



UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED
FACULTÉ DES SCIENCES EXACTES
Département D'Informatique



Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

MASTER ACADEMIQUE

Domaine : **Mathématique et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Distribués et Intelligence Artificielle**

Présenté par :

- **AZZOUZ IMTIAZ**
- **MEDELLEL WAFA**

Thème

Tri automatique des pommes de terre par techniques d'apprentissage automatique

Soutenu le 20-06- 2019 Devant le jury:

M.	KERTIOU ISMAIL	MAA	Président
M.	NEDIOU ABD ELHAMIDE	MAA	Rapporteur
M.	KHEBBACHE MOHIB EDDINE	MAA	Encadreur

Année Universitaire: 2018/2019



Remerciements

Nous tenons à remercier du fond du cœur, avant tous, «ALLAH» le tout puissant de mes avoir donné la force, la volonté et le courage pour réaliser ce travail.

*Nous tenons à remercier à notre encadreur
« Mr. KHEBBACHE MOHIB EDDINE » qui nos a aidés pendant toute la durée de notre travail et pour sa patience et ses précieux conseils et ses compétences.*

Nous ne pouvons de manquer de remercier le président du département de l'informatique le professeur « Mr. MEDILEH SASSI ».

Nous remercions vivement les membres du jury qui nous ont fait l'honneur d'accepter de juger notre travail .

N'oublions pas, bien sûr, nos professeurs de tout le cycle universitaire, de leur présenter nos meilleurs vœux.

Et enfin , nous tiens à remercier tous ceux qui ont contribué, de près ou de loin, par leurs encouragements, leurs conseils et leurs soutiens à mener à bien ce travail.

Dédicaces

Avant tout je remercie Allah le Tout-Puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré tous les difficultés on compterait tout d'abord.

Et je remercie mon encadreur «Mr. KHEBBACHE MOHIB EDDINE» à ses conseils et ses compétences, qui nous a aider beaucoup malgré son charge de travail .

Je dédié ce modeste travail

À la fontaine qui ne se laisse pas de donner à ceux qui ont tissé mon bonheur avec des fils tissé de son cœur , ma chère mère «AMINA» qui m'a aidé grâce à son prière et à son bénédiction.

Et puis j'ai remercié le cœur du compatissant qui était à côté de moi tous les étapes qui sont passé du plaisir et de la souffrance et était une bougie qui allume ma vie et grâce à tout mon crédit pour mon chère père «SADOK» .

A mes chères sœurs «IHSAN et ASMA» et mes chers frères «ANES et AMINE» et aussi son oublié tous la famille et mes amis de près ou de loin surtout mes chers amis «ASMA et NESRINE et Wafa» par leurs encouragements leurs conseils et leurs soutiens à mener à bien ce travail.

IMTIAZ AZZOUZ



Dédicaces

Avant tout, je remercie ALLAH le tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.

Je remercie mon encadreur «Mr. KHEBBACHE MOHIB EDDINE» à ses conseils et ses compétences, qui nous a aidé beaucoup malgré son charge de travail .

Tous d'abord Je dédie ce modeste travail À ma très cher mère qui m'a encouragé par tous les moyens à avoir la confiance et l'espoir au cours de la rédaction de ce travail.

Adorée et qui m'a aidé, grâce à leur prière et à leur bénédiction.

À mon père .

Je le dédie aussi À ma très cher sœurs .

A mes très chers frères .

Et à toute la famille. À mes meilleurs amies «Nesrine Imtiaz Asma».

À tous les étudiants de la faculté Informatique surtout les étudiants de la 2ème année master promotion 2019 .



Résumé

Le tri des produits agricoles est un enjeu dans l'agriculture afin d'assurer un produit de qualité. Aujourd'hui, les systèmes de tri automatique basés sur la vision par ordinateur et l'apprentissage machine se reposent sur des caractéristiques visuelles et sur des méthodes qui permettent de faire apprendre à une machine « un modèle » permettant de réaliser une classification ou de distinguer les différents qualités de produits agricoles. Les pommes de terre représentent pour la wilaya d'El oued, une partie importante de l'économie national et internationale. Dans ce mémoire, nous avons intéressé au processus de tri automatique des pommes de terre. Particulièrement, nous avons développé un système capable de reconnaître la catégorie de la pomme de terre selon des caractéristiques de texture, de couleur et ainsi leur fusion. En effet, un classifieur multi classes supervisé basé sur l'utilisation de l'algorithme d'apprentissage machine les SVM (machines à vecteur de support) est construit. En plus, nous avons constitué des données d'entraînement et de test (notre propre dataset) à partir des images acquises des pommes de terre de quatre classes à savoir : bartina, bartina avec maladie, Spunta et Spunta avec maladie. Les résultats obtenus nous donnent un système performant dont le taux de reconnaissance (accuracy) est $> 70\%$.

Mots Clés : apprentissage automatique, vision par ordinateur, systèmes de tri automatique, pommes de terre, extraction de caractéristiques, Classification, Machine à vecteurs de support, LBP, HOG , accuracy.

Abstract

The sorting of agricultural products is one of the challenges of agriculture to ensure a products of quality. Nowadays, automatic sorting systems based on computer vision and machine learning rely on visual characteristics and on methods that make it possible for a machine to learn a "model" and to classify or distinguish between the different qualities and/or status of agricultural products. The potatoes represent for El Oued region, an important part of the national and international economy. In this dissertation, we have been interested in the process of automatic sorting of potatoes. In particular, we have developed a system capable of recognizing the category of the potato according to characteristics of texture, color and their fusion. Indeed a supervised multi-class classifier based on the use of the machine learning algorithm SVM (Support Vector Machines) is built. In addition, we have also built up training and test data (our own dataset) from images acquired from potatoes of four classes namely: bartina, bartina with disease, Spunta and Spunta with disease. The results obtained give us a powerful system with accuracy is greater than 70%.

Keywords : Machine Learning, Computer Vision, Automatic Sorting Systems, Potatoes, Feature Extraction, Classification, Local Binary Pattern, Support Vector Machine, HOG , accuracy

الملخص

تشكل عملية فرز المنتجات الزراعية مصدر قلق رئيسي وتحدي يواجه الزراعة بغية ضمان منتوج ذات جودة. في أيامنا هاته ، تعتمد أنظمة الفرز الأوتوماتيكية القائمة على رؤية عن طريق الكمبيوتر والتعلم الآلي على الخصائص المرئية وعلى الأساليب التي تمكن الآلة من تعلم "النموذج" الذي يجعل من الممكن التصنيف أو التمييز بين الصفات المختلفة للمنتجات الزراعية. تمثل البطاطا لولاية الوادي جزء مهم من الاقتصاد الوطني و الدولي. في هذه المذكرة ، اهتمامنا بعملية الفرز التلقائي للبطاطا. على وجه الخصوص ، قمنا بتطوير نظام قادر على التعرف على فئات البطاطا وفقاً لخصائص الملمس واللون وكذا دمجهما. في الواقع ، تم تصميم مصنف متعدد الفئات خاضع للإشراف استناداً إلى استخدام خوارزمية التعلم الآلي SVM لبناء نموذج المصنف الخاص بنا. إضافة الى ذلك، قمنا بتكوين بيانات التدريب والاختبار (مجموعة البيانات الخاصة بنا) من الصور التي تم الحصول عليها من البطاطا المكونة من أربعة فئات وهي: برتينا ، برتينا مع المرض ، سبونتنا و سبونتنا مع المرض. النتائج التي تم الحصول عليها أعطتنا نظام قوي معدل دقته > 70 %.

الكلمات المفتاحية: التعلم الآلي ، رؤية الحاسوبية ، أنظمة الفرز الآلي ، البطاطس ، استخراج المعالم ، التصنيف ، الواصف LBP، الواصف HOG ، الدقة.

Table de matières

Liste des figures

Liste des tableaux

Liste des abréviations

Introduction générale1

Chapitre 1: Système de Tri automatique et machine Learning

1.Introduction4

2.Systèmes de Tri4

2.1.Définition du système de Tri4

2.2.Exemples du système de Tri5

2.2.1. Système de tri pour les pommes de terre 5

2.2.2. Système de tri automatique pour les dates 6

2.3.Critères de tri7

2.3.1. Critères basés sur vision par humain 7

2.3.2. Critères basés sur vision par ordinateur..... 7

2.3.2.1. Couleur 8

2.3.2.2. Texture 8

2.3.2.3. Taille..... 9

2.3.2.4. Forme..... 9

2.4.Architecture d'un système de tri automatique 10

2.4.1. La composition d'un système de tri10

2.4.2. Caractéristiques de la pomme de terre.....11

3.Reconnaissance de formes (pattern recognition)13

3.1.Qu'est-ce qu'une reconnaissance de formes 13

3.2.Reconnaissance statistique de formes..... 14

3.3.Que est ce que un Machine Learning 14

3.4.Types d'apprentissage 17

3.4.1. Apprentissage supervisé17

3.4.2. Apprentissage non supervisé17

3.4.3. Apprentissage semi-supervisé.....18

3.4.4. Apprentissage par renforcement.....18

4.Extraction de caractéristiques18

4.1.Caractéristiques locales 19

4.2.Caractéristiques globales..... 19

4.2.1. Caractéristiques basées sur la couleur19

4.2.2. Caractéristiques basées sur texture.....	19
4.2.3. Caractéristiques basées sur la forme.....	20
4.3.Fusion de caractéristiques	20
5.Classification.....	21
5.1.Classification binaire.....	21
5.2.Classification multi-classes.....	21
5.2.1. Approche un-contre-tout (One-Against-All).....	21
5.2.2. Approche un contre un (One-Against-One)	22
5.3.Algorithmes de classification	22
5.3.1. K plus proches voisins (KNN).....	22
5.3.2. Les machines à vecteurs de support (SVM)	23
6.Conclusion	24

Chapitre 2: Conception du système

1.Introduction	26
2.Conception du Système de tri de pomme de terre proposé	26
2.1 Description du notre système de tri	26
2.2 Architecture du système	27
2.3Conception détaillée du système	29
2.3.1 Module d'acquisition d'images	29
2.3.2 Module de prétraitement.....	29
2.3.3 Module d'extraction des caractéristiques	29
2.3.3.1 Motif Binaire Local (LBP).....	29
2.3.3.2 Histogrammes de Couleurs	30
2.3.3.3 Histogramme des Gradients Orientés (HOG)	33
2.3.4 Module de classification par SVM.....	34
2.3.4.1 Fondement mathématiques	34
2.3.4.2 SVM pour le cas multi-classes	38
3.Conclusion	41

Chapitre 3: Implémentation et évaluation des performances

1.Introduction	43
2.Environnement du développement	43
2.1 Description de langage de programmation « python »	43
2.2 Description de l'environnement de développement « Jupyter »	44
2.3 Description de la bibliothèque « scikit-image ».....	44
2.4Description de la bibliothèque « scikit-learn»	44
3.Évaluer les modèles.....	45

3.1.Mesures (métriques) de performance	46
3.1.1.Taux d'exactitude de prédiction « reconnaissance » (accuracy)	46
3.1.2.Précision	47
3.1.3.Recall : rappel	47
3.1.4.F1score	47
3.2.Validation croisée	47
3.2.1.k-fold cross-validation	48
3.2.2.Leave-one-out cross-validation	48
4.Détails de l'implémentation	48
4.1.Dataset	48
4.2.Extraction de caractéristique.....	49
4.2.1.Extraction de caractéristique de texture	50
4.2.2.Extraction de caractéristique HOG.....	52
4.2.3.Extraction de caractéristique de couleur	53
4.2.4.Combinaison des vecteurs des caractéristiques	55
4.3. Entraînement et test	56
4.4. Evaluation de performance	57
4.4.1.Résultats.....	57
4.4.1.1.Descripteur LBP.....	57
4.4.1.2.Descripteur HOG	64
4.4.1.3.Descripteur histogramme des couleurs	71
4.4.1.4.Fusion de caractéristiques	78
4.5. Discussions	79
4.6. Classification en ligne des images non vue	80
5.Conclusion	81
Conclusion générale	82

Liste des figures

Figure 1 : Système de tri	5
Figure 2 : le machine de tri CALIBREUSE	6
Figure 3 : Le système de vision par ordinateur	8
Figure 4 : Images d'oranges montrant différents couleurs et textures.....	9
Figure 5 : Plusieurs formes d'avocats.....	10
Figure 6 : Exemples sur la catégorie de la pomme de terre « Spunta »	12
Figure 7 : Exemples sur la catégorie de la pomme de terre « Bartina».....	122
Figure 8 : le processus typique du ML.....	155
Figure 9 : Schéma général d'un système de reconnaissance des formes	166
Figure 10 : Schéma descriptif de la fusion des caractéristiques.....	200
Figure 11 : Description du système de tri proposé	277
Figure 12 : L'architecture du système proposé.....	288
Figure 13 : Construction d'un motif binaire et calcul du code LBP	30
Figure 14 :Modèle de couleur HSV	322
Figure 15 : Couleur de HSV obtenir à partir de l'espace colorimétrique RVB.....	322
Figure 16 : Vue d'ensemble du calcul HOG	344
Figure 17 : Description des données dans un espace de plus grande dimension	355
Figure 18 : Hyperplans séparateurs	366
Figure 19 : Exemple de projection	377
Figure 20 : Approche une-contre-une	399
Figure 21 : Approche une-contre-reste.....	40
Figure 22 : Aperçu de fichier « feature_dataLBP.csv ».....	511
Figure 23 : Aperçu de fichier « feature_dataHOG .csv ».....	533
Figure 24 : L'histogramme d'orientation de gradient résultant.....	533
Figure 25 : Aperçu de fichier « feature_dataHCOLOR.csv »	555
Figure 26 : Images après classification en ligne.....	80

Liste des tableaux

Tableau 1 : Matrice de confusion pour la classification

Tableau 2 : Matrice de confusion pour la classification multi-classe

Tableau 4 : mesures de performance pour classification multi classes avec LBP et $C=3$

Tableau 5 : Matrice de confusion pour classification multi classes avec LBP et $C=3$

Tableau 6 Mesures de performance pour classification multi classes avec LBP et $C=0.001$

Tableau 7 : Matrice de confusion pour classification multi classes avec LBP et $C=0.001$

Tableau 8 : Mesures de performance pour classification multi classes avec LBP et $C=0.1$

Tableau 9 : Matrice de confusion pour classification multi classes avec LBP et $C=0.1$

Tableau 10 : Mesures de performance pour classification multi classes avec LBP et $C=1$

Tableau 11 : Matrice de confusion pour classification multi classes avec LBP et $C=1$

Tableau 12 : Mesures de performance pour classification multi classes avec LBP et $C=2$

Tableau 13 : Matrice de confusion pour classification multi classes avec LBP et $C=2$

Tableau 14 : Mesures de performance pour classification multi classes avec LBP et $C=4$

Tableau 15 : Matrice de confusion pour classification multi classes avec LBP et $C=4$

Tableau 16 : Mesures de performance pour classification multi classes avec LBP et $C=5$

Tableau 17 : Matrice de confusion pour classification multi classes avec LBP et $C=5$

Tableau 18 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold ($K=10$) et $C=3$

Tableau 19 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold ($K=10$) et $C=0.001$

Tableau 20 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold ($K=10$) et $C=0.1$

Tableau 21 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=1

Tableau 22 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=2

Tableau 23 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=4

Tableau 24 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=5

Tableau 25 : Mesures de performance pour classification multi classes avec HOG et C=3

Tableau 26 : Matrice de confusion pour classification multi classes avec HOG et C=3

Tableau 27 : Mesures de performance pour classification multi classes avec HOG et C=0.001

Tableau 28 : Matrice de confusion pour classification multi classes avec HOG et C=0.001

Tableau 29 : Mesures de performance pour classification multi classes avec HOG et C=0.1

Tableau 30 : Matrice de confusion pour classification multi classes avec HOG et C=0.1

Tableau 31 : Mesures de performance pour classification multi classes avec HOG et C=1

Tableau 32 : Matrice de confusion pour classification multi classes avec HOG et C=1

Tableau 33 : Mesures de performance pour classification multi classes avec HOG et C=2

Tableau 34 : Matrice de confusion pour classification multi classes avec HOG et C=2

Tableau 35 : Mesures de performance pour classification multi classes avec HOG et C=4

Tableau 36 : Matrice de confusion pour classification multi classes avec HOG et C=4

Tableau 37 : Mesures de performance pour classification multi classes avec HOG et C=5

Tableau 38 : Matrice de confusion pour classification multi classes avec HOG et C=5

Tableau 39 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=3

Tableau 40 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=0.001

Tableau 41 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=0.1

Tableau 42 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=1

Tableau 43 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=2

Tableau 44 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=4

Tableau 45 : multi classes avec HOG , validation croisée K-fold (K=10) et C=5

Tableau 46 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=3

Tableau 47 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=3

Tableau 48 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=0.001

Tableau 49 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=0.001

Tableau 50 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=0.1

Tableau 51 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=0.1

Tableau 52 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=1

Tableau 53 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=1

Tableau 54 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=2

Tableau 55 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=2

Tableau 56 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=4

Tableau 57 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=4

Tableau 58 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=5

Tableau 59 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=5

Tableau 60 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=3

Tableau 61 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=0.001

Tableau 62 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=0.1

Tableau 63 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=1

Tableau 64 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=2

Tableau 65 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=4

Tableau 66 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=5

Tableau 67 : Matrice de confusion pour classification multi classes avec fusion de caractéristiques

Tableau 68 : l'accuracy pour classification multi classes avec fusion de caractéristiques , validation croisée K-fold (K=10)

Liste des abréviations

IA : Intelligence Artificielle.

HOG : Histogramme des Gradients Orientés .

LBP : Motif Binaire Local .

SIFT : Scale-Invariant Feature Transform.

SURF : Speeded Up Robust Features.

HSV : Hue Saturation Value.

RVB : Rouge, Vert, Bleu.

U-C-U : Approche Un Contre Un.

U-C-T : Approche Un-Contre-Tout.

SVM : Support Vector Machine.

RBF :Radial Basis Fonctions.

KNN: k-Nearest Neighbors.

OAA : One-Against-All.

OAo :One-Against-One.

AVA : All-versus-all.

VP : Vrais Positifs.

FP : Faux Positifs.

FN : Faux Négatifs.

VN : Vrais Négatifs.

SVC : Support Vector Classification.

Introduction générale

Introduction générale

Aujourd'hui, l'agriculture en Algérie est en évolution progressive. La région d'El oued a connu un saut spéciale dans la production des produits agricoles surtout le produit de pomme de terre. Cependant, l'Agriculture fait face à de grands enjeux et défis dont les plus importants sont en ce qui concerne la production de la qualité et de façon durable des produits agricoles. En effet, l'Agriculture se construit et se raisonne sur de multiples échelles, avec des objectifs renforcés en termes de compétitivité économique, sociale et de réduction de l'impact environnemental. Dans cette optique, il existe plusieurs problèmes complexes tels ceux qui concerne le tri des produits agricoles et le contrôle de la qualité de ces produits pour distinguer entre les différentes catégories.

Dans ce contexte, l'exploration du potentiel de technologies émergentes en Agriculture, notamment l'IA, permet aujourd'hui d'ouvrir de toutes nouvelles perspectives en matière des systèmes intelligents pour mieux anticiper les tendances et aider les agriculteurs à prendre ses décisions. Cependant, les systèmes de tri existes actuellement, notamment pour les pommes de terre, sont des systèmes manuels, lourds et coûteux. Pour surmonter ces limites-là, des systèmes de tri automatique basés sur la vision par ordinateur sont développés. Par ailleurs, le système de tri automatique des dattes présente des limites en termes de performance qui reste à investir.

La problématique que nous procurons est comment trier des pommes de terre en matière de qualité de façon autonome, rapide et efficace tout en exploitant les propriétés visuelles de ce produit comme : la couleur, la forme et la texture?

Notre objectif est d'investiguer les systèmes informatique de tri automatique des pommes de terre basés sur la vision par ordinateur. En basant sur le processus de l'apprentissage automatique (machine), le système de tri proposé vise à reconnaître automatiquement la pomme de terre et à distinguer les différentes catégories de ce produit agricole. Ceci pour automatiser le processus de tri manuel afin de surmonter certaines souffrances de ceci en termes de temps et de coût. Aussi pour améliorer la qualité du produit de pomme de terre fourni au consommateur et augmenter le rendement des machines de tri.

Pour atteindre cet objectif, nous avons choisi d'utiliser les descripteurs : LBP, HOG et histogrammes de couleurs et leur fusion pour extraire les caractéristiques visuelles de texture et de couleur. Ces caractéristiques sont utilisées pour apprendre par exemple et hors ligne le modèle de décision en utilisant la méthode de classification supervisée « SVM ». Pour cela, nous avons construit notre propre jeu de données (base d'apprentissage) constitué des images de quatre catégories des pommes de terre (Spunta, Bartina, Spunta avec malade, Bartina avec malade) capturées par un appareil photo numérique.

