

MALLOWMETER

– Synthèse et Analyse des résultats –

ÉTUDE EN FILIÈRE INGÉNIEUR SOUS STATUT ÉTUDIANT
21.03.2025

MAXIME JOURNOUD LUCAS LESCURE AUBIN SIONVILLE
RUBEN VERCHERE

Table des Matières

I. Rappel du projet	3
II. Synthèse	3
1. k-NN.....	4
2. CNN.....	5
III. Conclusion	7
IV. Annexes	8
1. Matrices de distance des descripteurs.....	8
2. Cartes de confusion - k-NN	10
3. Cartes de confusion - CNN	11

I. Rappel du projet

L'objectif est de classer des photos de marshmallows blancs en fonction de leur niveau de cuisson. L'application devra fournir, à partir d'une photo de marshmallow, une étiquette correspondant à sa classe parmi les **4 degrés de cuisson** possibles:

① Pas cuit ② Peu cuit ③ Bien cuit ④ Trop cuit

II. Synthèse

Pour l'évaluation de nos données, nous utilisons la mesure de l'équilibre entre les classes, l'**entropie de Shannon normalisée**, définie par :

$$H_n = -\frac{1}{\log_2(k)} \sum_{i=0}^{k-1} p_i \log_2(p_i) \quad \text{où} \quad p_i = \frac{N_i}{\sum_{j=0}^{k-1} N_j}$$

Avec H_n qui varie entre 0 et 1, les valeurs proches de 0 correspondant à une base très déséquilibrée et les valeurs proches de 1 à une base très équilibrée. Ainsi, nous considérons que la base est suffisamment équilibrée si $H_n > 0.75$. Cette métrique est particulièrement pertinente dans notre contexte car un déséquilibre trop important entre les classes pourrait **biaisier l'apprentissage** des modèles et favoriser la classification vers les classes majoritaires.

En appliquant cette évaluation sur l'ensemble de la base de données, on obtient une **entropie** $H_n = 0.993$. On en conclut que la base est **très équilibrée** entre les différentes classes de marshmallow. Ce qui est cohérent puisque les acquisitions réalisées ont été réfléchies pour ne pas déséquilibrer les classes. Cette **valeur proche de 1 confirme la qualité de notre jeu de données** et nous permet d'avoir confiance dans la fiabilité des résultats de classification que nous obtiendrons par la suite.

La pertinence des caractéristiques est contrôlée par l'évaluation des **matrices de distance** (*voir Annexes*). Ces matrices permettent d'identifier les bons descripteurs par des matrices dont les distances entre classes sont différentes et **discriminantes**. Une caractéristique pertinente produira une matrice de distance présentant des **blocs distincts** correspondant aux différentes classes, indiquant que les échantillons d'une même classe sont plus proches entre eux qu'avec ceux des autres classes. Grâce à l'évaluation des performances des classifications pour les descripteurs choisis nous pourrons aussi relever les **taux d'associations correctes** parmi toutes les associations réalisées afin de déterminer si les caractéristiques choisies sont bonnes.

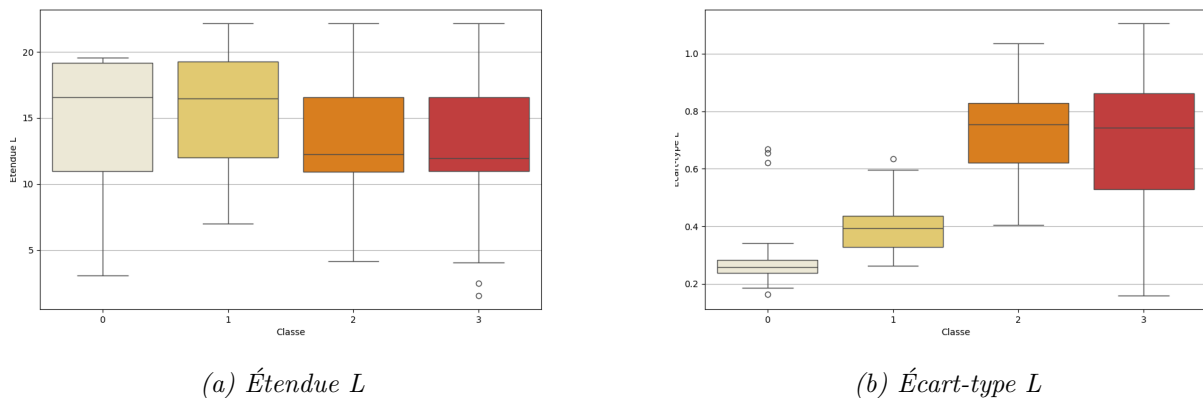


Figure 1. Exemple de descripteur inintéressant qui bruite la classification (a), et d'un autre qui discrimine les classes (b)

