

CR chef d'œuvre : Phase de définition

Table des matières

Table des matières	1
1) Base de données	2
1.1) Description de la base d'images sélectionnées ou construites	2
1.2) Définition de la procédure de préparation des données.....	2
1.2.1) Visualisation de la base.....	2
1.2.2) Analyse descriptive des données	3
1.2.3) Traitement	4
2) Caractéristiques	5
2.1) Etat de l'art des caractéristiques	5
2.2) Définition et description des caractéristiques	5
2.3) Définition des critères d'évaluation des caractéristiques.....	5
3) Classifieur	6
3.1) Etat de l'art des classifieurs	6
3.2) Définition et description du classifieur	6
Exemple :.....	6
3.3) Définition des critères d'évaluation du classifieur.....	7

1) Base de données

1.1) Description de la base d'images sélectionnées ou construites

Pour l'apprentissage, nous nous sommes basés sur une BDD libre de droit sur Kaggle.com

(<https://www.kaggle.com/datasets/aelchimminut/fruits262>)

Nous avons choisi de détecter la liste suivante de fruits :

Pomme
Orange
Abricot

Banane
Raisin
Myrtille

Fraise
Poire
Kiwi


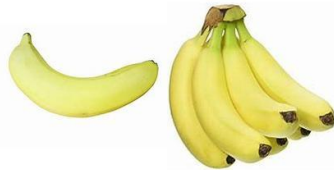
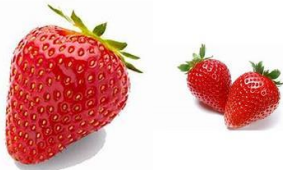






Chaque image de fruit contient une unité de fruit fermé (1 pomme, 1 grappe de raisin, une banane ou un régime de banane, plusieurs myrtilles, une fraise, une poire...) sur fond uni (avec potentielle dérive de fond) ou à peu près uni (pas au milieu d'un arbre, etc...)

Nous avons sélectionné au moins 41 images par classe dans cette BDD, en éliminant notamment les dessins, fruit dans leurs arbres ou images de synthèse hors-sujet...

L'arborescence est sous la forme de dossiers portant chacun le nom du fruit correspondant, en anglais. Dans chaque dossier, il y a les images du fruit correspondant, de taille 208x256. Chaque image est nommée par un numéro, et est au format jpg.