Ce mémoire se compose de trois chapitres :

- Chapitre 1 : dans ce chapitre, nous allons fournir quelques concepts de base pour le système de tri, en commençant par le concept de système de tri, quelques exemples de systèmes existants, ainsi que les critères de tri utilisés tels que le poids et la vision par ordinateur. Aussi en explique également certains fondements théoriques les plus importants de la machine Learning. Après, en finissant par un aperçu de l'extraction des caractéristiques générales et locales ... etc.
- Chapitre 2 : Ce chapitre se concentre sur la description générale et la structure de notre système de tri proposé, qui est basé sur la vision par ordinateur avec une explication détaillée de chaque étape, les méthodes utilisées pour extraire les propriétés et la méthode de classification multi classes appliquée (svm multi-classes).
- Chapitre 3 : Ce dernier chapitre constitue la base de notre sujet d'évaluation de la performance. Tout d'abord nous avons présenté l'environnement de développement utilisé, puis les mesures de performance, et enfin, une discussion de résultats obtenus.

Finalement, nous clôturons le travail faisant l'objet de ce mémoire par une conclusion générale, qui fournit aussi quelques perspectives d'amélioration au futur.

Chapitre 1

Systeme de Tri

automatique et machine

Learning

1. Introduction

Le tri automatique est devenu actuellement un processus fondamental pour garantir la qualité des produits fournis aux consommateurs, où elle était avant fait par le processus de sélection manuelle effectuée par le personnel expérimenté. Dans ce contexte, la détermination de la qualité des produits agricoles est devenue une grande importance dans les dernières années. Pour effectuer cette opération, la conception des systèmes de tri automatique est une nécessité cruciale. Ces systèmes sont basés sur la vision par ordinateur, et en particulier les techniques d'apprentissage automatique pour bien effectuer le processus de tri automatique des produits agricoles.

Dans ce chapitre, nous fournirons, en revue, les bases théoriques des systèmes de tri. Pour cela nous allons passer en détail certains concepts de base comme la reconnaissance de formes et l'apprentissage automatique. Aussi, nous allons jeter d'œil sur certaines caractéristiques utilisées comme critères de tri à savoir : basées sur la vision humaine ou la vision par ordinateur, tout en présentant quelques exemples des systèmes de tri manuel et automatique actuellement réalisés.

2. Systèmes de Tri

Dans cette section on va jeter un coup d'œil à la définition, aux exemples ,aux types et à l'architecture des systèmes de tri.

2.1. Définition du système de Tri

Le système de tri utilisé dans de nombreuses industries pour résoudre l'enjeu de la qualité des produits agricultures tout en exploitant des caractéristiques pertinentes comme le poids et les caractéristiques visuelles. Il nécessite un environnement spécifique pour son bon fonctionnement en termes de : bon éclairage, caméras à haute précision et un système d'aiguillage qui sont également des éléments très important pour un tri de qualité [1].

Le système de tri moderne se reposent sur des machines de tri pour leur fonctionnent. La figure 1 montre un exemple de la machine de tri utilisé au sein de l'usine de tri à Trifaoui, wilaya d'El-oued.



Figure 1 : Système de tri [photo acquise de l'usine]

2.2. Exemples du système de Tri

2.2.1. Système de tri pour les pommes de terre

La pomme de terre est l'une des cultures les plus importantes dans la Wilaya d' EL Oued au cours de ces années. Sa production a été lancée de manière semi-traditionnelle puis s'est rapidement développé en raison de l'intérêt croissant de ce produit sur le marché national et international. Cette production a vu l'émergence de diverses technologies modernes pour le développement de la culture de ce produit [3]. En plus, des petites et moyennes entreprises pour le groupement, le tri, le stockage, et l'exportation ont été créées. L'entreprise de refroidissement méditerranéenne qui située à Trifaoui dans la Wilaya d'EL Oued est l'une des exemples qui nous avons visité. Cette visite nous a permet d'identifier les différentes machines et méthode de tri des pommes de terre. Parmi les dernières machines, cette entreprise utilise la machine de tri CALIBREUSE (voir la figure 2) pour la classification manuelle et rapide de ce produit en fonction généralement de leur qualité, couleur et taille.



Figure 2 : le machine de tri CALIBREUSE [photo acquise de l'usine]

2.2.2. Système de tri automatique pour les dates

Un modèle de travail pour le système de tri et de Classement des fruits de dates est conçu par l'ordinateur, qui comprend deux systèmes : un système de vision par ordinateur et un système de traitement. Le système de vision par ordinateur comprend l'unité de traitement d'image et le module de reconnaissance de formes, qui permettent d'améliorer la qualité du produit dans le processus de tri automatique. Le système de traitement électromécanique comprend des bandes transporteuses, quatre appareils photo numériques pour prendre des photos de tous les côtés et un appareil à rayons X qui capture les rayons X de la vision interne des dattes. Ces photos ainsi que mesure appropriée du poids sont injectés au système de la vision par ordinateur afin d'examiner les fruits visuellement et d'évaluer la qualité et le tri. Le système de vision par ordinateur reçoit les images, les traite, puis extrait les caractéristiques de base telles que la couleur, la longueur et la largeur. Ces informations sont envoyées vers l'outil de reconnaissance de formes pour effectuer des évaluations de la qualité et classer le date dans des catégories prédéfinies après l'application d'un des algorithmes d'apprentissage, puis dirigé vers le conteneur approprié via le système d'affectation [4].

En résumé, la plupart des systèmes de tri moderne utilisent des machines fonctionnant sur la base de plusieurs paramètres de tri. Dans ce qui suit, nous allons présenter les différents paramètres du tri pouvant contribuer au développement de ces systèmes.

2.3. Critères de tri

Les facteurs principaux de l'importation et de l'exportation de produits alimentaires sont la sécurité et la qualité des produits agricoles. Il y'a plusieurs normes pour déterminer la qualité de ces produits. En outre, Il existe plusieurs types de systèmes de tri qui dépendent aux caractéristiques mesurées du produit en question [5]. Nous pouvons donc distinguer deux classes des systèmes de tri : à base de la vision humaine et de la vision par ordinateur nous allons voir comme la suite:

2.3.1. Critères basés sur vision par humain

Le poids dans l'agriculture est un bon indicateur de la qualité des produits agricoles. Par conséquent, il existe des systèmes de tri basés sur le poids comme critère de tri, qui constituent une solution rentable pour le tri manuel ou automatique, fournissant des techniques précises, fiables et cohérentes, tout en accélérant les processus fastidieux et en minimisant les erreurs qui se produisent pendant le processus de tri[5].

2.3.2. Critères basés sur vision par ordinateur

La vision par ordinateur (ou Computer Vision en anglais) est étroitement liée à l'intelligence artificielle. C'est le processus par lequel une machine, généralement un ordinateur, traite automatiquement le contenu des images numériques, pour mener une analyse approprié [2]. En effet, les techniques utilisées en analyse d'image, incluent l'image acquisition, prétraitement des images et interprétation des images, permettant de quantifier et de classer les images et les objets présentant un intérêt dans les images. Dans l'opinion de Timmermans (1998) [6], la vision par ordinateur inclut la capture, le traitement et l'analyse d'images afin de faciliter l'évaluation objective et non destructive des caractéristiques visuelles des produits alimentaires [6]. Comme montrer la figure 3, un système de vision par ordinateur accepte, en entrée, une image capturée par un capteur passif (caméra) et produit, en sortie, un certain nombre des indices visuels extraits de l'image. Ces primitives sont appelées caractéristiques et sont fournies à un système de reconnaissance de forme qui prend une décision concernant la classe d'appartenance de l'image ou de l'objet ou les objets incorporés [2].

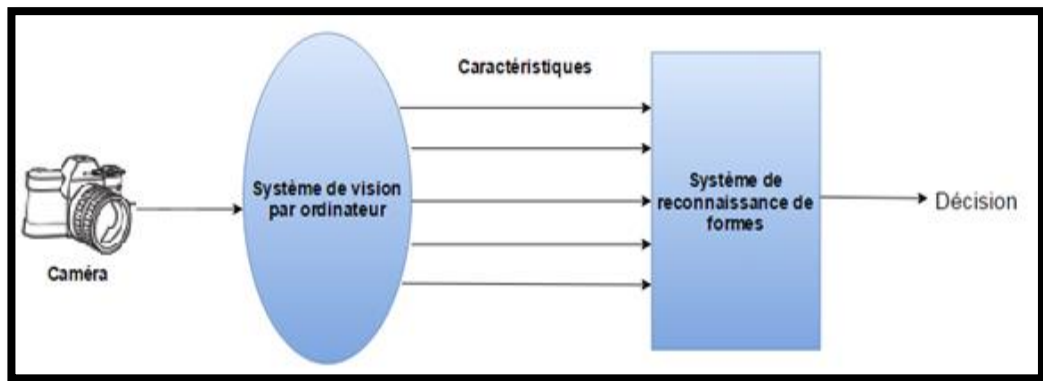


Figure 3 : Le système de vision par ordinateur[2]

A la lumière de ce qui précède, un système de tri automatique à base de la vision par ordinateur acquiert une ou plusieurs images du produit agricole, les prépare et les analyse pour extraire certaines caractéristiques visuelles du produit. Dans ce type de systèmes de tri automatique, plusieurs caractéristiques sont généralement utilisées telles que: la couleur, la taille, la forme et la texture.

2.3.2.1. Couleur

La couleur est un paramètre important pour définir la qualité de produit agricole, car elle influe directement sur la décision du consommateur à accepter ou rejeter un produit particulier. Donc, les producteurs cherchent pour prévenir les défauts de couleur d'apparaître dans les produits de sortir sur le marché. Actuellement, de nouveaux outils sont utilisés pour mesurer les changements dans la propriété de couleur des produits agricole. Ces techniques sont basées principalement sur la vision par ordinateur [8].

La propriété de couleur présente de nombreux avantages, tels qu'une efficacité élevée et une facilité d'extraction des informations de couleur à partir d'images. L'espace colorimétrique RVB, la zone HSI et la zone CIELab sont couramment utilisés pour contrôler la qualité des fruits et des légumes [7].

2.3.2.2. Texture

La perception d'une couleur par un système de vision par ordinateur est différente si les surfaces des objets ont des textures différentes. Pour cette raison, l'étude de la texture est souvent intégrée aux études sur les différences de couleur, y compris celles utilisées pour détecter la présence de défauts externes [5]. En extrayant les valeurs d'intensité entre les pixels, la texture est une primitive de classification très appropriée pour une large gamme des systèmes automatiques basés sur la vision par ordinateur

pour la reconnaissance et l'interprétation, où elle joue un rôle important dans la classification d'images, et par conséquent dans la résolution des problèmes de tri automatique des fruits et légumes [7]. La figure 4 montre des oranges de différents couleurs et textures.



Figure 4 : Images d'oranges montrant différents couleurs et textures [5]

2.3.2.3. Taille

La taille est une caractéristique importante qui peut influencer directement la qualité de nombreux produits agricoles. L'estimation de cet attribut dans des objets réguliers, tels que des fruits sphériques ou presque sphériques, est relativement facile, mais elle devient plus complexe dans les fruits et les légumes aux formes irrégulières. Le calcul de la taille d'un objet est en fonction des variables mesurés du périmètre, la longueur et la largeur. En raison des irrégularités dans la forme des produits agricoles naturels, cette caractéristique est largement utilisée pour le tri automatique dans les industries [5].

2.3.2.4. Forme

La forme est un indice visuel plus important dans l'identification des produits agricoles de qualité désirée. La caractérisation par forme est un avantage pour l'homme, mais difficile à identifier ou à définir par ordinateur. En outre, les produits agricoles et biologiques sont uniques par nature et les facteurs de l'environnement peuvent affecter leurs formes. Donc, ce semble un défi pour les techniques de classification basées sur la vision par ordinateur [8]. La figure 5 montre des avocats de différentes formes.

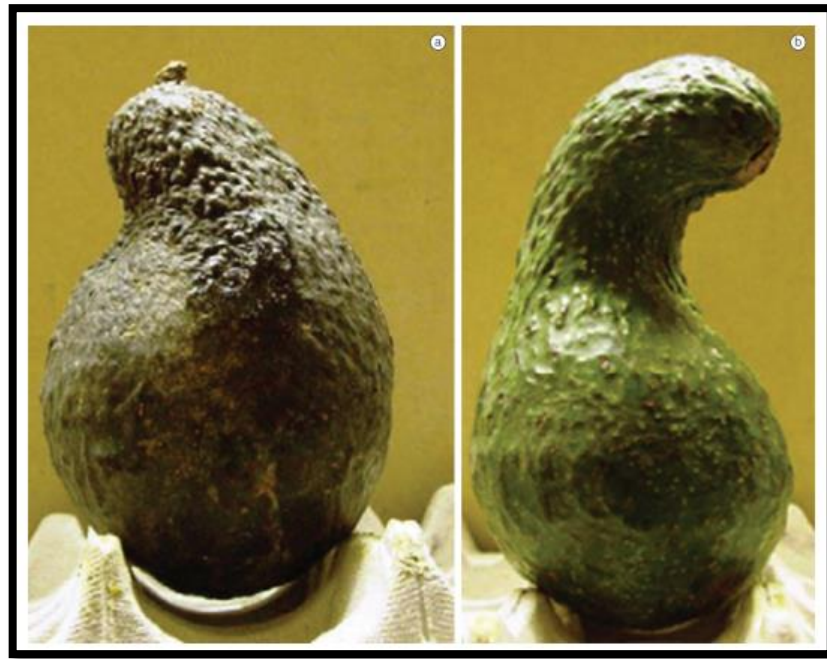


Figure 5 : plusieurs formes d'avocats [5]

2.4. Architecture d'un système de tri automatique

Par la suite nous allons découvrir les composants de base d'un système de tri et les caractéristiques de pommes de terre.

2.4.1. La composition d'un système de tri

Généralement les système de tri automatique se composent de plusieurs modules:

- **Module d'acquisition d'images**

La capture et la perception de la produit agricole est la première étape dans le processus de tri. L'image capturée dans cette étape est dans un état brut ce qui engendre un risque de bruit. Donc, Il vaut mieux de capter l'information pertinente sans bruit par le capteur (la camera), afin d'éviter la dégradation des performances du système [38].

- **Module de prétraitement**

Le prétraitement est la préparation des données du capteur pour la phase suivante. Le but de cette étape est d'éliminer autant que possible les informations non pertinentes en raison du bruit résultant de l'acquisition d'images tout en ne conservant que les informations importantes du modèle représentatif. Dans le prétraitement, une

entrée d'image ou une partie de l'image est également recadrée et redimensionnée à une taille fixe. Cela est nécessaire car à l'étape suivante, les caractéristiques sont extraites à partir d'images de taille fixe [4].

- **Module d'extraction des caractéristiques**

Les caractéristiques sont des informations pertinentes extraites d'une image. Elles sont les fonctions des variables de mesure d'origine utiles pour la classification et / ou la reconnaissance de formes, et les facteurs fondamentaux du système de vision par ordinateur car elles consistent en des données efficaces pour visualiser, interpréter et classer les images. Le but d'extraire des caractéristiques est de former des paramètres étiquetés de vecteurs pour identifier les entrées afin d'augmenter le taux de reconnaissance.

Au moment de décider sur les informations qui pourraient caractériser les produits agricoles, le choix des caractéristiques visuelles comme la couleur, la texture et la forme permet évidemment de quantifier et de représenter globalement au localement l'image de ces produits. Cependant, cette approche risque moins de produire de bons résultats dans le cas où un seul vecteur de caractéristiques est choisi. Car ces espèces ont de nombreux attributs en commun. En effet, un besoin de quantifier l'image en combinant différents descripteurs de caractéristiques est nécessaire afin qu'elle décrive l'image plus efficacement [9].

Comme notre projet traite la pomme de terre comme exemple d'un produit agricole, dans ce qui suit, nous allons présenter les différentes caractéristiques de la pomme de terre.

2.4.2. Caractéristiques de la pomme de terre

La pomme de terre est la plante de l'espèce solanacées et est l'une des cultures légumières les plus répandues dans le monde. De même, elle a aussi une valeur nutritive et des effets néfastes sur la santé. Par ailleurs, elle est classée à la quatrième place en termes d'importance après le blé, le riz et le maïs. Les pommes de terre ont besoin de soins spéciaux lorsqu'ils sont stockés, comme ils sont stockés dans des chambres froides pour éviter leur corruption. En Algérie, la wilaya d'EL Oued est l'un des principaux producteurs de pommes de terre, dont les plus importants cultivars cultivés dans cette région sont : Sputna, Kiroda, Bartina, Condor, Kabila et Cardina [3].

Nous nous sommes intéressé dans notre étude aux deux catégories de la pomme de terre à savoir: Sputna et Bartina

Les caractéristiques de Sputna, entre autres, sont [3]:

- Gros tubercules
- Forme longue
- une pelure jaune mince

La figure 6 montre quelques exemples des pommes de terre "Sputna".



Figure 6 : Exemples sur la catégorie de la pomme de terre « Spunta »

Les caractéristiques de Bartina, entre autres, sont [3]:

- Gros tubercules .
- Forme ronde .
- une pelure rouge rugueuse avec de grands yeux lisible.

La figure 7 montre quelques exemples des pommes de terre de catégorie Bartina:



Figure 7 : Exemples sur la catégorie de la pomme de terre « Bartina »

Le succès de la cultivassions de la pomme de terre dans la région d'El oued est dû à plusieurs facteurs naturels et économiques, notamment [3]:

- La disponibilité du sol et les terres de sable convient pour la culture des pommes de terre.
- La disponibilité des sources d'eau d'irrigation et la facilité d'exploitation des eaux de surface.

- Des facteurs climatiques favorables.
- La conviction et la volonté des citoyens de l'importance d'investir dans ce type...etc.

Après que nous ayons vu les différents concepts liés au processus de tri automatique, nous pouvons constater que le problème de tri automatique des pommes de terre ne se résume qu'à l'identification de la classe de la pomme de terre en se basant sur les caractéristiques visuelles extraites de une image, voire plusieurs, de celle-ci. On parle ainsi du problème de classification. Dans ce qui suit, nous allons présenter en détaille le domaine de reconnaissance de forme est plus particulièrement le problème de classification.

3. Reconnaissance de formes (pattern recognition)

3.1. Qu'est-ce qu'une reconnaissance de formes

La reconnaissance automatique (par machine), la description, la classification et le regroupement de formes (modèles ou motifs) sont des problèmes importants dans diverses disciplines scientifiques et d'ingénierie telles que la biologie, la psychologie, la médecine, le marketing, la vision par ordinateur et l'intelligence artificielle [10]. On désigne par reconnaissance de formes (ou parfois reconnaissance de motifs ou pattern recognition en anglais) un ensemble de techniques et méthodes visant à identifier des motifs à partir de données brutes afin de prendre une décision dépendant de la catégorie attribuée à ce motif. On considère que c'est une branche de l'intelligence artificielle qui fait largement appel aux techniques d'apprentissage automatique (machine Learning) et aux statistiques [11].

- **Qu'est-ce qu'un motif (a pattern)**

Watanabe a défini une forme comme: « l'opposé du chaos; c'est une entité vaguement définie, à laquelle on peut associer un nom », par exemple, un motif pourrait être une image d'empreinte digitale, un mot cursif manuscrit, un visage humain ou un signal de parole [10]. En d'autres termes, une forme, en informatique, est un ensemble de valeurs, appelés attributs, auxquels sont associés un nom (ou étiquette) et qui ont une classe d'appartenance. Plusieurs formes peuvent avoir la même classe, on dit alors que ce sont les exemples ou réalisations de la classe [11].

Les formes ou motifs à reconnaître peuvent être de nature très variée. Il peut s'agir de contenu visuel (code barre, visage, empreinte digitale...) ou sonore (reconnaissance de parole), d'images médicales (rayon X, EEG, IRM...) ou multi spectrales (images satellitaires) et bien d'autres [11].

Le processus de reconnaissance de formes, comme modèles, implique essentiellement les trois aspects suivants [10]:

- ✓ Acquisition de données et prétraitement.
- ✓ Représentation des données.
- ✓ Prise de décision.

La décision de ce qu'est un objet, sa classe, est prise selon plusieurs approches de reconnaissance des formes à savoir une reconnaissance de formes statistiques et celles structurelles.

3.2. Reconnaissance statistique de formes

La reconnaissance statistique de formes a été largement utilisée avec succès pour concevoir un certain nombre de systèmes de reconnaissance. En reconnaissance statistique de motifs, un motif est représenté par un ensemble de caractéristiques, ou des attributs, regroupé dans un vecteur, voire plusieurs, de n dimensions. Des concepts bien connus de la théorie de la décision statistique sont utilisés pour établir des limites de décision entre les classes de motifs [10]. Pratiquement, un système de reconnaissance statistique de formes fonctionne sur la base d'un processus de machine Learning.

