoff, U. o. P. Dr. Paulo J.Magalhaes et L. S. U. H. S. C. and Dr. Sunanda J. Ram, «image processing with ImageJ,» *BIOPHOTONICS INTERNATIONAL*, Juillet 2004.
- [4] [En ligne]. Available: <http://imagej.nih.gov/ij/docs/intro.html>. [Accès le 15 Avril 2015].
- [5] «OsiriX Imaging Software,» [En ligne]. Available: <http://www.osirix-viewer.com/AboutOsiriX.html>. [Accès le 02 Avril 2015].
- [6] «mathworks,» 2014. [En ligne]. Available: <http://fr.mathworks.com/products/datasheets/pdf/image-processing-toolbox.pdf>. [Accès le 15 Avril 2015].
- [7] «scilab,» 2015. [En ligne]. Available: <http://www.scilab.org/scilab/about>. [Accès le 15 04 2015].
- [8] [En ligne]. Available: <http://siptoolbox.sourceforge.net/>. [Accès le 18 Mai 2015].
- [9] NI Vision for LabVIEW TM User Manual, National Instruments, Novembre 2005.
- [10] V. L. Goff, Apprenez à programmer en python, 2011.
- [11] M. E. Fredrik Lundh, 12 Mars 2002. [En ligne]. Available: <http://www.pythonware.com/media/data/pil-handbook.pdf>. [Accès le 18 Mai 2015].
- [12] S. B. H. Sollie, Automatic segmentation and registration of CT and US images of abdominal aortic aneurysm using ITK, 2002.
- [13] S. F. University, «MATITK,» 2007. [En ligne]. Available: <http://matitk.cs.sfu.ca/>. [Accès le 18 Mai 2015].
- [14] E. Gregori, «Introduction to Computer Vision using OpenCV,» 2012.
- [15] P. Viola et M. J. Jones, Robust Real-time Object Detection, 2001.
- [16] D. i. C. L. L. Hendriks, p. d. i. L. J. v. Vliet, d. d. p. B. Rieger, d. i. M. v. Ginkel et i. R. Ligteringen, «DIPimage User Manual,» 30 Octobre 2014. [En ligne]. Available: [ftp://qiftp.tudelft.nl/DIPimage/latest/docs/dipimage\\_user\\_manual.pdf](ftp://qiftp.tudelft.nl/DIPimage/latest/docs/dipimage_user_manual.pdf). [Accès le 18 Mai 2015].
- [17] «aforgenet,» [En ligne]. Available: <http://www.aforgenet.com/framework/features/>. [Accès le

01 Mai 2015].

- [18] M. D. k. COULIBALY, «Aspects épidémiocliniques et histopathologiques du mélanome malin dans le service de dermatologie,» 2008.
- [19] «dermato info,» 1 octobre 2014. [En ligne]. Available: [http://dermatoinfo.fr/article/Cancers\\_de\\_la\\_peau\\_les\\_melanomes](http://dermatoinfo.fr/article/Cancers_de_la_peau_les_melanomes). [Accès le 15 05 2015].
- [20] 04 Aout 2013. [En ligne]. Available: <http://www.dermatonet.com/fiche-dermatonet-melanome.htm>. [Accès le 15 Mai 2015].
- [21] B. A. G. F. e. a. Joseph E, Results of complete lymph node dissection in 83 melanoma, 1998.
- [22] C. R. a. M. Cook, «Accuracy in clinically evaluating pigmented lesion,» *British Medical Journal*, n° %1299, pp. 16-18, 1989.
- [23] R. L. a. M. P. Koh H. K., «Screening for Melanoma / Skin Cancer,» *Journal of American Academy of Dermatology*, n° %120(2), pp. 159-172, 1989.
- [24] D. P. C. a. P. A. Kenny, «Fat Segmentation in Magnetic Resonance Images,» chez *BIOLOGICAL AND MEDICAL PHYSICS, BIOMEDICAL ENGINEERING*, Springer, 2011, pp. 98-99.
- [25] G. Matheron et J. Serra, *The Birth of Mathematical Morphology*, CSIRO PUBLISHING, 2002.
- [26] S. MOUINE, PFE. Traitement morphologique des images de feuilles, 2009.
- [27] J. Bakmezjian et A. Mauclerc, *Programmation Mobile avec C# .NET*, DUNOD, 2004.
- [28] N. Hilaire, *Apprenez à Développer en C#*, IMPRIM' VERT, 2012.
- [29] T. GROUSSARD, *C# Développez avec Visual Studio 2008*, eni éditions, 2008.
- [30] «C# : L'essentiel en concentré,» 01 04 2008. [En ligne]. Available: <http://www.dotnet-france.com/Documents/IntroCSharpVBNET/Csharp%20-%20L'essentiel%20en%20concentr%C3%A9.pdf>. [Accès le 02 05 2015].
- [31] A. M. Julien Bakmezjian, *Programmation Mobile avec C# .NET*, DUNOD, 2004.
- [32] «aforge.NET,» [En ligne]. Available: <http://www.aforgenet.com/framework/>. [Accès le 06 05 2015].