1.2) Définition de la procédure de préparation des données

1.2.1) Visualisation de la base

Pomme	Banane	Fraise
		
Orange	Raisin	Poire
		
Abricot	Myrtille	Kiwi
		

1.2.2) Analyse descriptive des données

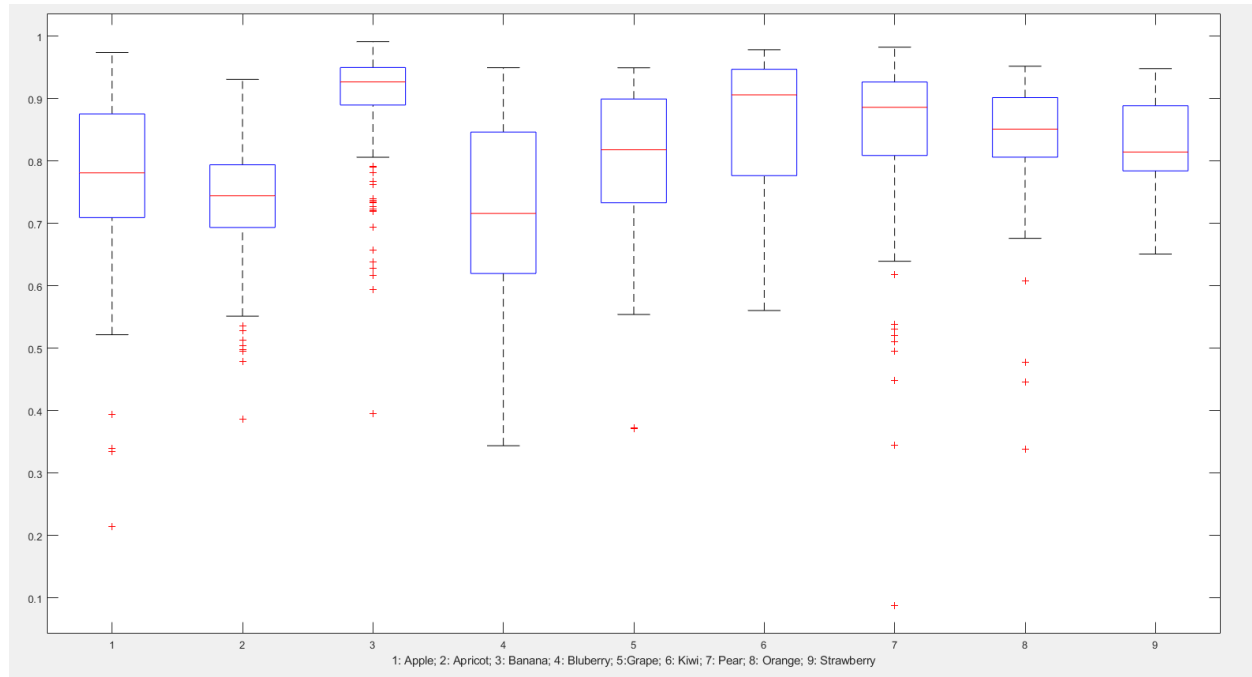


Figure 1: Boite à moustache présentant le niveau de gris moyen par classe de fruit (légende en titre de la figure Matlab)

On peut observer sur la boîte à moustache qu'il y a quelques cas extrêmes (représentés par des croix rouges éloignées de la boîte correspondant à leur classe), principalement dans les classes des poires, des pommes et des raisins, qui sont liés au fond. Nous avons cependant anticipé ce problème, dont on a prévu de s'occuper durant l'étape du pré-traitement, comme indiqué sur le plan d'architecture.

Les images sont déjà à l'échelle.

1.2.3) Traitement

En ce qui concerne le traitement, nous allons enlever le fond des images à l'aide du gradient et d'un seuil pour détecter les contours des fruits. Ensuite nous pourrons faire les analyses des images sur les pixels à l'intérieur de ces contours.

2) Caractéristiques

2.1) Etat de l'art des caractéristiques

Nous avons trouvé des caractéristiques qui sont connues pour être de solides descripteurs pour les images de fruits.

<https://www.sciencedirect.com/science/article/abs/pii/S0168169921002842> : "We propose a novel combination of Hue, Color-SIFT, Discrete Wavelet Transform and Haralick features in fruit classification problem that outperforms other handcrafted visual features. [...] It is found that Color-SIFT features alone work very well for fruit classification problem by outperforming other individual handcrafted features."

La méthode SIFT en couleur est cependant difficile à implémenter, nous la réservons donc au cas où les autres ne sont pas suffisantes.

2.2) Définition et description des caractéristiques

Nous allons extraire dans un premier temps quelques caractéristiques élémentaires (couleur, motifs locaux et fréquences). Ces caractéristiques nous permettront normalement de séparer les classes qui sont éloignées, par exemple les myrtilles et les fraises. Nous allons ensuite nous concentrer sur les quelques classes (probablement de l'ordre de 3-4 classes) qui ne seront pas séparables de cette façon (oranges et abricots par exemple car elles semblent proches).

- Caractéristique couleur : Hue dominante (médiane) d'une image (espace HSV), permet de distinguer les fruits dont la couleur est différente.
- Caractéristique de motifs locaux : LBP sur la luminance (à plusieurs échelles : pleine échelle, échelle $\frac{1}{2}$, échelle $\frac{1}{4}$), permet de distinguer les fruits selon la texture de leur peau.
- Caractéristique fréquentielle : histogramme des gradients orientés sur la luminance (mêmes échelles que LBP), permet de distinguer les fruits par leurs variations

2.3) Définition des critères d'évaluation des caractéristiques

Evaluation par analyse PCA pour trouver les caractéristiques les plus discriminantes, si on a des caractéristiques qui sont faiblement représentées, que l'on peut alors supprimer. Une sous-caractérisation sera évaluée au travers du classifieur. Si celui-ci ne converge pas vers une précision satisfaisante, il peut alors être envisageable d'ajouter des caractéristiques.