3.3. Que est ce que un Machine Learning

L'apprentissage automatique (ou Machine Learning en anglais) est une discipline de l'IA qui offre aux ordinateurs la possibilité d'apprendre à partir d'un ensemble d'observations que l'on appelle ensemble d'apprentissage [12]. En général, l'objectif de l'apprentissage automatique est de comprendre la structure des données et de les intégrer dans des modèles qui peuvent être compris et utilisés par tout le monde [13]. En conséquence, il est maintenant possible de produire de façon rapide et automatique des modèles capables d'analyser plus rapidement des tailles de données plus grandes et plus complexes [14]. Machine Learning est maintenant utilisé dans de nombreux domaines, tels que le développement de véhicules autonomes, les systèmes de

recommandation en ligne tels que Netflix et Amazon, l'analyse du moral des clients et la détection des fraudes [1].

Comme montrer la figure 8, le processus de ML est constitué d'enchaînement des étapes. A partir des données d'entraînement, une étape d'extraction de caractéristiques est effectuée. Ce qui résulte un vecteur de caractéristiques. Ce vecteur (voire plusieurs) sera utilisé dans l'étape d'apprentissage hors ligne. Ceci afin de construire le modèle de décision. Le modèle construit est utilisé à des fins de test (mesurer les performances) en fournissant des données de test. Bien qu'il est aussi utilisé pour prédire la classe des nouvelles données.

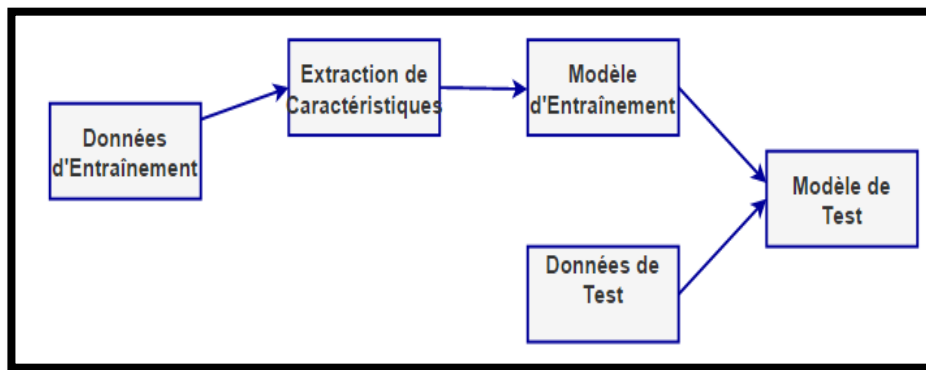


Figure 8 : le processus typique du ML[12]

- **Schéma générique d'un système de reconnaissance de formes**

La figure 9 illustre le schéma générique d'un système de reconnaissance de forme. Sur cette figure, on peut observer les deux tâches réussies: d'une part, la tâche d'analyse / description (voir 1 sur la figure 9) et d'autre part, la tâche de classification / reconnaissance (voir 2 sur la figure 9) [15].

En se basant sur les processus de machine Learning, un système de reconnaissance fonctionne en deux phases superposées : 1) une phase entraînement « d'apprentissage hors ligne » (Learning) et 2) une phase de classification (testing) (voir Figure 9).

- ✓ En phase d'entraînement: le module d'extraction / de sélection d'entités trouve les entités appropriées pour représenter les modèles d'entrée (images capturées) et le classificateur est entraîné à partitionner l'espace des entités.

- ✓ En phase de classification : le classifieur formé assigne le motif d'entrée à l'une des classes de motifs considérées en fonction des caractéristiques mesurées (donner une interprétation aux images d'entrée) [10].

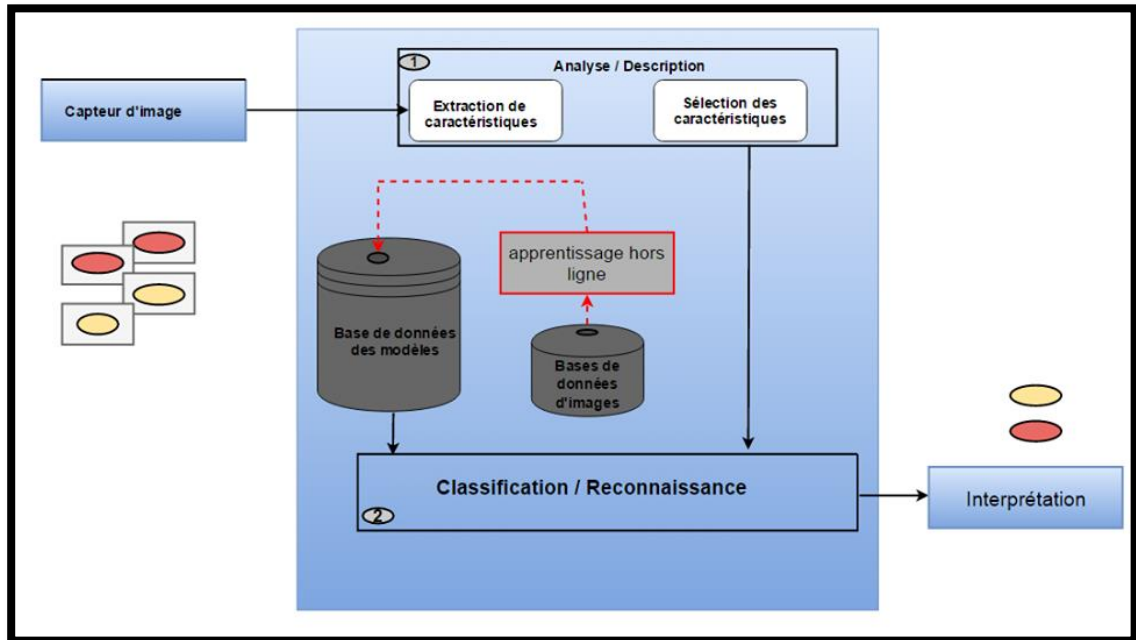


Figure 9 : Schéma général d'un système de reconnaissance des formes [15]

Les caractéristiques extraites doivent être discriminantes que possible pour réduire les fausses alarmes dues à une mauvaise classification. En plus ces caractéristiques doivent également présenter certaines des propriétés essentielles telles que [15]:

- Invariance de rotation: les caractéristiques extraites ne doivent pas varier avec la rotation du motif.
- Invariance d'échelle: le changement d'échelle ne doit pas affecter les caractéristiques extraites.
- Résistance au bruit: les caractéristiques doivent être aussi robustes que possible contre le bruit, c.-à-d. ils doivent être les mêmes quelle que soit la force du bruit qui affecte le motif.

3.4. Types d'apprentissage

Il existe une multitude de différents algorithmes d'apprentissage automatique. Compte tenu des données appropriées, ces algorithmes créent des modèles du problème en interagissant avec ces données de manière spécifique. En fonction de la manière de création du modèle ainsi que de présentation des données d'apprentissage, les algorithmes d'apprentissage automatique peuvent s'inscrire dans quatre catégories de techniques d'apprentissage à savoir : l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé et l'apprentissage par renforcement.

3.4.1. Apprentissage supervisé

Dans l'apprentissage supervisé, l'ordinateur est fourni avec des exemples d'entrées qui sont étiquetés avec les sorties souhaitées. Le but de cette méthode est que l'algorithme puisse «apprendre» en comparant sa sortie réelle avec les sorties «enseignées» pour trouver des erreurs de prédiction et modifier le modèle en conséquence. L'apprentissage supervisé vise à prédire les valeurs d'étiquettes sur des données non étiquetées supplémentaires en utilisant des modèles construits [13].

3.4.2. Apprentissage non supervisé

Dans l'apprentissage non supervisé, les données sont non étiquetées, de sorte que l'algorithme d'apprentissage trouve tout seul des points communs parmi ses données d'entrée. Les données non étiquetées étant plus abondantes que les données étiquetées, les méthodes d'apprentissage automatique qui facilitent l'apprentissage non supervisé sont particulièrement utiles. L'objectif de l'apprentissage non supervisé est de découvrir des modèles cachés dans un ensemble de données, mais il peut aussi avoir un objectif d'apprentissage des caractéristiques.

L'apprentissage non supervisé est souvent utilisé pour la détection d'anomalies, par exemple les photos non marquées des chiens peuvent être utilisées comme données d'entrée pour l'algorithme afin de trouver des similitudes et de classer les photos de chiens ensemble [13].

3.4.3. Apprentissage semi-supervisé

L'apprentissage semi-supervisé est utilisé pour les mêmes applications que l'apprentissage supervisé. Tous les deux utilisent à la fois des données étiquetées et non-étiquetées pour s'entraîner. En règle générale, une petite quantité de données étiquetées est utilisée avec une grande quantité de données non étiquetées.

Ce type d'apprentissage peut être utilisé avec des méthodes comme la classification, la régression et la prédiction. L'apprentissage semi-supervisé est utile quand les coûts associés à l'étiquetage sont trop élevés pour permettre un processus d'apprentissage entièrement étiqueté. Il est par exemple utilisé pour identifier le visage d'une personne sur une webcam [16].

3.4.4. Apprentissage par renforcement

L'apprentissage de renforcement est souvent utilisé pour la robotique, le jeu vidéo, échecs et la navigation. Grâce à l'apprentissage de renforcement, l'algorithme multiplie les tentatives pour tenter de découvrir quelles actions apportent les plus grandes récompenses. Ce type d'apprentissage regroupe trois principaux composants : l'agent (qui apprend ou prend les décisions), l'environnement (tout ce avec quoi l'agent interagit), et les actions (ce que peut faire l'agent). L'objectif est que l'agent choisisse les actions qui maximisent les récompenses attendues sur une période donnée. L'objectif de l'apprentissage de renforcement est donc d'apprendre les meilleures règles [16].

4. Extraction de caractéristiques

Il existe plusieurs caractéristiques, ou descripteurs, de l'image qui peuvent être utilisées pour mesurer la similarité d'une image ou la classification des images ou des parties (patches) d'une image. Généralement on peut distinguer des caractéristiques globales, voire locales.

4.1. Caractéristiques locales

Les caractéristiques locales concernent uniquement la détection de points d'intérêt sur des régions (une partie) de l'image à l'aide de segmentation. Après, les caractéristiques locales sont calculées autour de points d'intérêt détectés. En effet, l'analyse basée sur des caractéristiques locales risque de perdre le sens global de l'image, en submergeant celui-ci dans des petits détails inutiles. Il y a des algorithmes SIFT [17] et SURF [18] permettent de détecter les points d'intérêt et de construire des descripteurs locaux robustes [16].

4.2. Caractéristiques globales

Ils concernent la recherche des caractéristiques globales sur toute l'image comme la couleur, texture, forme, ...etc. cependant, un système basé uniquement sur des caractéristiques globales ne peut pas donner les résultats désirés. Un descripteur global est plus facile à utiliser car il traite l'image dans son intégralité. Le descripteur est ainsi moins sensible aux déformations et perturbation d'une image à l'autre. Un descripteur global est plus facile à utiliser car il traite l'image dans son intégralité. Le descripteur est ainsi moins sensible aux déformations du piéton d'une image à l'autre [16].

4.2.1. Caractéristiques basées sur la couleur

La couleur est l'une des caractéristiques les plus importantes des images. Les caractéristiques de couleur sont définies en fonction d'un espace de couleur ou d'un modèle particulier. Un certain nombre d'espaces de couleurs ont été utilisés dans la littérature, tels que RVB[20] [21], LUV[22], HSV[23]et HMMD [24] [25] . Une fois que l'espace colorimétrique est spécifié, la fonction de couleur peut être extraite à partir d'images ou de régions[19].

4.2.2. Caractéristiques basées sur texture

La texture est une caractérisation très utile pour une large gamme d'images. Au contraire à la caractéristique de couleur, la mesure de la texture ne se fait qu'à partir d'un groupe de pixels. Un grand nombre de techniques ont été proposées pour extraire les caractéristiques de texture [19]. Parmi celles nous mentionnons LBP [26], HOG [27] et HTD [24] comme des descripteurs de caractéristiques les plus performants.

4.2.3. Caractéristiques basées sur la forme

La forme est considérée comme un indice important permettant aux êtres humains d'identifier et de reconnaître les objets du monde réel. Les techniques d'extraction des caractéristiques de forme peuvent être classées en deux groupes, à savoir les méthodes basées sur les contours et les régions. La première calcule les caractéristiques de forme uniquement à partir de la limite de la forme, tandis que la dernière méthode extrait les caractéristiques de la région entière [19]. Parmi celles nous mentionnons MPEG7 [28], 3Dshape descripteur [28] comme des descripteurs de caractéristiques les plus performants.

4.3. Fusion de caractéristiques

Comme montrer la figure suivante, la fusion au niveau des caractéristiques implique que les données enregistrées à partir d'un ou plusieurs capteurs, et ainsi traitées par un algorithme d'extraction de caractéristiques dédié, sont fusionnées en un seul vecteur c.-à-d. Les vecteurs caractéristiques estimés de chaque capteur sont concaténés en un seul vecteur caractéristique. Par conséquent, l'ensemble d'apprentissage est une matrice de données $M \times (N.f)$, où M est le nombre de trames dans l'ensemble d'apprentissage, N le nombre d'unités de capteur et f le nombre de caractéristiques extraites de chaque unité de capteur. En effet, un vecteur de caractéristique de grande dimensionnalité, imposant le besoin soit de sélection de caractéristique avant la classification soit de transformation de données dans un espace de dimension inférieure [29].

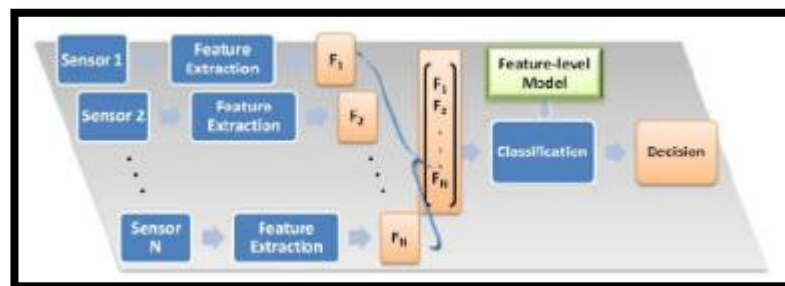


Figure 10 : schéma descriptif de la fusion des caractéristiques[29]

5. Classification

La classification est généralement considérée comme une spécialisation dans le contexte de l'apprentissage automatique, un sous-champ de l'intelligence artificielle. La classification est une forme d'apprentissage supervisé [30]. C'est une tâche très importante dans la vision par ordinateur. En utilisant les modèles (paramètres) obtenus lors de l'apprentissage, cette phase constitue le noyau de la reconnaissance des formes [11]. De manière plus générale, le terme pourrait couvrir tout contexte dans lequel une décision ou une prévision est prise sur la base des informations actuellement disponibles, et une procédure de classification est alors une méthode formelle pour faire à plusieurs reprises ces jugements dans de nouvelles situations [31].

5.1. Classification binaire

La classification supervisée vise à regrouper, par apprentissage, divers vecteurs de caractéristiques en sous-ensembles de classes d'objets. Comme son nom l'indique, la classification binaire est tout simplement un cas particulier dans lequel il n'y a que deux classes c.-à-d. chaque vecteur appartient à l'un des 2 différentes classes. L'objectif donc est de construire une fonction de décision qui prédit correctement la classe à laquelle un nouveau vecteur appartient [32].

5.2. Classification multi-classes

Dans l'apprentissage de la machine, la classification multi-classes ou multinomiale est le problème de la classification dans les cas où on a plus de deux classes. Le problème de la classification multi-class peut être décomposé en plusieurs tâches de classification binaires pouvant être résolues efficacement à l'aide de classificateurs binaires [33].

5.2.1. Approche un-contre-tout (One-Against-All)

Ce schéma classique de classification multi-classes est reconnu comme une méthode robuste, simple à mettre en place et capable de gérer des problèmes volumineux. Cette méthode a l'avantage d'être parallélisable car l'apprentissage et l'utilisation des classifieurs sont indépendants les uns des autres [34]. L'approche U-C-T (O-A-A) utilise les classificateurs binaires distinguant chaque classe des autres classes [35]. D'où on construit k classificateurs, un pour chaque classe. Le classifieur construit un hyperplan entre la classe m et les $k-1$ autres classes. Cette méthode a été largement

utilisée dans la littérature sur les vecteurs de support pour résoudre des problèmes de reconnaissance de formes à multi-class [36].

5.2.2. Approche un contre un (One-Against-One)

L'approche U-C-U (O-A-O) est assez semblable au schéma de classification (O-A-A). Le principe est de constituer un ensemble de classifieurs correspondant à l'ensemble des paires de classes qu'il est possible de constituer dans la base d'apprentissage [34]. Cette approche implique la construction d'une machine pour chaque paire de classes, ce qui donne $k(k-1)^2$ machines. Pour chaque paire distincte m_1 et m_2 , nous exécutons l'algorithme d'apprentissage sur un problème binaire dans lequel les exemples étiquetés $y = m_1$ sont considérés comme positifs et ceux étiquetés $y = m_2$ sont négatifs [36].

5.3. Algorithmes de classification

5.3.1. K plus proches voisins (KNN)

L'algorithme des K plus proches voisins (ou K-Nearest Neighbors en anglais) est un algorithme de classification supervisé. Elle consiste à supposer qu'une observation est similaire à celle de ses voisins. On suppose que chaque observation de l'ensemble d'entraînement représentée par un point dans un espace comportant autant de dimensions qu'il y a de variables prédictives. Pour prédire la classe d'une observation, on cherche les k points les plus proches de cet exemple. La classe de la variable cible, est celle qui est la plus représentée parmi les k plus proches voisins. Il existe des variantes de l'algorithme ou on pondère les k observations en fonction de leur distance à l'exemple dont on veut classer, les observations les plus distantes étant considérées comme les moins importantes Les k plus proches voisins. Les avantages et les inconvénients de KNN sont[14]:

✓ Avantages

- On ne présuppose rien sur la distribution des données si ce n'est qu'il existe une notion de similarité
- simple à concevoir.

✓ Inconvénients

- L'algorithme est très sensible au bruit.
- L'apprentissage est uniquement local et ne tient pas compte des effets collectifs à longue distance.

- Lorsque le nombre de variable est grand (> 10) le calcul de la distance peut s'avérer très coûteux. Il faut alors utiliser des techniques de réduction dimensionnelle.

5.3.2. Les machines à vecteurs de support (SVM)

Les machines à vecteurs de support, ou séparateurs à vaste marge (SVM), sont des algorithmes de classification binaire non linéaires qui font partie des méthodes d'apprentissage supervisé. Ce sont les algorithmes les plus performants et les plus répandus en classification image [14]. Les échantillons d'apprentissage sont représentés par un ensemble de paires d'entrée / sortie où la sortie est une étiquette binaire pour une discrimination à deux classes. Le but est de construire une fonction à partir de ces exemples d'apprentissage qui peut prédire les sorties pour des entrées n'appartenant pas à l'ensemble des données d'apprentissage. Les entrées peuvent être des descriptions d'objets et les sorties sont les classes de ces objets données en entrées [37].

Le principe des SVM consiste à construire une bande séparatrice non linéaire de largeur maximale qui sépare deux ensembles d'observations et à l'utiliser pour faire des prédictions. Les avantages et les inconvénients de SVM sont [14]:

✓ Avantages

- Il permet de traiter des problèmes avec un très grand nombre de dimensions et d'éviter le fléau de la dimension.
- Il permet de traiter des problèmes de classification non linéaire complexes.
- Les SVM constituent une alternative aux systèmes de neurones car ils répondent au même besoin de classification non linéaire tout en étant beaucoup plus faciles à entraîner que ces derniers .

✓ Inconvénients

- Le choix de la fonction noyau K est délicat et possède un caractère un peu mystérieux qui ne peut être étayé que par l'expérience.
- Bien que l'algorithme puisse être entraîné avec des ensembles de données de plusieurs dizaines de milliers d'observations il n'est cependant pas scalable. On estime que sa complexité croît comme N^3 où N est le nombre d'observations.
- Il est souvent moins performant que les forêts aléatoires.

6. Conclusion

Dans ce chapitre, nous avons identifié les éléments et les concepts fondamentaux les plus importants en ce qui concerne les systèmes de tri en général, les principaux types de mesures qui ont été utilisés pour donner une idée des systèmes de tri automatique et son importance dans la qualité des produits agricoles. Aussi nous avons expliqué quelques-uns des concepts liés à la vision par ordinateur, la reconnaissance des formes et des techniques d'apprentissage automatique.....etc .

Dans le chapitre suivant, nous viserons à la conception de notre système proposé de tri automatique basé sur la vision par ordinateur pour les pommes de terre. En effet, nous présenterons l'architecture de notre système, puis les descripteurs de caractéristiques ainsi que la méthode de classification utilisée dans ce système proposé.

Chapitre 2

Conception du système

1. Introduction

Les systèmes de tri automatique des produits agricoles sont apparus pour améliorer l'efficacité du processus de tri traditionnel. Ces systèmes sont basés sur la vision par ordinateur pour trier automatiquement et distinguer entre les différents produits selon des caractéristiques visuelles décrivant la qualité de ceux-ci. Ces caractéristiques sont extraites à partir des images acquises des produits en question.

Dans ce chapitre, nous allons donner une description générale du système de tri automatique de pomme de terre que nous proposons. En effet, nous présentons la conception architecturale de notre système proposé puis nous détaillons cette architecture en clarifiant le principe de fonctionnement de chaque module.

2. Conception du Système de tri de pomme de terre proposé

Dans cette section nous allons évoquer la conception de notre système de tri proposé.

2.1 Description du notre système de tri

Le fonctionnement du système de tri que nous avons proposé dépend des éléments suivants (voir figure 11) qui consistent en :

- ✓ Un convoyeur transportant les pommes de terre un à un.
- ✓ La chambre de capture d'image dans laquelle une caméra numérique installée dans le centre supérieur de la boîte pour la prise de vue plongée des pommes de terre. En plus une lampe est placée dans le coin supérieur. Ceci dans l'objectif de fixer l'intensité de la lumière pour obtenir les meilleures images. Par la suite, les images acquises sont envoyées à l'ordinateur pour le traitement.
- ✓ Un ordinateur connecté à la caméra numérique afin de recevoir des images acquises à des fins de traitement, d'analyse et de prédire la classe en se basant sur un algorithme d'apprentissage supervisé.
- ✓ Un système d'aiguillage est au bout du convoyeur qui peut diriger les pommes de terre transportées en fonction de la décision prise par l'ordinateur.

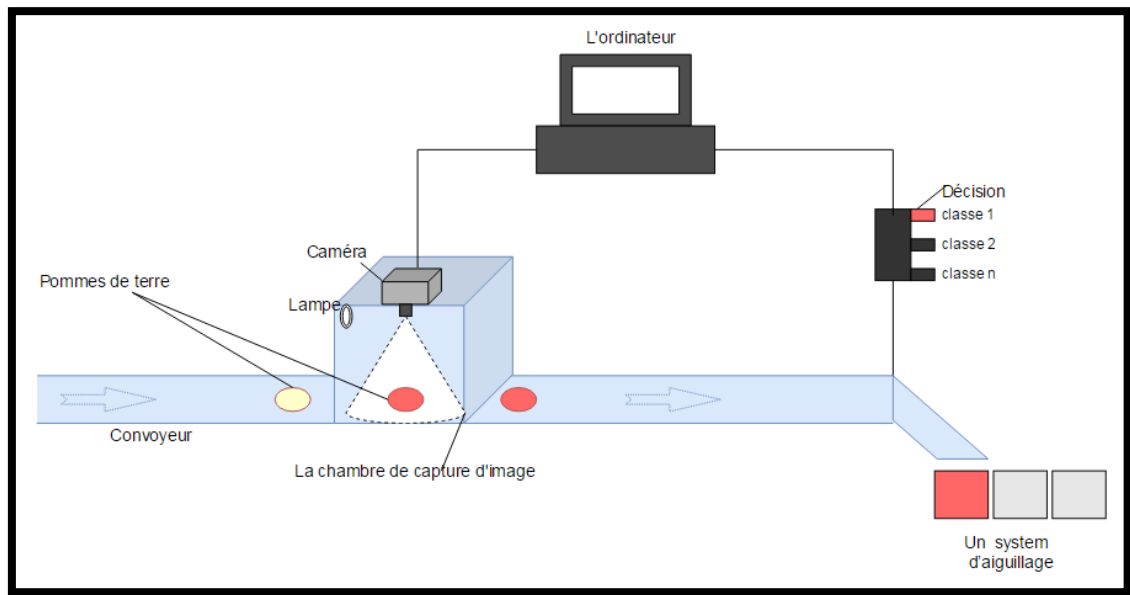


Figure 11 : Description du système de tri proposé

Dans cette chaîne de traitement, nous ne nous intéressons qu'au logiciel présent sur l'ordinateur qui résout le problème de l'apprentissage et de la classification dans le tri automatique des produits agricoles (pomme de terre dans notre cas). Ce logiciel reçoit toutes les images acquises pour le prétraitement, l'extraction de leurs principales caractéristiques, puis l'utilisation de la méthode de classification de différentes catégories pour obtenir le modèle de décision à utiliser lors de la phase de tri.

Dans la phase de tri, les caractéristiques extraites, d'une pomme de terre à trier, sont présentées au modèle de classification utilisé pour déterminer et prédire la catégorie de cette pomme de terre. Par conséquent, compte tenu de cette décision, le système d'aiguillage va diriger cette pomme de terre dans la bonne direction qui décrit la catégorie appropriée.

2.2 Architecture du système

Notre système proposé est constitué de plusieurs modules arrangés en deux étages :

- **Etage d'entraînement** : il représente la phase d'apprentissage hors ligne.
- **Etage de classification (testing)** : il s'agit de la phase de classification multi-classes en ligne des nouvelles images non vues.

Ces étages sont constitués de plusieurs modules comme montré la figure 12.

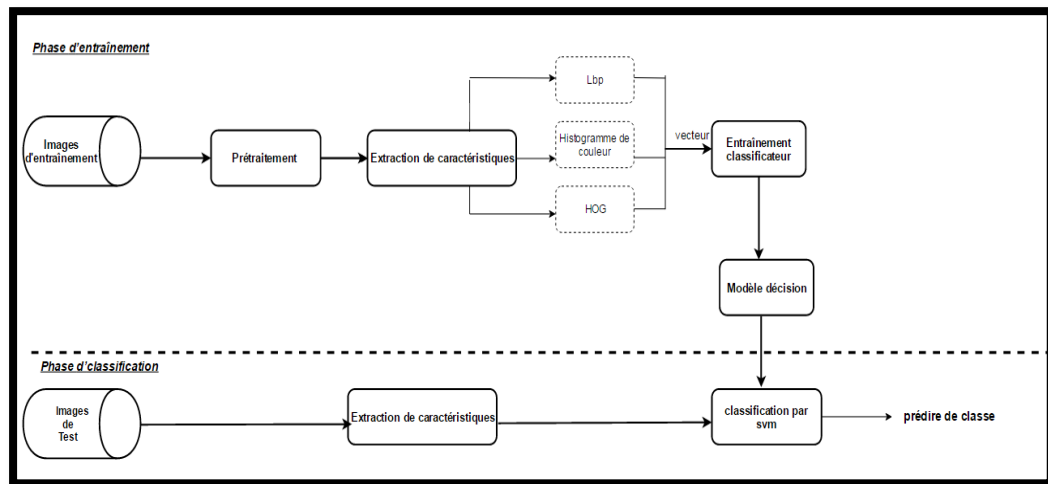


Figure 12 : l'architecture du système proposé

Dans la phase d'entraînement, le système prend comme entrée un ensemble d'images des pommes de terre. Afin de constituer la base d'entraînement, ces images sont étiquetées et redimensionnées pour avoir la même taille. Après et pour chaque image de notre base, une représentation intermédiaire, encapsulant les informations importantes concernant la classe des pommes de terre, est calculée. Les vecteurs de caractéristiques obtenus sont utilisés pour entraîner le classificateur afin de différencier entre les modèles de classes. Cette représentation a été calculée par l'utilisation de trois descripteurs différents, qui sont : le Motif Binaire Local (LBP), les histogrammes de couleur, et l'Histogramme des Gradients Orientés (HOG). Ceci pour pouvoir distinguer l'information la plus pertinente pour une bonne caractérisation. Afin d'améliorer de la performance de notre classificateur, la méthode de validation croisée a été utilisée dans le but d'aboutir à une bonne minimisation d'erreur de généralisation de nos données d'apprentissage.

Dans la phase de test, nous nous intéressons à la classification multi-classe de la pomme de terre dans la nouvelle image présentée au système. Ce dernier fait prédire la classe de la pomme de terre représentée par cette image. En effet, on extrait les mêmes caractéristiques prédéfinies dans la phase d'entraînement, ensuite elles sont injectées dans le classificateur (SVM). La sortie va déterminer si l'image représente une pomme de terre ou pas et ainsi sa catégorie.

Nous notons que dans cette phase, il est nécessaire de redimensionner l'image en entrée et effectuer le même prétraitement utilisé dans la phase d'entraînement.

2.3 Conception détaillée du système

Notre système développé est constitué de quatre modules de base comme suit :

2.3.1 Module d'acquisition d'images

la première étape dans le processus de tri est la capture et la perception de la pomme de terre . Pour cela nous avons utilisé un caméra comme un capteur passif pour acquérir ces images-là .

2.3.2 Module de prétraitement

Après l'acquisition des images des pommes de terre ,une phase de prétraitement est nécessaire. Dans cette phase une image ou une partie de l'image en entrée est recadrée et redimensionnée à une taille fixe. Cela est nécessaire car à l'étape suivante, les caractéristiques sont extraites à partir d'images de taille fixe .

2.3.3 Module d'extraction des caractéristiques

Les caractéristiques sont des informations pertinentes extraites d'une image qui pourraient caractériser les pommes de terre. Le choix de ces informations à une influence sur la décision de système de tri. En effet, nous pourrions éventuellement penser à la couleur, la texture et la forme comme étant les principales caractéristiques visuelles.

Dans ce qui suit, nous allons discuter les trois plus importants descripteurs globales qui nous avons l'utilisés pour l'extraction de caractéristiques.

2.3.3.1 Motif Binaire Local (LBP)

Dans notre système nous avons utilisé le descripteur « Motif Binaire Local » (local binary patterns en anglais) pour extraire la caractéristique de texture. Ce descripteur a été mentionné pour la première fois en 1993 pour mesurer le contraste local d'une image [39]. Mais réellement popularisé trois ans plus tard par Ojala et al pour analyser les textures [40]. Les motifs Binaires Locaux (LBP) ont suscités un intérêt croissant dans le traitement d'images et la vision par ordinateur. Il résume les structures locales des images de manière efficace par comparaison de chaque pixel avec ses pixels voisins. Il a été utilisé dans de nombreuses applications telles que la reconnaissance de visage [41].

- **Concept**

L'idée de cet opérateur de texture est d'assigner à chaque pixel un code dépendant des niveaux de gris de son voisinage. Le niveau de gris du pixel central (i_c) de coordonnées (x_c, y_c) est comparé à celui de ses voisins (i_n) suivant l'équation (1) :

$$LBP(x_c, y_c) = \sum_{m=0}^{p-1} s(i_n - i_c) \times 2^n \quad (1)$$

$$s(i_n - i_c) = \begin{cases} 1, & \text{si } i_n - i_c \geq 0 \\ 0, & \text{si } i_n - i_c < 0 \end{cases}$$

avec p est le nombre de pixels voisins. Dans notre travail, nous considérons un voisinage de 3×3 d'où $p=8$ voisins. Nous obtenons donc, comme pour une image en niveaux de gris, une matrice contenant des valeurs des LBP comprises entre 0 et 255 pour chaque pixel. Un histogramme est calculé en se basant sur ces valeurs pour former le descripteur LBP [42].

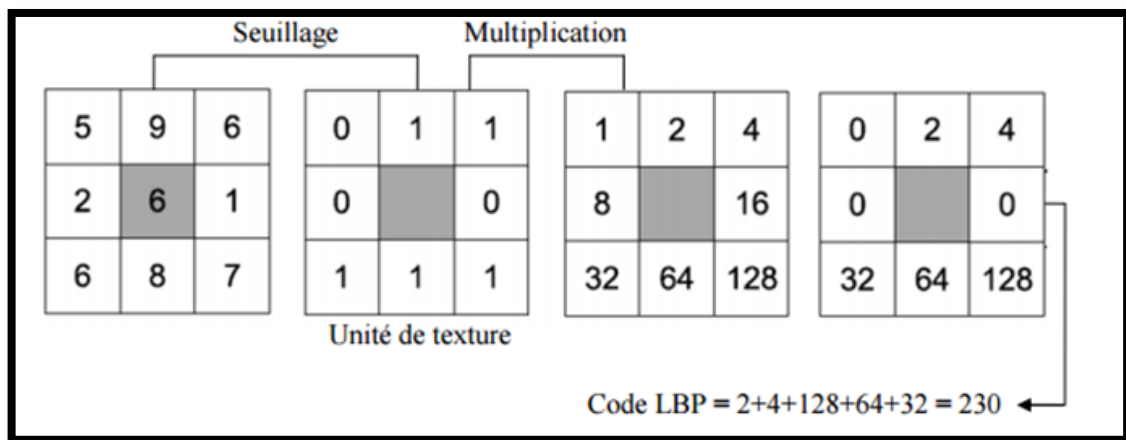


Figure 13 : Construction d'un motif binaire et calcul du code LBP [42]

2.3.3.2 Histogrammes de Couleurs

Les histogrammes de couleurs sont fréquemment utilisés pour classifier les images et identifier les objets dans les images. Leur utilisation est motivée du fait qu'ils sont faciles à calculer et ont tendance à être robustes face à de petites modifications du point de vue de la caméra [43]. Dans notre système, nous avons utilisé les histogrammes de couleurs pour extraire la caractéristique de couleur.

Pratiquement, les images sont généralement représentées dans un espace de couleur où chaque couleur est représenté par un canal. Un histogramme de couleur H est un vecteur h_1, h_2, \dots, h_n , dans lequel chaque groupe h_j contient le nombre de pixels de couleur j dans l'image où chaque image doit avoir le même nombre de pixels (redimensionnement nécessaire) [44]. En outre, un histogramme d'image fait référence à la fonction de masse de probabilité des intensités d'image. Plus formellement, l'histogramme de couleur est défini par :

$$h_{A,B,C}(a, b, c) = N \cdot \text{Prob}(A = a, B = b, C = c) \quad (2)$$

où A, B et C représentent les trois canaux de couleur (R,G,B or H,S,V) et N est le nombre de pixels de l'image. Pour calculer, l'histogramme de couleur est formé en discrétisant les couleurs dans une image et en comptant le nombre de pixels de chaque couleur [45].

✓ Espace de couleur

Un espace de couleur est défini comme un modèle pour représenter la couleur en termes de valeurs d'intensité. Généralement, un espace de couleur définit un espace de une à quatre dimensions d'où une dimension représente une composante ou un canal de couleur. Un espace dimensionnel de couleur (c'est-à-dire une dimension par pixel) représente l'espace en niveaux de gris. En général, la classification basée sur l'histogramme dans l'espace couleur HSV affichait de meilleures performances que dans l'espace couleur RVB[46].

✓ Modèle de couleur HSV

L'espace de couleur HSV (Hue, saturation, value en anglais) est développé pour fournir une représentation intuitive de la couleur et pour approximer la façon dont les humains perçoivent et manipulent les couleurs. [24].

Une vue du modèle de couleur HSV est illustrée dans la figure 14 :

- La valeur représente l'intensité d'une couleur, qui est découplée des informations de couleur dans l'image représentée.
- Les composantes de teinte (Hue) et de saturation sont intimement liées à la façon dont l'œil humain perçoit les couleurs, ce qui donne lieu à des algorithmes de traitement d'images à base physiologique.

- Comme la teinte varie de 0 à 1.0, les couleurs correspondantes varient du rouge au jaune, le vert, le cyan, le bleu et le magenta de sorte qu'il y ait réellement des valeurs rouges à la fois à 0 et à 1.0.
- Comme la saturation varie de 0 à 1.0, les couleurs correspondantes (Hue) varient de non saturé (nuances de gris) à totalement saturé (pas de composant blanc).
- Lorsque la valeur ou la luminosité varie de 0 à 1.0, les couleurs correspondantes deviennent de plus en plus lumineuses [45].

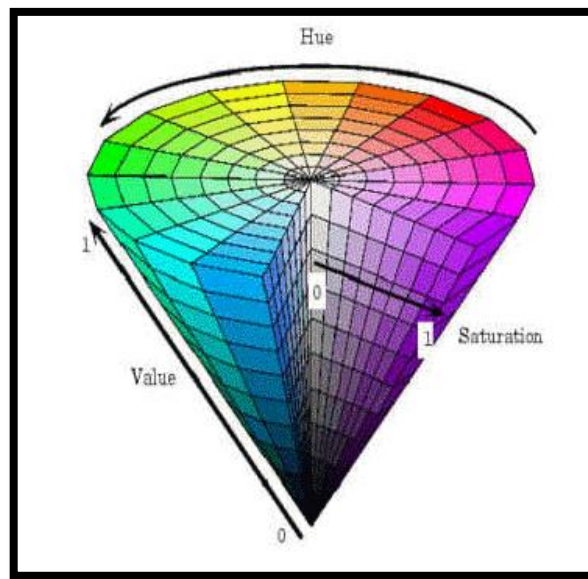


Figure 14 :Modèle de couleur HSV

✓ Conversion RVB à HSV

RVB à HSV est une transformation non-linéaire, mais réversible [45]. Dans la figure 15, les couleurs HSV pouvant être obtenues se situent dans un triangle dont les sommets sont définis par les trois couleurs primaires de l'espace RVB.

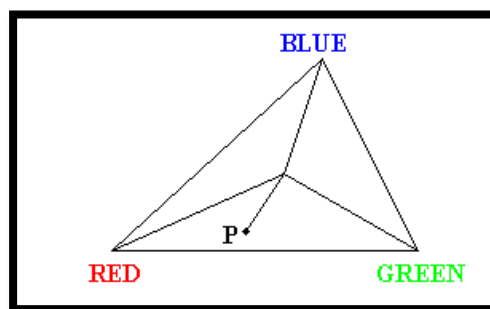


Figure 15 : Couleur de HSV obtenir à partir de l'espace colorimétrique RVB [45]

La teinte (hue) du point P est l'angle mesuré entre la ligne reliant P au centre du triangle et la ligne reliant le point ROUGE au centre du triangle. La saturation du point P est la distance entre p et le centre du triangle. La valeur (intensité) du point P est représentée comme hauteur sur une ligne perpendiculaire au triangle et passant par son centre. Les points en niveaux de gris sont situés sur la même ligne. La formule de conversion est la suivante : [45]

$$H = \cos^{-1} \left\{ \frac{\frac{1}{2(R-G)} + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right\} \quad (3)$$

$$S = 1 - \frac{3}{R+G+B} [\min(R, G, B)] \quad (4)$$

$$V =$$

$$\frac{1}{3}(R, G, B) \quad (5)$$

2.3.3.3 Histogramme des Gradients Orientés (HOG)

Le descripteur HOG (Histogram of Oriented Gradients) [27] est couramment appliquée dans la reconnaissance d'objets et les tâches de détection humaine. HOG est un descripteur basé sur une fenêtre calculé localement sur un point d'intérêt détecté (ou dans certains cas, échantillonné de manière dense sur tous les points de l'image) [47].

Le but de cette méthode est de décrire une image par un ensemble d'histogrammes locaux. Ces histogrammes comptent les occurrences de l'orientation du dégradé dans une partie locale de l'image [48].

La Figure 16 illustre l'essentiel du processus de calcul HOG . Pour extraire le HOG d'une image [49]

- Dans un premier temps, on calcule les orientations des gradients au niveau de chaque pixel (Fig. 16(a)).
- Dans un second temps, un histogramme de chaque orientation dans une petite région rectangulaire est calculé (Fig. 16(b)).
- finalement, les histogrammes de toutes les petites régions sont concaténés pour former le vecteur de caractéristiques HOG (Fig. 16(c)).

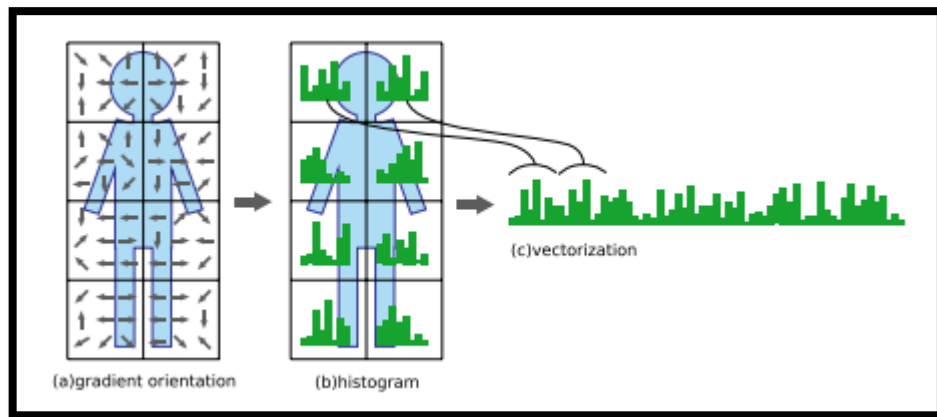


Figure 16 : Vue d'ensemble du calcul HOG [49]

L'un des avantages de HOG est la résistance à la variance de l'éclairage. En outre, la robustesse contre les déformations, car de légers décalages et des déformations affines entraînent de petites modifications de la valeur de l'histogramme [49].