3) Classifieur

3.1) Etat de l'art des classifieurs

Au niveau des classifieurs, dans la même source que pour les caractéristiques, nous avons : "Out of different classifiers best results are obtained using Back Propagation Neural Network, SVM and KNN classifier with classification accuracies between 99% and 100%."

3.2) Définition et description du classifieur

Un classifieur KNN serait le plus simple à implémenter donc nous choisissons celui-ci.

Méthodologie de mesure : On utilise un jeu de données étiquetées grâce à l'algorithme présenté ci-dessous.

Les k-plus proches voisins (k-NN) : Cet algorithme consiste à attribuer une étiquette à un nouvel exemple en fonction de la classe majoritaire parmi les k exemples les plus proches dans l'espace des caractéristiques. Il est facile à comprendre et à mettre en œuvre, mais peut être sensible aux données bruyantes et nécessite une sélection appropriée du nombre de voisins.

Fonctionnement de l'algorithme : On calcule la distance euclidienne entre la nouvelle donnée et les données de la base de données afin d'attribuer à cette donnée la classe dont la distance est la plus faible (dans notre cas nous devons tenir compte du nombre d'image par classe qui n'est pas fixe pour toutes les classes de fruits).

Exemple :

Images	Caractéristique Couleur	Caractéristique LPB	Caractéristique fréquentielle
Image1.jpg	0.1	0.2	0.3
Image2.jpg	0.2	0.3	0.4
Image3.jpg	0.3	0.4	0.5

Supposons qu'on ait une nouvelle donnée d'entrée :

$$Image\ 4 = [0.35; 0.45; 0.55]$$

Il calcule alors la distance euclidienne à chaque image. Ici on suppose que $k = 3$, c'est à dire qu'il y a que 3 classes de fruits (dans notre cas il y a 10 classes). On applique la formule suivante :

$$d(X, Y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

On a alors :

$$D1 \text{ (distance entre image 1 et image 4) } = 0.424$$

$$D2 = 0.280$$

$$D3 = 0.117$$

Ainsi, ici, le classifieur donnerait ici pour l'image 4 la classe o\$ \$p\$ \$p\$ \$.

3.3) Définition des critères d'évaluation du classifieur

Pour évaluer la performance de l'algorithme de classification à l'aide de la méthode de validation croisée, on suit les étapes suivantes :

- Diviser l'ensemble de données en deux parties : un ensemble d'apprentissage et un ensemble de test. L'ensemble d'apprentissage est utilisé pour entraîner le modèle, tandis que l'ensemble de test est utilisé pour évaluer la performance du modèle.
- Diviser l'ensemble d'apprentissage en k sous-ensembles, également appelés « folds » (plis). Par exemple, pour k = 5, on divise l'ensemble d'apprentissage en cinq sous-ensembles de taille égale.
- Entraîner le modèle sur k-1 folds et évaluer la performance du modèle sur le fold restant. On répète cette étape k fois, chaque fois en utilisant un fold différent comme ensemble de test.
- Calculer la performance moyenne du modèle sur les k folds. Par exemple, on peut calculer la précision moyenne, le rappel moyen, la F-mesure moyenne ou tout autre métrique de performance pertinente pour le problème de classification (on prendra probablement l'accuracy). On peut également utiliser une matrice de confusion, puisque notre sortie est binaire (classe correctement attribuée ou non).
- Répéter les étapes 2 à 4 pour différentes valeurs de k et calculer la performance moyenne du modèle pour chaque valeur de k.
- Choisir la valeur de k qui donne la meilleure performance moyenne pour le modèle. Cette valeur de k est utilisée pour entraîner le modèle final sur l'ensemble d'apprentissage complet et évaluer la performance du modèle sur l'ensemble de test.
- Analyser les résultats de la performance du modèle sur l'ensemble de test pour déterminer si le modèle est suffisamment performant pour répondre aux besoins de l'application.