Après l'obtention du vecteur de caractéristiques, le module de classification prend en entrée ce vecteur afin de construire le modèle de classifieur (phase d'apprentissage hors ligne) ou de prédire en sortie la classe de ce vecteur (phase de classification en ligne). Cette classification multi-classes supervisée est basée sur l'algorithme d'apprentissage SVM.

2.3.4 Module de classification par SVM

Comme nous avons vu auparavant, le problème de Tri automatique des pommes de terre est résolu comme un problème de classification multi-classes supervisée des images de pommes de terre. En effet, nous choisissons la méthode à base des machines à vecteurs de support comme la base de notre classifieur. Par la suite, nous allons présenter dans cette section le fondement théorique de cette méthode.

2.3.4.1 Fondement mathématiques

En exploitant les résultats de la théorie de l'apprentissage statistique développée en grande partie par Vladimir Vapnik, la machine à vecteurs de support (SVM) est une famille d'algorithmes d'apprentissage particulièrement bien adaptée pour traiter des données de très hautes dimensions telles que les textes, les images et la voix...etc.[50].

Les SVM ont été appliqués à de très nombreux domaines, notamment dans les domaines de reconnaissance et de classification d'image [51].

✓ **Principe des machines à vecteurs de support :**

Le principe de SVM consiste à utiliser une transformation non linéaire pour redécrire les données d'apprentissage dans un espace de plus grande dimension, dans lequel elles pourront être traitées efficacement de manière linéaire. Dans le cas d'un problème de classification binaire, l'objectif est de déterminer dans le nouvel espace, un hyperplan qui permet de séparer les données d'apprentissage en deux classes de manière optimale. Dans l'exemple illustré à la figure 17, les données initialement décrites par deux caractéristiques x_1 et x_2 sont redécrites par trois nouvelles caractéristiques $z_1 = x_1^2$, $z_2 = x_2^2$, et $z_3 = x_1 \cdot x_2$ [52].

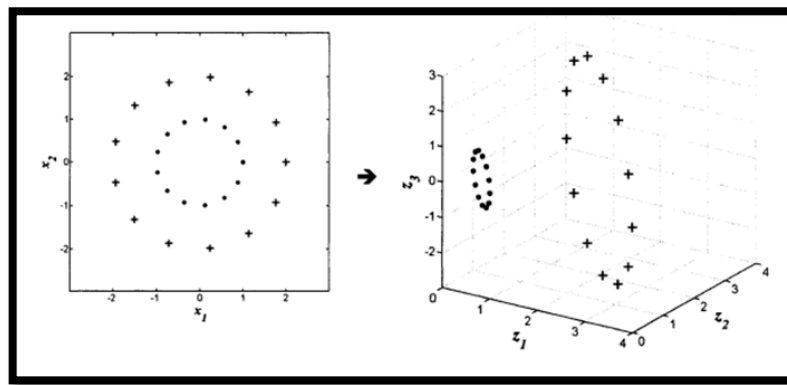


Figure 17 : description des données dans un espace de plus grande dimension[52].

L'objectif des SVM est alors de trouver un hyperplan optimal permettant de séparer les données d'apprentissage de sorte que tous les points d'une même classe soient du même côté de l'hyperplan. Donc, cet hyperplan d'un point de vue géométrique divise l'espace en deux comme illustré dans la Figure 18 [53].

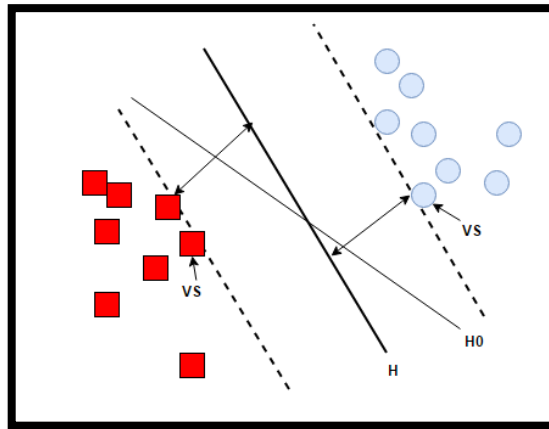


Figure 18 : Hyperplans séparateurs[53]

- H_0 est un hyperplan quelconque, H est l'hyperplan optimal et VS sont les Vecteurs Supports.

Généralement, on peut distinguer deux cas de séparation de données à savoir : cas des données linéairement séparables, et cas des données non linéairement séparables [54].

✓ Fonction noyau (kernel)

Dans le cas linéaire, on pouvait transformer les données dans un espace où la classification serait plus aisée. Dans ce cas, l'espace de redescription utilisé le plus souvent est \mathbb{R} (ensemble des nombres réels). Néanmoins, Dans des cas non linéaires, plus la dimension de l'espace de description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples et les contre exemples est élevée [54].

$$\phi: \begin{matrix} \mathbb{R}^d \rightarrow F \\ x \rightarrow \phi(x) \end{matrix} \quad (6)$$

Notons ϕ une transformation non linéaire ou mappage « mapping » de l'espace d'entrée x en un espace de redescription $\phi(x)$. Voici un exemple de projection (voir figure 19):

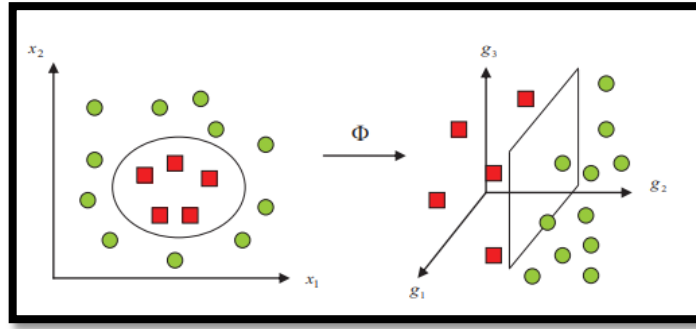


Figure 19 : Exemple de projection[53]

$$x = (x_1 \dots x_p)^T \rightarrow \theta(x) = \langle \theta_1(x) \dots \theta_n(x) \rangle \quad (7)$$

Dans ce nouvel espace de caractéristiques, la fonction objective à optimiser devient :

$$Q(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2 \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle} \quad (8)$$

Dans le calcul de l'optimum de la fonction, on utilise une astuce appelée "Noyau" ("Kernel"), au lieu de calculer $\phi(x_i)$, $\phi(x_j)$ et leur produit scalaire, on calcule plutôt une fonction $K(x_i, x_j)$ qui représente à la fois les deux transformations (qui peuvent être inconnues) et leur produit scalaire. Cette fonction permet de surmonter le problème de détermination de la transformation ϕ et permet d'apprendre des relations non linéaires par des machines linéaires. En pratique, il existe certains noyaux qui sont très utilisés et qui sont considérés comme standards [55].

Une fois le noyau choisi, la fonction objective peut être calculée comme suit:

$$Q(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2 \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)} \quad (9)$$

Et la fonction de décision devient :

$$H(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x) + b \quad (10)$$

Exemples de noyaux : [55]

- **Noyau linéaire** : Si les données sont linéairement séparables, on n'a pas besoin de changer d'espace, et le produit scalaire suffit pour définir la fonction de décision : $k(x_i, x_j) = x_i^T x_j$ (11)

- **Noyau polynomial** : Le noyau polynomial élève le produit scalaire à une puissance naturelle d :

$$k(x_i, x_j) = (x_i^T x_j)^d \quad (12)$$

Si $d = 1$ le noyau devient linéaire. Le noyau polynomial dit non homogène $k(x_i, x_j) = (x_i^T x_j + C)^d$ est aussi utilisé.

- **Noyau RBF** : Les noyaux RBF (Radial Basis fonctions) sont des noyaux qui peuvent être écrits sous la forme : $k(x_i, x_j) = f(d(x_i, x_j))$ où d est une métrique sur x et f est une fonction dans \mathbb{R} . Un exemple des noyaux RBF est le noyau Gaussien:

$$k(x_i, x_j) = e^{-\left(\|x_i - x_j\|^2 / 2\sigma^2\right)} \quad (13)$$

Où σ est un réel positif qui représente la largeur de bande du noyau.

2.3.4.2 SVM pour le cas multi-classes

Il existe plusieurs méthodes pour faire la classification multi-classes. les plus utilisées sont :

- La première méthode dite Un-contre-Un. Au lieu d'apprendre N fonctions de décisions, ici chaque classe est discriminée d'une autre.

- La deuxième méthode est appelé Un-Contre-Tous. C'est une approche étendant la notion de marge aux cas multi-classes. Cette formulation intéressante permet de poser un problème d'optimisation unique. Le problème fait intervenir N fonctions de décision [4].

✓ **Un contre un** : La technique un contre un, quant à elle, construit $N_c(N_c - 1)/2 = 2$ classifieurs SVMs binaires, en utilisant toutes les paires de combinaisons binaires des N_c classes.

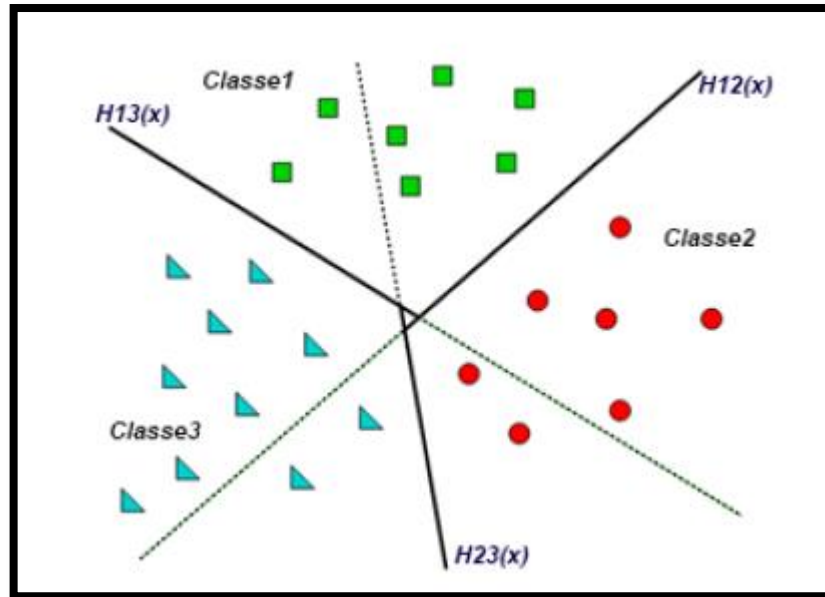


Figure 20 : Approche une-contre-une [4]

L'apprentissage de chaque classifieur SVM est réalisé en considérant les images de la première classe comme des exemples positifs et les images de la seconde classe comme des exemples négatifs. Pour combiner les différents classifieurs SVM, l'algorithme de vote majoritaire pour max-winsvoting est adapté. Cet algorithme permet de trouver la classe résultante en choisissant la classe votée par la majorité des classifieurs.

Le nombre d'images utilisées pour l'apprentissage de chaque classifieur SVM est très réduit, car seules les images appartenant à deux classes parmi les N_c classes sont prises en considération. D'où, un temps d'apprentissage plus court.

L'inconvénient de cette méthode réside dans le fait que chaque image requête doit être présentée à un grand nombre de classifieurs $N_c(N_c - 1)/2 = 2$. Il en résulte un ralentissement de la phase de généralisation, en particulier lorsque le nombre de classes est considérable.

✓ **Un contre tous:** Un problème de classification de N_c classes ($N_c > 2$) est décomposé selon la technique un contre tous en N_c classifications binaires. Pour cela, N_c classifieur SVMs binaires sont construits.

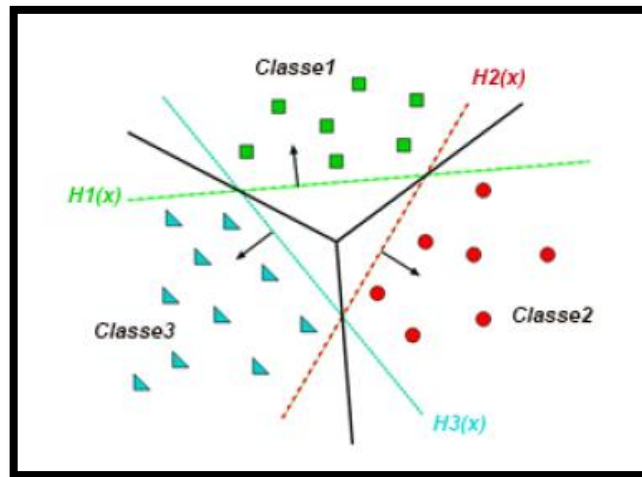


Figure 21 : Approche une-contre-reste [4]

L'apprentissage du i ème classifieur SVM est réalisé de façon que les images de la i ème classe soient considérées négatives. En phase de généralisation, une image requête est présentée à l'ensemble des N_c classifieurs SVM. Elle est étiquetée ainsi en fonction du rendement maximum parmi les N_c classifieurs.

L'inconvénient de cette méthode réside dans sa complexité d'apprentissage, vu que le nombre d'échantillons d'apprentissage devient très important tous les échantillons disponibles .

3. Conclusion

Dans ce chapitre, nous avons fourni la description générale du système de tri de pomme de terre que nous proposons ainsi que son architecture. Ceci en présentant brièvement les deux phases de fonctionnement à savoir : La phase d'entraînement et de classification et en expliquant de façon détaillée la composition de différents modules à savoir : prétraitement, extraction de caractéristiques et d'entraînement de classifieur où la méthode de classification adoptée c'est le SVM.

Dans le chapitre suivant, nous allons mettre en œuvre les différents modules décrits au-dessus pour faire fonctionner notre système de tri basé sur les caractéristiques extraites, ainsi que nous utiliserons les mesures de performance pour les méthodes de classification et afficherons les résultats de la classification en termes de taux de reconnaissance.

Chapitre 3

Implémentation et évaluation des performances

1. Introduction

Une fois que les résultats de l'étape de conception soient disponibles, l'étape de l'implémentation d'un logiciel est devenue indispensable dont son but principal est de réaliser un produit capable de résoudre les problèmes en question en utilisant des algorithmes ainsi que des outils et langages de programmation particulier pour les implémenter.

Après que nous avons donné dans le précédent chapitre la conception de façon globale et détaillée. Dans ce chapitre nous allons présenter en premier temps, l'environnement du développement avec la bibliothèque utilisée. Ensuite, on va présenter les algorithmes utilisés illustrés par quelques résultats obtenus tout en terminant ce chapitre par une conclusion.

2. Environnement du développement

2.1 Description de langage de programmation « python »

Python est un langage de programmation interprété, orienté objet et de haut niveau avec une sémantique dynamique. Ses structures de données intégrées de haut niveau, combinées à un typage dynamique et à une liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour l'utiliser comme un langage de script ou de la colle pour connecter les composants existants ensemble. Python est simple, facile à apprendre la syntaxe et met l'accent sur la lisibilité. Python prend en charge les modules et les packages, ce qui encourage la modularité du programme et la réutilisation du code. L'interpréteur Python et la vaste bibliothèque standard, et notamment fournie pour la machine Learning, sont disponibles gratuitement sous forme binaire ou source pour toutes les grandes plates-formes et peuvent être distribués librement. [56]



2.2 Description de l'environnement de développement « Jupyter »

Jupyter Notebook est une application Web à source ouverte qui permet à un développeur de créer et de partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif. Les utilisations incluent : nettoyage et transformation de données, simulation numérique, modélisation statistique, visualisation de données, apprentissage automatique, etc. [57].



2.3 Description de la bibliothèque « scikit-image »

scikit-image est une collection d'algorithmes pour le traitement d'images. Il est disponible gratuitement et sans restriction. Il est écrit en Python sous forme de paquet (package) d'où le paquet est importé comme « skimage ». La plupart des fonctions de « skimage » se trouvent dans les sous-modules [58].



2.4 Description de la bibliothèque « scikit-learn »

Pour la mise en œuvre des fondamentaux de machine Learning en python, la bibliothèque « scikit-learn » fournit des fonctions permettant d'implémenter les différents concepts, entre autres : méthode de classification supervisée et régression, dataset, validation de performance, apprentissage et sélection de modèle[59].



3. Évaluer les modèles

L'évaluation des modèles est le processus servant à comprendre comment ils fournissent la classification correct, puis à mesurer la valeur d'une prédiction dans un certain contexte. Dans cette section nous allons aborder les sujets autour l'outil de base pour l'évaluation des modèles, les mesures de performance et la validation croisée [60]. En Python, Le module « *sklearn.metrics* » implémente plusieurs fonctions de perte, de score et d'utilitaire pour mesurer les performances de classification [61].

- **La matrice de confusion (confusion matrix)**

La matrice (ou table) de confusion est un tableau dont les lignes et les colonnes représente les prédictions et les résultats réels (les étiquettes) pour un classifieur. Elle est utilisée pour comprendre le fonctionnement du classifieur en fonction de la réponse correcte [60].

- Dans le cas d'une méthode de classification binaire : Dans une tâche de classification binaire, les termes «positif» et «négatif» font référence à la prédiction du classificateur, et les termes «vrai» et «faux» indiquent si cette prédiction correspond au jugement externe (parfois appelée «observation»). Étant donné ces définitions, nous pouvons formuler le tableau 1 en fonction des 4 catégories de prédiction d'un classifieur binaire:

Prédiction Réalité	Positif	Négatif
Positif (P)	Vrai Positif (VP)	Faux Négatif (FN)
Négatif(N)	Faux Positif (FP)	Vrai Négatif (VN)

Tableau 1 : Matrice de confusion pour la classification

- Le tableau 2 c'est une généralisation au problème multi-classe

Prédiction \ Réalité	Classe 1	Classe 2	Classe 3	Classe n
Classe 1	N11	N12	N13	N1n
Classe 2	N21	N22	N23	N2n
Classe 3	N31	N32	N33	N3n
Classe n	Nn1	Nn2	Nn3	Nnn

Tableau 2 : Matrice de confusion pour la classification multi-classe

En Python, la fonction ***confusion_matrix*** du module « ***sklearn.metrics*** » permet de calculer la matrice de confusion [61].

Cette matrice est utilisée pour calculer les mesures de performances. Par la suite nous allons définir les plus utilisées.

3.1. Mesures (métriques) de performance

3.1.1. Taux d'exactitude de prédiction « reconnaissance » (accuracy)

Permet de mesurer la qualité du modèle en termes de score d'exactitude. Dans le cas d'un classifieur binaire, le taux d'exactitude de prédiction « reconnaissance » (accuracy):

$$t_r = \frac{VP + VN}{P + N}$$

Tandis que pour un modèle de classification multi classes le taux d'exactitude est [60] :

$$t_r = \frac{\sum_i n_{i,i}}{\sum_{i,j} n_{i,j}}$$

En Python, la fonction ***accuracy_score*** du module « ***sklearn.metrics*** » permet de calculer l'accuracy. « *from sklearn.metrics import accuracy_score* » [61].

3.1.2. Précision

La précision est la capacité du classificateur à ne pas étiqueter comme positif un échantillon négatif. Dans le cas d'un classifieur binaire.

$$precision = \frac{vp}{vp + fp}$$

3.1.3. Recall : rappel

Le rappel est la capacité du classificateur de trouver tous les échantillons positifs. Dans le cas d'un classifieur binaire.

$$recall = \frac{vp}{vp + fn}$$

3.1.4. F1score

Le F1score peut être interprété comme une moyenne pondérée de la précision et du rappel, où un score de F1 atteint sa meilleure valeur à 1 et plus mauvais score à 0. La contribution relative de la précision et du rappel au F1score est égale. La formule pour le F1score est la suivante :

$$F1 = 2 * (Précision * recall) / (precision + recall)$$

Dans le cas multi-classe et multi-label, il s'agit de la moyenne du F1score de chaque classe avec pondération en fonction du paramètre moyen.

En Python, la fonction « *classification_report* » du module « *sklearn.metrics* » permet de calculer F1score , Précision et recall : rappel [61].

3.2. Validation croisée

Dans le but d'évaluer la capacité à généraliser, plusieurs problèmes ont faits face ce processus. L'un de ces problèmes c'est le risque de sur-apprentissage. Pour remédier à ce problème, on a besoin d'un jeu d'exemples et de test indépendant. Cependant, ce besoin est loin d'être satisfait dans certaines situations. Ceci vu au nombre d'échantillons limité. La validation croisée fournit une réponse à ces deux problèmes. La validation croisée est un ensemble de méthodes ayant pour but soit de calibrer les paramètres d'un modèle, soit d'estimer sa fiabilité et sa crédibilité. En se focalisant sur

un processus de ré-échantillonnage répétitif, elle permet d'utiliser l'intégralité de jeu de données pour l'entraînement et pour la validation. Elle est basée sur des techniques d'échantillonnage, dont les plus courantes sont [62]:

3.2.1. k-fold cross-validation

Elle consiste à diviser l'ensemble des données d'apprentissage en k partitions distinctes de taille égales pour utiliser $k - 1$ partitions pour l'apprentissage et la dernière pour évaluer la performance du modèle. Ce processus est répété k fois pour chacun des k segments disponibles pour le test. L'erreur de test finale est la moyenne sur les tests croisés des k estimations possibles, elle est aussi appelée erreur de généralisation. A chaque test de validation, les paramètres du modèle changent suivant une grille de recherche prédéfinie, appelée « *Grid search method* ». Les paramètres qui ont obtenu la plus faible erreur de généralisation en moyenne seront sauvegardés pour effectuer l'apprentissage final sur les données de l'échantillon test [62].

3.2.2. Leave-one-out cross-validation

Il s'agit d'un cas particulier de la méthode k -folds où $k = n$ (taille de la base d'apprentissage). On apprend donc sur $n - 1$ observation puis on valide le modèle sur la n ième observation, et l'on répète cette opération n fois [62].

4. Détails de l'implémentation

Dans ce qui suit on va présenter les différents modules de notre système, en spécifiant leurs implémentations, tout en illustrant quelques détails sur les différents algorithmes implémentés. Notre système peut être divisé en trois sous-systèmes : Extraction des caractéristiques, Apprentissage et Classification (utilisation).

4.1. Dataset

Les jeux de données (dataset) est un élément essentiel pour fonctionner n'importe quel système de reconnaissance (tri). Ces dataset sont constitué de deux façons à savoir : **1)** collectées les images que les compose à partir de l'Internet (Google image, Flickr ...). **2)** ou constitués par des images acquises via des plateformes d'acquisition implantées dans les champs d'application en question. De plus, ce dataset doit être fractionnée en deux parties à savoir des données d'entraînement et de test.

Dans notre cas et en raison de l'absence d'un dataset standard (benchmark), nous avons constitué notre jeu de données par des images couleurs des pommes de terre acquises par un Smartphone. Comme montrer le tableau 3, notre base d'images est constituée de 488 images de 04 catégories des pommes de terre (bartina, bartina avec maladie, Spunta et Spunta avec maladie).





Catégorie (classe)	Nombre	Image
Bartina	180	
bartina avec maladie	80	
Spunta	130	
Spunta avec maladie	70	
Totale	460	

Tableau 3 : Exemple de chaque catégorie des pommes de terre

4.2. Extraction de caractéristique

Dans l'application que nous avons développée, nous avons instigué plusieurs types de caractéristiques. Pour cela nous avons implémenté plusieurs algorithmes d'extraction des caractéristiques sous python (voir l'annexe) par l'importation de la bibliothèque "*skimage.feature*". Ces algorithmes traitent en entrée l'ensemble des images de la pomme de terre, ensuite ils enregistrent les valeurs ainsi obtenus (vecteurs de caractéristiques) dans la base de données de fichier csv. Le contenu de ce fichier est

représenté sous forme un pair (X,'target(classe)') où X=['feature1','feature2','feature3','feature4',.....'feature K'] est le vecteur de caractéristiques.

Algorithme 1 : Algorithme d'extraction des caractéristiques

Entrée : ensemble d'images.

Sortie : vecteur de caractéristique ;

Début

Pour i = 1::n, **faire**

 Prétraitement(i) ;

 Calculer le Motif Binaire Local (LBP) (i) ;

 Calculer le Histogramme des Gradients Orientés (HOG) (i) ;

 Calculer histogramme de couleurs(i) ;

 Enregistrer dans un vecteur () ;

FPour

Combinaison les 3 vecteurs dans un seul vecteur ;

Enregistrer le vecteur dans la base de donnée ;

Fin**4.2.1. Extraction de caractéristique de texture**

Pour extraire les caractéristiques globales de texture, nous avons utilisé la méthode LBP. Notre choix est motivé par le fait qu'elle est la méthode la plus utilisée dans la littérature notamment dans le contexte de classification d'image. Pratiquement, on a importé la fonction « *local_binary_pattern* » du module « *skimage.feature* ». Cette méthode accepte en entrée des images en niveau de gris (l'ensemble des de notre dataset converti) avec les paramètres suivants : nombre de voisins est égale à « 8 », le rayon de cercle (résolution spatiale de l'opérateur) est égale à « 1 » et la méthode pour déterminer le motif est 'uniform'.

Algorithme 2 : calcul de motif binaire local (LBP)**Entrée** : image au niveau de gris.**Sortie** : vecteur de caractéristique ;**Début**

Prétraitement (image) ;

Patterns = calcule lbp (image, nb_voisin, rayon, méthode de déterminer le motif)

Calculer les histogrammes à partir des motifs

Former un tableau a partir les histogrammes calculés

Ajouter l'étiquette cible (target) à la fin du tableau

Enregistrer dans un vecteur () ;

Enregistrer les vecteurs dans la base de donnée « fichier (.csv) »;

Fin

Les vecteurs de caractéristiques ainsi obtenus (contenu du fichier « feature_dataLBP.csv ») sont illustrés dans la figure au-dessous.

importing train data ..

```
In [2]: feature_file = './feature_dataLBP.csv'
data = pd.read_csv(feature_file)
data.head(10)
```

Out[2]:

	lbp_gray_0	lbp_gray_1	lbp_gray_2	lbp_gray_3	lbp_gray_4	lbp_gray_5	lbp_gray_6	lbp_gray_7	lbp_gray_8	lbp_gray_9	Class
0	0.036192	0.079117	0.040090	0.095207	0.148613	0.153399	0.089471	0.092036	0.136883	0.128991	0
1	0.068625	0.107219	0.039381	0.062288	0.081580	0.084877	0.071628	0.119988	0.155105	0.209309	3
2	0.091766	0.107441	0.047219	0.062745	0.071471	0.073982	0.070535	0.114817	0.151387	0.208637	1
3	0.097183	0.109483	0.049465	0.057598	0.060805	0.065940	0.068288	0.113784	0.152438	0.225015	0
4	0.079075	0.104601	0.042637	0.062270	0.080372	0.080637	0.071285	0.116907	0.154162	0.208054	3
5	0.044198	0.079339	0.043015	0.081718	0.121393	0.137622	0.077135	0.089778	0.195279	0.130523	0
6	0.075856	0.105922	0.046787	0.068258	0.084186	0.083508	0.073405	0.113387	0.144601	0.204090	0
7	0.092529	0.110036	0.048162	0.059676	0.063700	0.067417	0.068480	0.115712	0.153105	0.221183	0
8	0.072973	0.102012	0.041598	0.062288	0.078799	0.085417	0.074745	0.114895	0.157700	0.209574	3
9	0.070282	0.108769	0.039994	0.060883	0.081880	0.083526	0.070330	0.119237	0.153526	0.211574	3

Figure 22 : aperçu de fichier « feature_dataLBP.csv »

4.2.2. Extraction de caractéristique HOG

Aussi, nous avons utilisé la méthode HOG. Cette méthode permet d'extraire des caractéristiques comme descripteur d'apparence. La méthode « **HOG_Q** » importée du module « **skimage.feature** » a été utilisée. Nous avons testé les paramètres suivant : le nombre d'orientations du gradient : 8, la taille de cellule en pixels (pixels_per_cell) : 32, la taille du bloc est en cellules (cells_per_block) : 4, et la méthode normalisation est : L2 dont les résultats sont : la taille de vecteur descripteur de l'histogramme du gradient orienté « HOG » est (460 ,10882), le temps d'extraction est 143.763 ms

Algorithme 3: calcule un histogramme de gradient (HOG)

Entrée : image au niveau de gris.

Sortie : vecteur de caractéristique ;

Début

Prétraitement (image) ;

Calculer l'image de gradient dans X et Y

Construction de l'histogramme de l'orientation des gradients

Normalisation de descripteur (formation des blocs)

Former un tableau à partir les histogrammes calculés

Ajouter l'étiquette cible (target) à la fin du tableau

Enregistrer dans un vecteur ;

Enregistrer les vecteurs dans la base de donnée fichier (.csv);

Fin

Le contenu du fichier « feature_dataHOG .csv » est montré dans la figure 23.

```
In [2]: feature_file = './feature_dataHOG.csv'
data = pd.read_csv(feature_file)
data.head(10882)
```

his4	his5	his6	his7	his8	his9	...	his10872	his10873	his10874	his10875	his10876	his10877	his10878	his10879	his10880	Class
218	0.006766	0.012561	0.014422	0.028185	0.002196	...	248.37500	249.203125	250.031250	250.859375	251.687500	252.515625	253.343750	254.171875	255.0	0
108	0.007396	0.027401	0.009654	0.029293	0.009735	...	248.28125	249.121094	249.960938	250.800781	251.640625	252.480469	253.320312	254.160156	255.0	2
175	0.030157	0.058845	0.032165	0.062400	0.023834	...	210.00000	210.750000	211.500000	212.250000	213.000000	213.750000	214.500000	215.250000	216.0	3
700	0.035198	0.052434	0.025654	0.041198	0.023315	...	197.12500	197.734375	198.343750	198.953125	199.562500	200.171875	200.781250	201.390625	202.0	3
590	0.003036	0.009835	0.006285	0.013824	0.001595	...	248.21875	249.066406	249.914062	250.761719	251.609375	252.457031	253.304688	254.152344	255.0	0
280	0.005908	0.022086	0.014655	0.027379	0.004964	...	212.93750	213.570312	214.203125	214.835938	215.468750	216.101562	216.734375	217.367188	218.0	1
543	0.004957	0.013984	0.005619	0.016618	0.004472	...	232.37500	233.203125	234.031250	234.859375	235.687500	236.515625	237.343750	238.171875	239.0	2
671	0.003611	0.014440	0.017234	0.031058	0.002862	...	247.65625	248.574219	249.492188	250.410156	251.328125	252.246094	253.164062	254.082031	255.0	0
759	0.004181	0.019327	0.018816	0.029587	0.003506	...	248.34375	249.175781	250.007812	250.839844	251.671875	252.503906	253.335938	254.167969	255.0	0
605	0.011487	0.032144	0.019712	0.063232	0.020820	...	215.31250	216.023438	216.734375	217.445312	218.156250	218.867188	219.578125	220.289062	221.0	1
116	0.003700	0.013466	0.000000	0.017738	0.001824	...	248.06250	248.920625	249.766875	250.614062	251.461250	252.308438	253.155625	254.002812	255.0	2

Figure 23 : aperçu de fichier « feature_dataHOG .csv »

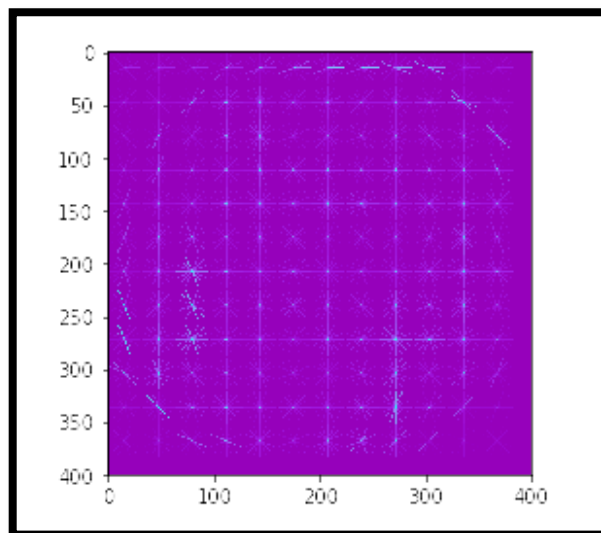


Figure 24 : l'histogramme d'orientation de gradient résultant

4.2.3. Extraction de caractéristique de couleur

Dans un troisième point, nous avons utilisé l'histogramme de couleur comme un descripteur de caractéristique globale de couleur. Pour cela nous avons utilisé la fonction « **calcHist** » de la bibliothèque « **OpenCV** » pour calculer la classe d'histogramme pour chaque couleur (chaque canal). Nous avons testé les paramètres suivants : liste de canaux (channels) de couleurs, la taille des histogrammes pour chaque dimension.

Algorithme 4: calcule de l'Histogramme de Couleur

Entrée : image couleur.

Sortie : vecteur de caractéristique ;

Début

Prétraitement (image) ;

Transformer l'espace de couleur d'image « RGB » à l'espace de couleur HSV

Calcul de l'histogramme à partir l'image pour chaque couleur (canal)

Normalisation d'histogrammes calculés

Former un tableau à partir les histogrammes normalisés

Ajouter l'étiquette cible (target) à la fin du tableau

Enregistrer dans un vecteur () ;

Enregistrer les vecteurs dans la base de donnée fichier (.csv);

Fin

Après un temps d'extraction égale à 49.563, nous avons obtenu un vecteur de caractéristiques de taille (460, 513). Les résultats (contenu du fichier « feature_dataHCOLOR.csv ») sont montrés dans la figure 25.

```
feature_file = './feature_dataHCOLOR.csv'
data = pd.read_csv(feature_file)
data.head(513)
```

	his2	his3	his4	his5	his6	his7	his8	his9	...	his503	his504	his505	his506	his507	his508	his509	his510	his511	Class
05242	0.013304	0.020657	0.036284	0.045522	0.031595	0.000000	0.000014	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
00063	0.002068	0.004980	0.027767	0.021894	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3
00000	0.000000	0.000000	0.000000	0.001067	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
00000	0.000000	0.000000	0.000376	0.005394	0.000753	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
00000	0.000070	0.003235	0.010348	0.031009	0.057653	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
02182	0.004940	0.004254	0.024797	0.001015	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3
03386	0.009280	0.005893	0.012406	0.001482	0.000000	0.000000	0.018429	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
02906	0.078139	0.036361	0.012681	0.002179	0.000000	0.000000	0.000025	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3
01190	0.038947	0.001281	0.014544	0.001695	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3
00000	0.000000	0.000000	0.018677	0.082652	0.000981	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
07356	0.018022	0.004900	0.038988	0.001003	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3
01199	0.018115	0.003932	0.016026	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3
00000	0.000000	0.000000	0.001537	0.005256	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

Figure 25 : aperçu de fichier « feature_dataHCOLOR.csv »

4.2.4. Combinaison des vecteurs des caractéristiques

Finalement, nous avons investigué la combinaison des caractéristiques globales. Pour cela, nous avons fusionné, en un seul vecteur, les caractéristiques susmentionnées (LBP+HOG +histogramme de couleur). Après la concaténation des vecteurs de caractéristiques en un seul vecteur, on doit l'enregistrer dans la base de données. Ensuite, cette base sera utilisée pour entraîner notre modèle de décision.

Algorithme 5 : Combinaison des caractéristiques

Entrée : images couleurs.

Sortie : vecteur de caractéristiques concaténé ;

Début

Pour $i = 1::n$, **faire**

 Prétraitement(i) ;

 Extraire la caractéristique de texture(LBP) ;

 Enregistrer dans un vecteur () ;

 Extraire la caractéristique de couleur (histogramme de couleur) ;

 Enregistrer dans un vecteur () ;

Extraire la caractéristique de texture(HOG) ;

Enregistrer dans un vecteur () ;

FPour

Combinaison les 3 vecteurs dans un seul vecteur ;

Normaliser le vecteur obtenu

Ajouter dans un vecteur la classe désirée (target) ;

Enregistrer les vecteurs dans la base de donnée ;

Fin

Après l'expérimentation, nous avons obtenu un vecteur de taille (460, 11403) dans un temps d'extraction égale à 130.801 ms .

4.3. Entraînement et test

Comme nous avons vu au chapitre précédent, notre système repose sur un classifieur. La construction de ce dernier nécessite une phase d'apprentissage hors ligne. En se basant sur un jeu de données (d'entraînement et de test) l'objectif est d'obtenir un modèle de décision tout en prend en entrée les fichiers (.csv) résultants de l'étape de l'extraction des caractéristiques. Dans notre système, nous avons utilisé les machines de vecteurs de support (SVM) pour entraîner le modèle de décision de notre classifieur. Pour l'implémenter nous avons appliqué la méthode **svm.SVC()** avec plusieurs configurations de paramètres. Cette méthode est implémentée dans le module « **sklearn.svm** » de la librairie « **sklearn** ».

Dans le but de préparer les jeux de données, nous avons décomposé notre dataset (les fichiers (.csv) obtenus dans la phase d'extraction de caractéristiques) en deux parties (d'entraînement et de test). Cette décomposition a été faite selon deux méthodes : 1) décompositions manuelles dont la partie d'entraînement est constitué de 322 et celle de test est constitué de 138 (30%). Pour cela, nous avons appliqué la fonction helper « **train_test_split** » dans « **scikit-learn** », qui permet de calculer rapidement un fractionnement aléatoire en jeux d'entraînement et en tests. 2)

décompositions par validation croisée en utilisant la technique k-fold (k=10 dans notre expérimentation).

4.4. Evaluation de performance

Dans le souci d'évaluer les performances de notre système, nous avons effectué plusieurs scénarios de test. Pour l'effectuer on a implémenté un code (voir l'annexe) pour chaque scénario (caractéristiques utilisées et méthode d'évaluation).

4.4.1. Résultats

4.4.1.1. Descripteur LBP

Premièrement, nous avons testé la caractéristique de texture avec le descripteur LBP. La méthode LBP accepte de nombreux paramètres, les principaux étant : nombre de voisins est égale à « 8 » et le centre est égal à « 1 ».

- Dans un premier temps, nous avons choisi aléatoirement 30% instances parmi les 460 exemples d'apprentissage dont l'entraînement est 322 et le reste pour les tests (138). Nous avons utilisé les paramètres suivants : Type SVM : C-SVC , type de Noyau : RBF , gamma= auto, degré= 3 pour plusieurs valeurs de « C » comme suit :

✓ **C = 3**

Les résultats ainsi obtenus sont montrés dans les tableaux 4 et 5 où le taux de reconnaissance obtenu est : 76%

	Précision	Recall	F1-score
Bartina	0.68	0.94	0.79
MaladieBartina	1.00	0.15	0.26
MaladieSpunta	0.86	0.79	0.83
Spunta	0.82	0.94	0.88

Tableau 4 : mesures de performance pour classification multi classes avec LBP et C=3

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	49	0	1	2
MaladieBartina	21	4	2	0
MaladieSpunta	0	0	19	5
Spunta	2	0	0	33

Tableau 5 : Matrice de confusion pour classification multi classes avec LBP et C=3

✓ $C = 0.001$

Les résultats ainsi obtenus sont montrés dans les tableaux 6 et 7 où le taux de reconnaissance obtenu est: 37%

	Précision	Recall	F1-score
Bartina	0.38	1.00	0.55
MaladieBartina	0.00	0.00	0.00
MaladieSpunta	0.00	0.00	0.00
Spunta	0.00	0.00	0.00

Tableau 6 Mesures de performance pour classification multi classes avec LBP et C=0.001

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	52	0	0	0
MaladieBartina	27	0	0	0
MaladieSpunta	24	0	0	0
Spunta	35	0	0	0

Tableau 7 : Matrice de confusion pour classification multi classes avec LBP et C=0.001

✓ $C = 0.1$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 71%

	Précision	Recall	F1-score
Bartina	0.60	0.98	0.74
MaladieBartina	0.00	0.00	0.00
MaladieSpunta	0.93	0.58	0.72
Spunta	0.87	0.94	0.90

Tableau 8 : Mesures de performance pour classification multi classes avec LBP et $C=0.1$

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	51	0	1	0
MaladieBartina	27	0	0	0
MaladieSpunta	5	0	14	5
Spunta	2	0	0	33

Tableau 9 : Matrice de confusion pour classification multi classes avec LBP et $C=0.1$

✓ $C = 1$

Les résultats ainsi obtenus sont montrés dans les tableaux 10 et 11 où le taux de reconnaissance obtenu est : 74%

	Précision	Recall	F1-score
Bartina	0.66	1.94	0.78
MaladieBartina	1.00	0.04	0.07
MaladieSpunta	0.82	0.75	0.78
Spunta	0.83	0.97	0.89

Tableau 10 : Mesures de performance pour classification multi classes avec LBP et $C=1$

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	49	0	1	2
MaladieBartina	23	1	3	0
MaladieSpunta	1	0	18	5
Spunta	1	0	0	34

Tableau 11 : Matrice de confusion pour classification multi classes avec LBP et C=1

✓ $C = 2$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 75%

	Précision	Recall	F1-score
Bartina	0.66	1.96	0.78
MaladieBartina	1.00	0.04	0.07
MaladieSpunta	0.86	0.75	0.80
Spunta	0.85	0.97	0.91

Tableau 12 : Mesures de performance pour classification multi classes avec LBP et C=2

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	50	0	1	1
MaladieBartina	24	1	2	0
MaladieSpunta	1	0	18	5
Spunta	1	0	0	34

Tableau 13 : Matrice de confusion pour classification multi classes avec LBP et C=2

✓ $C = 4$

Les résultats ainsi obtenus sont montrés dans les tableaux 14 et 15 où le taux de reconnaissance obtenu est : 80%

	Précision	Recall	F1-score
Bartina	0.73	1.94	0.82
MaladieBartina	1.00	0.30	0.46
MaladieSpunta	0.90	0.79	0.84
Spunta	0.83	1.00	0.91

Tableau 14 : Mesures de performance pour classification multi classes avec LBP et $C=4$

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	49	0	1	2
MaladieBartina	18	8	1	0
MaladieSpunta	0	0	19	5
Spunta	0	0	0	35

Tableau 15 : Matrice de confusion pour classification multi classes avec LBP et $C=4$

✓ $C = 5$

Les résultats ainsi obtenus sont montrés dans les tableaux 16 et 17 où le taux de reconnaissance obtenu est : 79%

	Précision	Recall	F1-score
Bartina	0.73	1.9	0.81
MaladieBartina	0.82	0.33	0.47
MaladieSpunta	0.90	0.79	0.84
Spunta	0.83	1.00	0.91

Tableau 16 : Mesures de performance pour classification multi classes avec LBP et $C=5$

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	47	2	1	2
MaladieBartina	17	9	1	0
MaladieSpunta	0	0	19	5
Spunta	0	0	0	35

Tableau 17 : Matrice de confusion pour classification multi classes avec LBP et C=5

- Dans un second temps, nous avons effectué des tests avec la validation croisée en utilisant le K-fold (K=10). Nous avons utilisé les paramètres suivants : Type SVM : C-SVC , type de Noyau : RBF , gamma= auto, degré= 3 pour plusieurs valeurs de « C » comme suit :

✓ **C = 3**

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyen obtenu est : 75%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,74	0,74	0,83	0,7	0,76	0,76	0,65	0,78	0,76	0,78

Tableau 18 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=3

✓ **C = 0.001**

Les résultats ainsi obtenus sont montrés dans le tableau 19 où le taux de reconnaissance moyenne obtenu est : 39%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39

Tableau 19 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=0.001

✓ $C = 0,1$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 72%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,72	0,67	0,74	0,72	0,69	0,78	0,65	0,71	0,76	0,72

Tableau 20 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=0.1

✓ $C = 1$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 74%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,72	0,67	0,78	0,72	0,74	0,76	0,65	0,74	0,78	0,78

Tableau 21 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=1

✓ $C = 2$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 75%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,72	0,72	0,78	0,72	0,74	0,78	0,65	0,74	0,76	0,78

Tableau 22 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=2

✓ $C = 4$

Les résultats ainsi obtenus sont montrés dans le tableau 23 où le taux de reconnaissance moyenne obtenu est : 77%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,76	0,76	0,84	0,74	0,78	0,78	0,65	0,85	0,74	0,80

Tableau 23 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=4

✓ $C = 5$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 77%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,76	0,76	0,80	0,76	0,78	0,78	0,65	0,87	0,74	0,83

Tableau 24 : l'accuracy pour classification multi classes avec LBP , validation croisée K-fold (K=10) et C=5

- Pour déterminer les paramètres C (régularisation de la marge), les paramètres du noyau (gamma, degré ou coef) les plus appropriés, nous avons utilisé une recherche (non-exhaustive) dans une grille discrète de valeurs possibles des paramètres c.-à-d. la stratégie de Grille de recherche (grid-search). Les différentes valeurs de paramètres testés sont comme suit : $C=\{0,001; 0,1; 1; 2; 3; 4; 5\}$. noyau (kernel) ={'RBF', 'linear'}.

Les résultats optimaux ainsi obtenus sont : $C=5$, noyau= 'RBF', où le best score est 75 % ; le taux de reconnaissance obtenu est : 82 %.

4.4.1.2. Descripteur HOG

Deuxièmement, Nous avons essayé la méthode d'histogramme des gradients (HOG) qui permet d'extraire des caractéristiques de chaque image sous forme d'histogrammes sur lesquels on applique ensuite notre classifieur SVM (RBF).

La méthode HOG accepte aussi de nombreux paramètres, les principaux étant :
 Taille de l'image : 1500x1000, Taille de cellule pour HOG : 8 , Nombre d'intervalle : 10, le nombre d'orientations regardées (en testant 2, 4, 6, 8, 9, 12, 16, la meilleure valeur semble 9) et le nombre de petits morceaux d'images à regarder, ou leur taille (pixels_per_cell dans skimage.feature.HOG). En testant des patches de tailles (7, 7), (14, 14) ou (28, 28) (qui remplissent entièrement l'image de taille 28, 28), ou d'autres tailles, la meilleure valeur semble (14, 14), ce qui correspond à $2 \times 2 = 4$ histogrammes dans chaque image :

- Dans un premier temps, nous avons choisi aléatoirement 30% instances parmi les 460 exemples d'apprentissage. Nous avons utilisé les paramètres suivants : Type SVM : C-SVC , type de Noyau : RBF , gamma= auto, degree= 3 pour plusieurs valeurs de « C » comme suit :

✓ $C = 3$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 75%

	Précision	Recall	F1-score
Bartina	0.71	0.87	0.78
MaladieBartina	0.61	0.44	0.51
MaladieSpunta	0.94	0.70	0.80
Spunta	0.89	0.92	0.91

Tableau 25 : Mesures de performance pour classification multi classes avec HOG et C=3

	Bartina	MaladieBartina	MaladieSpunta	Spunta
bartina	47	2	1	2
MaladieBartina	17	9	1	0
MaladieSpunta	0	0	19	5
Spunta	0	0	0	35

Tableau 26 : Matrice de confusion pour classification multi classes avec HOG et C=3

✓ $C = 0.001$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est: 38%

	Précision	Recall	F1-score
Bartina	0.38	1.00	0.55
MaladieBartina	0.00	0.00	0.00
MaladieSpunta	0.00	0.00	0.00
Spunta	0.00	0.00	0.00

Tableau 27 : Mesures de performance pour classification multi classes avec HOG et C=0.001

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	53	0	0	0
MaladieBartina	25	0	0	0
MaladieSpunta	23	0	0	0
Spunta	37	0	0	0

Tableau 28 : Matrice de confusion pour classification multi classes avec HOG et C=0.001

✓ $C = 0.1$

Les résultats ainsi obtenus sont montrés dans les tableaux 29 et 30 où le taux de reconnaissance obtenu est : 39%

	Précision	Recall	F1-score
Bartina	0.39	1.00	0.56
MaladieBartina	0.00	0.00	0.00
MaladieSpunta	0.00	0.00	0.00
Spunta	1.00	0.03	0.05

Tableau 29 : Mesures de performance pour classification multi classes avec HOG et C=0.1

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	53	0	0	0
MaladieBartina	25	0	0	0
MaladieSpunta	23	0	0	0
Spunta	36	0	0	1

Tableau 30 : Matrice de confusion pour classification multi classes avec HOG et C=0.1

✓ $C = 1$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 75%

	Précision	Recall	F1-score
Bartina	0.64	0.96	0.77
MaladieBartina	0.86	0.24	0.38
MaladieSpunta	1.00	0.57	0.72
Spunta	0.89	0.92	0.91

Tableau 31 : Mesures de performance pour classification multi classes avec HOG et C=1

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	51	1	0	1
MaladieBartina	18	6	0	1
MaladieSpunta	8	0	13	2
Spunta	3	0	0	34

Tableau 32 : Matrice de confusion pour classification multi classes avec HOG et C=1

✓ $C = 2$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 77%

	Précision	Recall	F1-score
Bartina	0.70	0.87	0.77
MaladieBartina	0.61	0.44	0.51
MaladieSpunta	0.94	0.65	0.77
Spunta	0.89	0.92	0.91

Tableau 33 : Mesures de performance pour classification multi classes avec HOG et C=2

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	46	6	0	1
MaladieBartina	13	11	0	1
MaladieSpunta	5	1	15	2
Spunta	2	0	1	34

Tableau 34 : Matrice de confusion pour classification multi classes avec HOG et C=2

✓ $C = 4$

Les résultats ainsi obtenus sont montrés dans les tableaux 35 et 36 où le taux de reconnaissance obtenu est : 78%

	Précision	Recall	F1-score
Bartina	0.71	0.87	0.78
MaladieBartina	0.61	0.44	0.51
MaladieSpunta	0.94	0.70	0.80
Spunta	0.89	0.92	0.91

Tableau 35 : Mesures de performance pour classification multi classes avec HOG et C=4

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	46	6	0	1
MaladieBartina	13	11	0	1
MaladieSpunta	4	1	16	2
Spunta	2	0	1	34

Tableau 36 : Matrice de confusion pour classification multi classes avec HOG et C=4

✓ $C = 5$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 78%

	Précision	Recall	F1-score
Bartina	0.71	0.87	0.78
MaladieBartina	0.61	0.44	0.51
MaladieSpunta	0.94	0.70	0.80
Spunta	0.89	0.92	0.91

Tableau 37 : Mesures de performance pour classification multi classes avec HOG et C=5

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	46	6	0	1
MaladieBartina	13	11	0	1
MaladieSpunta	4	1	16	2
Spunta	2	0	1	34

Tableau 38 : Matrice de confusion pour classification multi classes avec HOG et C=5

- Dans un second temps, nous avons effectué des tests avec la validation croisée en utilisant le K-fold (K=10) . Nous avons utilisé les paramètres suivants : Type SVM : C-SVC , type de Noyau : RBF ,gamma= auto, degré= 3 pour plusieurs valeurs de « C » comme suit :

✓ $C = 3$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 81%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,78	0,76	0,78	0,80	0,84	0,93	0,83	0,76	0,80	0,80

Tableau 39 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=3

✓ $C = 0.001$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 39%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39

Tableau 40 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=0.001

✓ $C = 0,1$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 44%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,39	0,43	0,47	0,47	0,47	0,39	0,50	0,43	0,47	0,41

Tableau 41 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=0.1

✓ $C = 1$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 76%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,71	0,78	0,74	0,72	0,74	0,89	0,78	0,71	0,73	0,76

Tableau 42 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=1

✓ $C = 2$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 80%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,76	0,76	0,80	0,80	0,80	0,93	0,82	0,73	0,80	0,80

Tableau 43 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=2

✓ $C = 4$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 81%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,78	0,76	0,78	0,80	0,84	0,93	0,82	0,76	0,80	0,80

Tableau 44 : l'accuracy pour classification multi classes avec HOG , validation croisée K-fold (K=10) et C=4

✓ $C = 5$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 81%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,78	0,76	0,78	0,80	0,84	0,93	0,82	0,76	0,80	0,80

Tableau 45 multi classes avec HOG , validation croisée K-fold (K=10) et C=5

- Pour déterminer des meilleures valeurs des paramètres, nous avons utilisé la stratégie de Grille de recherche qui nous a fourni les valeurs optimales. En effet, nous avons mesuré les performances en validation croisée. Les résultats optimaux ainsi obtenus sont : $C = 3$, noyau= 'RBF', où le best score est 79% ; le taux de reconnaissance obtenu est : 100 %.

4.4.1.3. Descripteur histogramme des couleurs

Finalement, Nous avons essayé la méthode d'histogramme des couleurs qui permet d'extraire des caractéristiques de couleur de chaque image sous forme d'histogrammes sur lesquels on applique ensuite notre classifieur SVM (RBF).

- Dans un premier temps, nous avons choisi aléatoirement 30% instances parmi les 460 exemples d'apprentissage. Nous avons utilisé les paramètres suivants : Type SVM : C-SVC , type de Noyau : RBF , gamma= auto, degré= 3 pour plusieurs valeurs de « C » comme suit :

✓ $C = 3$

Les résultats ainsi obtenus sont montrés dans les tableaux 46 et 47 où le taux de reconnaissance moyenne obtenu est : 79%

	Précision	Recall	F1-score
Bartina	0.75	0.91	0.82
MaladieBartina	0.62	0.20	0.30
MaladieSpunta	0.76	0.80	0.78
Spunta	0.92	1.00	0.96

Tableau 46 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=3

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	53	3	2	0
MaladieBartina	17	5	3	0
MaladieSpunta	1	0	16	3
Spunta	0	0	0	35

Tableau 47 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=3

✓ $C = 0.001$

Les résultats ainsi obtenus sont montrés dans les tableaux 48 et 49 où le taux de reconnaissance obtenu est: 42%

	Précision	Recall	F1-score
Bartina	0.42	1.00	0.59
MaladieBartina	0.00	0.00	0.00
MaladieSpunta	0.00	0.00	0.00
Spunta	0.00	0.00	0.00

Tableau 48 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=0.001

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	58	0	0	0
MaladieBartina	25	0	0	0
MaladieSpunta	20	0	0	0
Spunta	35	0	0	0

Tableau 49 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et $C=0.001$

✓ $C = 0.1$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 43%

	Précision	Recall	F1-score
Bartina	0.42	1.00	0.59
MaladieBartina	0.00	0.00	0.00
MaladieSpunta	0.00	0.00	0.00
Spunta	1.00	0.03	0.06

Tableau 50 : Mesures de performance pour classification multi classes avec histogramme des couleurs et $C=0.1$

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	58	0	0	0
MaladieBartina	25	0	0	0
MaladieSpunta	20	0	0	0
Spunta	34	0	0	1

Tableau 51 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et $C=0.1$

✓ $C = 1$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 78%

	Précision	Recall	F1-score
Bartina	0.71	0.95	0.81
MaladieBartina	1.00	0.12	0.21
MaladieSpunta	0.74	0.70	0.72
Spunta	0.90	1.00	0.95

Tableau 52 : Mesures de performance pour classification multi classes avec histogramme des couleurs et $C=1$

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	55	0	2	1
MaladieBartina	19	3	3	0
MaladieSpunta	3	0	14	3
Spunta	0	0	0	35

Tableau 53 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et $C=1$

✓ $C = 2$

Les résultats ainsi obtenus sont montrés dans les tableaux 54 et 55 où le taux de reconnaissance obtenu est : 80%

	Précision	Recall	F1-score
Bartina	0.75	0.97	0.84
MaladieBartina	0.67	0.16	0.26
MaladieSpunta	0.84	0.80	0.82
Spunta	0.92	1.00	0.96

Tableau 54 : Mesures de performance pour classification multi classes avec histogramme des couleurs et $C=2$

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	56	2	0	0
MaladieBartina	18	4	3	0
MaladieSpunta	1	0	16	3
Spunta	0	0	0	35

Tableau 55: Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=2

✓ $C = 4$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 78%

	Précision	Recall	F1-score
Bartina	0.74	0.91	0.82
MaladieBartina	0.62	0.20	0.30
MaladieSpunta	0.81	0.85	0.83
Spunta	0.95	1.00	0.97

Tableau 56 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=4

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	53	3	2	0
MaladieBartina	18	5	2	0
MaladieSpunta	1	0	17	2
Spunta	0	0	0	35

Tableau 57 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=4

✓ $C = 5$

Les résultats ainsi obtenus sont montrés dans les tableaux suivants où le taux de reconnaissance obtenu est : 81%

	Précision	Recall	F1-score
Bartina	0.75	0.91	0.82
MaladieBartina	0.67	0.24	0.35
MaladieSpunta	0.82	0.90	0.86
Spunta	0.97	1.00	0.99

Tableau 58 : Mesures de performance pour classification multi classes avec histogramme des couleurs et C=5

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	53	3	2	0
MaladieBartina	17	6	2	0
MaladieSpunta	1	0	18	1
Spunta	0	0	0	35

Tableau 59 : Matrice de confusion pour classification multi classes avec histogramme des couleurs et C=5

- Dans un second temps, nous avons effectué des tests avec la validation croisée en utilisant le K-fold (K=10) . Nous avons utilisé les paramètres suivants : Type SVM : C-SVC , type de Noyau : RBF , gamma= auto, degré= 3 pour plusieurs valeurs de « C » comme suit :

✓ $C = 3$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 80%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,89	0,85	0,76	0,78	0,73	0,82	0,82	0,73	0,82	0,78

Tableau 60 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=3

✓ $C = 0.001$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 39%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39

Tableau 61 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=0.001

✓ $C = 0,1$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 54%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,52	0,56	0,52	0,47	0,58	0,56	0,56	0,50	0,58	0,52

Tableau 62 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=0.1

✓ $C = 1$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 78%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,80	0,82	0,73	0,78	0,74	0,76	0,82	0,80	0,76	0,73

Tableau 63 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=1

✓ $C = 2$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 80%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,86	0,84	0,76	0,80	0,73	0,78	0,86	0,76	0,82	0,76

Tableau 64 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=2

✓ $C = 4$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 79%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,89	0,84	0,76	0,78	0,69	0,82	0,82	0,73	0,82	0,78

Tableau 65 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=4

✓ $C = 5$

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 80%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,89	0,84	0,78	0,18	0,67	0,89	0,83	0,74	0,83	0,76

Tableau 66 : l'accuracy pour classification multi classes avec histogramme des couleurs , validation croisée K-fold (K=10) et C=5

- Pour déterminer des meilleures valeurs des paramètres, nous avons utilisé la stratégie de Grille de recherche qui nous a fourni les valeurs optimales. En effet, nous avons mesuré les performances en validation croisée.

Les résultats optimaux ainsi obtenus sont : $C = 4$, noyau= 'RBF', où le best score est 79% ; le taux de reconnaissance obtenu est : 96 %.

4.4.1.4. Fusion de caractéristiques

Après qu'on a testé chaque descripteur de façon isolé (LBP, HOG , histogramme de couleur), on est intéressé finalement à tester la combinaison de ces descripteurs. Pour cela, nous avons fusionné, en un seul vecteur, les caractéristiques susmentionnées (LBP+HOG +histogramme de couleur). Après la concaténation des vecteurs de caractéristiques en un seul vecteur, on doit le passer à l'algorithme d'apprentissage SVM pour la construction du modèle de décision.

- Nous avons utilisé les paramètres de classifieur suivants : Type SVM : C-SVC , type de Noyau : RBF , gamma= auto, degré= 3 ,C=3.

✓ Dans un premier temps, nous avons choisi aléatoirement 30% instances parmi les 460 exemples d'apprentissage.

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance obtenu est : 70%

	Bartina	MaladieBartina	MaladieSpunta	Spunta
Bartina	63	0	0	0
MaladieBartina	24	0	0	0
MaladieSpunta	11	0	0	3
Spunta	4	0	0	33

Tableau 67 : Matrice de confusion pour classification multi classes avec fusion de caractéristiques

✓ Dans un second temps, nous avons effectué des tests avec la validation croisée en utilisant le K-fold (K=10).

Les résultats ainsi obtenus sont montrés dans le tableau suivant où le taux de reconnaissance moyenne obtenu est : 56%

K	1	2	3	4	5	6	7	8	9	10
Accuracy	0,97	0,84	0,95	0,91	0,00	0,00	0,00	0,41	0,63	0,87

Tableau 68 : l'accuracy pour classification multi classes avec fusion de caractéristiques , validation croisée K-fold (K=10)

4.5. Discussions

Les résultats obtenus nous montrent que le système soit performant (taux de reconnaissance > 70%) avec des valeurs de paramètres $C > 2$ pour les deux tests (la subdivision manuelle et la validation croisée avec k-fold (k=10). Tandis que les meilleurs résultats sont obtenus (taux de reconnaissance > 70%) avec le grille search. Ceci pour les caractéristiques de texture (LBP) et de couleur (histogramme de couleur). Cependant , les performances de notre système sont dégradées si on cherche à fusionner les descripteurs. Cette dégradation est à cause de manque des exemples pour certaines classes (bartina avec maladie par exemple).

4.6. Classification en ligne des images non vue

Pour tester en ligne notre système, nous avons utilisé le modèle de classification performant pour classer des nouvelles images de pommes de terre. La figure 26 montre des exemples des pommes de terre après classification en ligne.



Figure 26 : Images après classification en ligne

5. Conclusion

Dans ce chapitre, nous avons implémenté le programme correspondant au système proposé, décrit l'environnement de développement et les mesures de performance. Ainsi qu'on a présenté certains algorithmes utilisés avec les résultats obtenus.

Conclusion générale

Notre objectif de ce mémoire est de résoudre le problème du tri manuel des pommes de terre au fait de donner la capacité d'apprendre à la machine pour distinguer automatiquement les différentes catégories de ce produit agricole. Ceci par la réalisation d'un système de tri automatique, en utilisant les dernières techniques d'apprentissage machine et de Vision par ordinateur, dont le principal objectif est d'améliorer la production de pommes de terre. Ce dernier vise à analyser une image de pomme de terre, ensuite extraite les propriétés visuelles de la pomme de terre puis de prédire la classe de cette pomme de terre.

Nous avons vu en premier temps le concept du système de tri avec quelques exemples et mesures basées sur la vision par ordinateur pour le processus de tri. On a vu aussi des concepts de base en reconnaissance de formes, apprentissage automatique, extraction de caractéristiques locales et globales, et deux types d'algorithmes de classification avec leurs Avantages et leurs Inconvénients respectifs. Dans la phase de conception, nous avons présenté l'architecture de notre système développé, avec une explication détaillé de chaque étape et les méthodes utilisées pour l'extraction de caractéristiques et la classification multi-classes appliquée à notre problème (plusieurs classes de pomme de terre). Dans le dernier chapitre une vue sur l'évaluation des performances de notre système a été présentée. Nous avons présenté les méthodes de validation qui permettent de les calculer notamment la validation croisée, les critère d'évaluation des techniques de classification les plus couramment utilisée à savoir la matrice de confusion, ainsi que les outils de programmation et les algorithmes utilisés.

A la fin de ce mémoire, la solution que nous proposons dans le domaine de l'agriculture consiste à introduire un système intelligent dans le processus de tri des produits agricoles en utilisant des techniques d'apprentissage automatisé et une vision par ordinateur basée sur les caractéristiques visuelles.

Comme perspective, ce système peut également être utilisé pour trier les fruits et autres légumes tels que les pommes, les oranges, etc. Deux autres axes peuvent être aussi investigués afin d'améliorer les performances. D'une part, on cherche à combiner des caractéristiques globales et locales et tester d'autres algorithmes d'apprentissage tels que les KNN. D'autre part, et pour surmonter les défis du processus de Machine

Conclusion générale

Learning qui est complexe et coûteux en temps, nous avons l'idée d'appliquer les nouvelles perspectives de ce processus qui concerne l'apprentissage profond (ou Le Deep Learning).

Bibliographie

- [1] **Deep Learning et Agriculture** – Une étude de la Chaire AgroTIC – Novembre 2018
- [2] Snyder, W. E., & Qi, H. (2017). **Fundamentals of computer vision** . Cambridge University Press.
- [3] HADJAMMAR Tedjani(2018), **Effet des techniques culturales et type de sol sur la qualité du rendement de la culture de pomme de terre dans la région d'Oued Souf** , MÉMOIRE DE MASTER, Université de Biskra.
- [4] Boumaraf Ibtissam (2016). **tri des dates à base de caractéristiques approfondies**, mémoire de master université de Biskra,
- [5] Saldaña, E., Siche, R., Luján, M., & Quevedo, R. (2013). **Computer vision applied to the inspection and quality control of fruits and vegetables**. Brazilian Journal of Food Technology, 16(4), 254-272.
- [6] Narendra, V. G., & Hareesha, K. S. (2010). **Prospects of computer vision automated grading and sorting systems in agricultural and food products for quality evaluation** . International Journal of Computer Applications, 1(4), 1-9.
- [7] Bhargava, A., & Bansal, A. (2018). **Fruits and vegetables quality evaluation using computer vision: A review**. Journal of King Saud University-Computer and Information Sciences.
- [8] Mahendran, R., Jayashree, G. C., & Alagusundaram, K. (2012). **Application of computer vision technique on sorting and grading of fruits and vegetables** . J Food Process Technol, 10, 2157-7110.
- [9] Noura et Rima BEN CHEIKH et BEN BEZZIANE,(2011), **La recherche d'images par la sémantique** , MÉMOIRE Master, UNIVERSITE KASDI MERBAH-OUARGLA-. en ligne [https://www.memoireonline.com/05/14/8871/m_La-recherche-d-images-par-la-semantique11.html]
- [10] Jain, A. K., Duin, R. P. W., & Mao, J. (2000). **Statistical pattern recognition: A review** . IEEE Transactions on pattern analysis and machine intelligence, 22(1), 4-37.
- [11] DJABEUR, D. M. R., & BENKADA, F. A(2018). **Mise au Point d'une Application de Reconnaissance de Formes** (Doctoral dissertation, 08-01-2018).
- [12] Moualek, D. Y. (2017). **Deep Learning pour la classification des images** (Doctoral dissertation, 07-03-2017).

- [13] JOSEPH Metomo, BERTRAND RAPHAËL. **Machine Learning : Introduction à l'apprentissage automatique**. SUPINFO international university. 2017. Web 10/10/2017. <https://www.supinfo.com/articles/single/6041-machine-learning-introduction-apprentissage-automatique>
- [14] Pirmin Lemberger, Marc Batty, Médéric Morel, Jean-Luc Raffaëlli(2016). **Big Data et Machine Learning - Les concepts et les outils de la data science**, 2e édition, Dunod.
- [15] Kpalma, K., & Ronsin, J. (2007). **An overview of advances of pattern recognition systems in computer vision**.
- [16] Bastien L. Machine Learning – **Définition, fonctionnement et secteurs d'application**. Artificiel.net – Le magazine de l'Intelligence Artificielle. 2017. Web mai 26, 2017. <http://www.artificiel.net/machine-learning-definition>.
- [17] Lowe, D. G. (2004). **Distinctive image features from scale-invariant keypoints**. International journal of computer vision, 60(2), 91-110.
- [18] Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). **Surf: Speeded up robust features**. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.
- [19] Ping Tian, D. (2013). **A review on image feature extraction and representation techniques**. International Journal of Multimedia and Ubiquitous Engineering, 8(4), 385-396. En ligne
[<https://pdfs.semanticscholar.org/a820/b7882ea2365e0396d100a412126e93611a6f.pdf>]
- [20] Denis, P. (2007). **Quaternions et Algèbres Géométriques, de nouveaux outils pour les images numériques couleur** (Doctoral dissertation, Université de Poitiers).
- [21] Denis, P., Carre, P., & Fernandez-Maloigne, C. (2007). **Spatial and spectral quaternionic approaches for colour images**. *Computer Vision and Image Understanding*, 107(1-2), 74-87.
- [22] Larson, G. W. (1998). **LogLuv encoding for full-gamut, high-dynamic range images**. *Journal of Graphics Tools*, 3(1), 15-31.
- [23] Rafael C. Gonzalez and Richard E. Woods(2001). **Digital Image Processing**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [24] Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., & Yamada, A. (2001). **Color and texture descriptors**. *IEEE Transactions on circuits and systems for video technology*, 11(6), 703-715.
- [25] Manjunath, B. S., Salembier, P., & Sikora, T. (Eds.). (2002). **Introduction to MPEG-7: multimedia content description interface**. John Wiley & Sons.

- [26] Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). **Multiresolution gray-scale and rotation invariant texture classification with local binary patterns**. IEEE Transactions on Pattern Analysis & Machine Intelligence, (7), 971-987.
- [27] Dalal, N., & Triggs, B. (2005, June). **Histograms of oriented gradients for human detection**. In international Conference on computer vision & Pattern Recognition (CVPR'05) (Vol. 1, pp. 886-893). IEEE Computer Society.
- [28] Bober, M. (2001). **MPEG-7 visual shape descriptors**. IEEE Transactions on circuits and systems for video technology, 11(6), 716-719.
- [29] Pippa, E., Zacharaki, E. I., Özdemir, A. T., Barshan, B., & Megalooikonomou, V. (2018, July). **Global vs local classification models for multi-sensor data fusion**. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence (p. 43). ACM
- [30] Gideon Dror . May 31, 2007. **Analysis of Gene Expression Data**. Scribe: Yonit Halperin and David Burstein1.
- [31] Fulkerson, B. (1995). **Machine Learning, Neural and Statistical Classification**.
- [32] DERKAOU, A., & DERKAOU, S. **ETUDE COMPARATIVE DES METHODES ENSEMBLISTES DE CLASSIFICATION DES DONNES MEDICALES** (Doctoral dissertation).
- [33] Aly, M. (2005). **Survey on multiclass classification methods**. Neural networks, 19, 1-9.
- [34] Puget, R. (2016). **Etude de la classification dans un très grand nombre de catégories** (Doctoral dissertation, Paris 6).
- [35] Fu, J., Huang, C., & Lee, S. (2008). **A multi-class svm classification system based on methods of self-learning and error filtering**. Taiwan, Republic of China
- [36] Chen, Z., Pears, N., Freeman, M., & Austin, J. (2009, November). **Road vehicle classification using support vector machines**. In 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems (Vol. 4, pp. 214-218). IEEE.
- [37] Kallas, M. (2012). **Méthodes à noyaux en reconnaissance de formes, prédiction et classification. Applications aux biosignaux** (Doctoral dissertation, Université de Technologie de Troyes).
- [38] Shih, F. Y. (2010). **Image processing and pattern recognition: fundamentals and techniques**. John Wiley & Sons.

- [39] Harwood D, Ojala T, Pietikäinen M, Kelman S & Davis S (1993) „**Texture classification a by center-symmetric auto-correlation, using Kullback discrimination of distributions**, Technical report, Computer Vision Laboratory“, Center for Automation Research, University of Maryland, College Park, Maryland. CAR-TR678.
- [40] Ojala, T., Pietikäinen, M., & Harwood, D. (1996). **A comparative study of texture measures with classification based on featured distributions**. Pattern recognition, 29(1), 51-59.
- [41] Huang, D., Shan, C., Ardabilian, M., Wang, Y., & Chen, L. (2011). **Local binary patterns and its application to facial image analysis: a survey**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6), 765-781.
- [42] Soltana, W. B., Porebski, A., Vandenbroucke, N., Ahmad, A., & Hamad, D. (2014). **Contribution des descripteurs de texture LBP à la classification d'images de dentelles**. Article RI 2014
- [43] Swain, M. J., & Ballard, D. H. (1991). **Color indexing**. International journal of computer vision, 7(1), 11-32.
- [44] Pass, G., & Zabih, R. (1996, December). **Histogram refinement for content-based image retrieval**. In *Proceedings Third IEEE Workshop on Applications of Computer Vision*. WACV'96 (pp. 96-102). IEEE.
- [45] Jeong, S. (2001). **Histogram-based color image retrieval**. Psych221/EE362 project report
- [46] James Z. Wang, 2001. **"Integrated Region-Based Image Retrieval"**, Boston, Kluwer Academic Publishers.
- [47] Hu, R., & Collomosse, J. (2013). **A performance evaluation of gradient field HOG descriptor for sketch based image retrieval**. *Computer Vision and Image Understanding*, 117(7), 790-806.
- [48] Suard, F., Rakotomamonjy, A., Bensrhair, A., & Broggi, A. (2006, June). **Pedestrian detection using infrared images and histograms of oriented gradients**. In *2006 IEEE Intelligent Vehicles Symposium* (pp. 206-212). IEEE.
- [49] Watanabe, T., Ito, S., & Yokoi, K. (2009, January). **Co-occurrence histograms of oriented gradients for pedestrian detection**. In *Pacific-Rim Symposium on Image and Video Technology* (pp. 37-47). Springer, Berlin, Heidelberg.
- [50] V. N. Vapnik. 1999. **The Nature of Statistical Learning Theory**. Second Edition, Springer.

- [51] Schölkopf, B., Smola, A. J., & Bach, F. (2002). **Learning with kernels: support vector machines, regularization, optimization, and beyond**. MIT press.
- [52] Milgram, J. (2007). **Contribution à l'intégration des machines à vecteurs de support au sein des systèmes de reconnaissance de formes: application à la lecture automatique de l'écriture manuscrite** (Doctoral dissertation, École de technologie supérieure).
- [53] Laouti, N. (2012). **Diagnostic de défauts par les Machines à Vecteurs Supports: application à différents systèmes multivariables nonlinéaires** (Doctoral dissertation, Université Claude Bernard-Lyon I)
- [54] Hasan, M., & Boris, F. (2006). **Svm: Machines à vecteurs de support ou séparateurs à vastes marges**. Rapport technique, Versailles St Quentin, France. Cité, 64.
- [55] DJEFFAL, A. (2012). **Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données** (Doctoral dissertation, Université Mohamed Khider-Biskra).
- [56] Python . En ligne [<https://www.python.org/doc/essays/blurb/>].
- [57] Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages. En ligne [https://jupyter.org/?fbclid=IwAR2rcvcDSc-cbOEB8p9rdFjCTo6U1czQtuEgk1H4GyV8Fom98t_fGEDoa-M].
- [58] Scikit-image En ligne [[language.https://scikit-image.org/](https://scikit-image.org/)].
- [59] Documentation of scikit-learn 0.21.2 En ligne [<https://scikit-learn.org/stable/documentation.html>].
- [60] Josh Patterson, Adam Gibson, « Deep learning en action: La référence du praticien », First Ed, O'Reilly, 24 mai 2018 - 534 pages
- [61] Scikit-learn En ligne [https://scikit-learn.org/stable/modules/model_evaluation.html?fbclid=IwAR39Q7kYo3j3COAPbpp7vOeiCuZunVHGyC6yYjgUkBH89rzLD5mrYhBTx14].
- [62] Morgane LAUR, 2017, « Anticipation des changements de notes des obligations du portefeuille d'un assureur par méthode de machine learning », Master Actuariat de Dauphine, Université Paris.

Annexe

- **Feature extraction-01LBP**

```
In [ ]: def lbp_histogram(img):  
        patterns = local_binary_pattern(img, 8, 1, method='uniform')  
        hist, _ = np.histogram(patterns, bins=np.arange(11), density=True)  
        return hist
```

```
In [3]: feature_file = './feature_dataLBP.csv'  
        data_path = './DataSet1/'  
  
        target = 0  
        sample_n = 0  
  
        feature_array = [] # ['feature1', 'feature2', 'feature3', 'feature4', .. 'feature K', 'target']  
  
        t1 = time.time()  
  
        for data in listdir(data_path):  
            # navigate each class and all data within it  
            # print(f'Class: {data}, target: {target}')  
            print('Class: {} , target: {}'.format(data, target))  
            for img in listdir(data_path + data):  
                # extract their features  
                # lbp_texture extraction  
                img_path = data_path + data + '/' + img  
                image = cv.imread(img_path, 0)  
                image = cv.resize(image, (int((image.shape[0])/3), int((image.shape[1])/3)))  
                # gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)  
                lbp_hist = lbp_histogram(image)  
                lbp_hist = np.append(lbp_hist[:], target) # lbp histogram data selection  
                feature_array = np.append(feature_array, lbp_hist[:])  
                sample_n += 1 # count number of data samples  
  
            target += 1  
        t2 = time.time()
```

```
Class: bartina , target: 0  
Class: maladieBartina . target: 1
```

- Create and train SVC model from imported csv dataset 01LBP

importing train data ..

```
In [2]: feature_file = './feature_dataLBP.csv'
data = pd.read_csv(feature_file)
data.head(10)
```

Out[2]:

	lbp_gray_0	lbp_gray_1	lbp_gray_2	lbp_gray_3	lbp_gray_4	lbp_gray_5	lbp_gray_6	lbp_gray_7	lbp_gray_8	lbp_gray_9	Class
0	0.071207	0.098631	0.044727	0.072643	0.104078	0.091970	0.071964	0.107568	0.143447	0.193766	2
1	0.089574	0.108883	0.048541	0.061267	0.071616	0.070571	0.068222	0.115441	0.147934	0.217952	0
2	0.028631	0.083339	0.039165	0.098048	0.145988	0.158144	0.089465	0.095135	0.138060	0.124024	0
3	0.071784	0.099447	0.050480	0.076583	0.096312	0.090132	0.076126	0.109381	0.135724	0.194030	0
4	0.096901	0.108973	0.050456	0.059315	0.061538	0.065069	0.069429	0.113483	0.150498	0.224336	0
5	0.095303	0.108468	0.049550	0.060925	0.067267	0.067988	0.069784	0.114535	0.146492	0.219688	1
6	0.016703	0.069688	0.028318	0.099706	0.148012	0.172216	0.090967	0.089988	0.168859	0.115544	0
7	0.084006	0.106787	0.049117	0.067261	0.077333	0.074781	0.070529	0.113682	0.148565	0.207940	0
8	0.027273	0.068330	0.032252	0.099243	0.159658	0.171177	0.088511	0.088300	0.152559	0.112697	0
9	0.085706	0.105357	0.048204	0.066517	0.082210	0.080763	0.072090	0.111345	0.141123	0.206685	0

```
In [8]: y_pred = clf.predict(X_test)
```

Evaluating the Classifier

```
In [9]: from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
#matrice de confusion
#confrontation entre Y obs. sur l'éch. test et la prédiction
cm = confusion_matrix(y_test,y_pred)
print(f'Confusion_Matrix: {cm}')
#taux de succès
acc = accuracy_score(y_test,y_pred)
print(f'Accuracy: {acc}')

targets = ['bartina', 'maladeBartina', 'maladeSpunta', 'spunta']
print(classification_report(y_test,y_pred,target_names=targets))
```

```
Confusion_Matrix: [[49  0  1  2]
 [21  4  2  0]
 [ 0  0 19  5]
 [ 2  0  0 33]]
Accuracy: 0.7608695652173914
```

	precision	recall	f1-score	support
bartina	0.68	0.94	0.79	52
maladeBartina	1.00	0.15	0.26	27
maladeSpunta	0.86	0.79	0.83	24
spunta	0.82	0.94	0.88	35
micro avg	0.76	0.76	0.76	138
macro avg	0.84	0.71	0.69	138
weighted avg	0.81	0.76	0.72	138

- **Create and train SVC model from imported csv dataset 01Validation croiseekfoldLBP**

Evaluating the Classifier

```
In [12]: from sklearn.metrics import classification_report, confusion_matrix
#utilisation du module model_selection
from sklearn import model_selection
#évaluation en validation croisée : 10 cross-validation
succes = model_selection.cross_val_score(clf,new_X,y,cv=10,scoring='accuracy')
#détail des itérations
print('k-fold k=10')
print(f'accuracy={succes}')
#moyenne des taux de succès = estimation du taux de succès en CV
print(f'myenne accuracy={succes.mean()}')
```

k-fold k=10
accuracy=[0.73913043 0.73913043 0.82608696 0.73913043 0.76086957 0.76086957
0.65217391 0.7826087 0.76086957 0.7826087]
myenne accuracy=0.7543478260869565

- **Create and train SVC model from imported csv dataset 01Grille searchLBP**

Evaluating the Classifier

```
In [31]: #import de la classe
from sklearn import model_selection
#combinaisons de paramètres à évaluer
parametres = [{'C':[1e-3, 1e-1, 1, 2, 3, 4, 5], 'kernel':['rbf', 'linear']}]
#évaluation en validation croisée de 3 x 2 = 6 configurations
#accuracy sera le critère à utiliser pour sélectionner la meilleure config
#mvs est l'instance de la classe svm.SVC (cf. page précédente)
grid = model_selection.GridSearchCV(estimator=clf,param_grid=parametres,scoring='accuracy')
#lancer la recherche - attention, gourmand en calculs
grille = grid.fit(new_X,y)
#résultat pour chaque combinaison
print(pd.DataFrame.from_dict(grille.cv_results_).loc[:,["params", "mean_test_score"]])
#meilleur paramétrage
print(grille.best_params_) # {'C' : 10, 'kernel' : 'linear'}
#meilleure performance - estimée en interne par validation croisée
print(grille.best_score_)
#prédiction avec le modèle « optimal » c.-à-d. {'C' : 10, 'kernel' : 'linear'}
y_pred3 = grille.predict(X_test)
#taux de succès en test
print(metrics.accuracy_score(y_test,y_pred3))
```


- **Feature extraction-01GlobalFeaures**

```
In [2]: def lbp_histogram(img):
        patterns = local_binary_pattern(img, 8, 1, method='uniform')
        hist, _ = np.histogram(patterns, bins=np.arange(11), density=True)
        return hist

        # feature-descriptor-3: Color Histogram
        def fd_histogram(image, mask=None):
            # convert the image to HSV color-space
            image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
            # compute the color histogram
            hist = cv2.calcHist([image], [0, 1, 2], None, [bins, bins, bins], [0, 256, 0, 256, 0, 256])
            # normalize the histogram
            cv2.normalize(hist, hist)
            # return the histogram
            return hist.flatten()

In [3]: #feature_file = './Globalfeature_data.h5'
        data_path = './dataset1/'
        target = 0
        sample_n = 0
        global_features = [] # ['feature1', 'feature2', 'feature3', 'feature4', .. 'feature K', 'target']
        labels = []
        t1 = time.time()

        for data in listdir(data_path):
            # navigate each class and all data within it
            print(f'Class: {data}, target: {target}')
            # get the current training label
            current_label = data
            for img in listdir(data_path + data):
                # extract their features
```

- **Create and train SVC model from imported h5 dataset Globalfeautres**

Evaluating the Classifier

```
In [8]: kfold = KFold(n_splits=10, random_state=7)
        cv_results = cross_val_score(clf, global_features, global_labels, cv=kfold, scoring=scoring)
        print('k-fold k=10')
        print(f'accuracy={cv_results}')
        #moyenne des taux de succès = estimation du taux de succès en CV
        print(f'moyenne accuracy={cv_results.mean()}')
        #results.append(cv_results)
        #names.append(name)
        # msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
        #print(msg)
        #from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
        #matrice de confusion
        #confrontation entre Y obs. sur l'éch. test et la prédiction
        cm = confusion_matrix(testLabelsGlobal, y_pred)
        print(f'Confusion_Matrix: {cm}')
        #taux de succès
        acc = accuracy_score(testLabelsGlobal, y_pred)
        print(f'Accuracy: {acc}')

k-fold k=10
accuracy=[0.97826087 0.84782609 0.95652174 0.91304348 0.          0.
          0.41304348 0.63043478 0.86956522]
moyenne accuracy=0.5608695652173913
Confusion_Matrix: [[63  0  0  0]
 [24  0  0  0]
 [11  0  0  3]
 [ 4  0  0 33]]
Accuracy: 0.6956521739130435
```