On utilisera également l'évaluation par boîtes à moustaches pour déterminer quels descripteurs sont les plus **discriminants**, et lesquels ne font que **bruiter** la classification. L'objectif étant de ne garder les caractéristiques qui permettent réellement de distinguer les classes les unes des autres. Les boîtes à moustaches nous offrent une **visualisation claire de la distribution** des valeurs au sein de chaque classe, ce qui facilite l'analyse et l'identification des caractéristiques les plus pertinentes.

Après la classification par l'algorithme k-NN ou CNN, il s'agit d'évaluer la **justesse des étiquetages** à partir des cartes de confusion. Celles-ci nous permettront d'obtenir des statistiques intéressantes pour juger la **précision** des classifications, notamment le **F₁ score**, et le taux de réussite moyen. Les matrices de confusion sont particulièrement utiles car elles présentent une vue complète des performances du modèle, indiquant non seulement les classifications correctes, mais aussi la nature des erreurs commises, révélant ainsi les confusions potentielles entre certaines classes.

Pour déterminer quels paramètres sont les plus **optimaux** pour l'algorithme de classification. Dans le cas du k-NN on établira cette carte de confusion pour des différents k , et des ratio d'*entraînement/évaluation* différents. Et pour le CNN on réalisera les **cartes de confusion** pour des enrichissements différents et des ratio d'*entraînement/évaluation* différents. Cette approche méthodique nous permettra d'identifier avec précision la configuration la plus performante pour chaque type d'algorithme.

1. k-NN

L'analyse du taux de réussite moyen des 9 matrices de confusion calculées avec $k \in \{3, 5, 7\}$ et ratio $\in \{90/10, 80/20, 70/30\}$ (voir Annexes) nous oriente vers un k-NN où $k = 5$ et ratio = 90/10. Cette combinaison offre le meilleur **équilibre entre généralisation et précision**, évitant à la fois le sur-apprentissage (risque avec k trop petit) et la perte de spécificité locale (risque avec k trop grand).

Dans cette configuration, on obtient un taux de Vrais Positifs de **75,99%**, ce qui signifie que plus des trois quarts des marshmallows sont correctement classifiés selon leur niveau de cuisson. Ce résultat est **encourageant** compte tenu de la nature **subjective** des catégories de cuisson et des **subtilités visuelles** qui les distinguent.

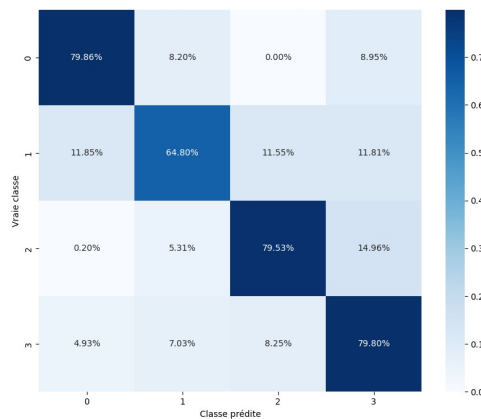


Figure 2. Matrice de confusion : $k = 5$, ratio = 90/10

Le **F₁ score** combine les taux de Vrais Positifs, Faux Positifs et Faux Négatifs. Il permet de s'assurer que les **bonnes prédictions** sont présentes en **grand nombre** dans la classe pertinente. Cette métrique est particulièrement adaptée aux problèmes de classification où les classes peuvent être déséquilibrées, car elle prend en compte à la fois la **précision** (proportion de prédictions positives correctes) et le **rappel** (proportion de positifs réels correctement identifiés).

$$F_1 = 2 \times \frac{(\text{précision} \times \text{rappel})}{(\text{précision} + \text{rappel})} \text{ où } \text{précision} = \frac{VP}{VP + FP} \text{ et } \text{rappel} = \frac{VP}{VP + FN}$$

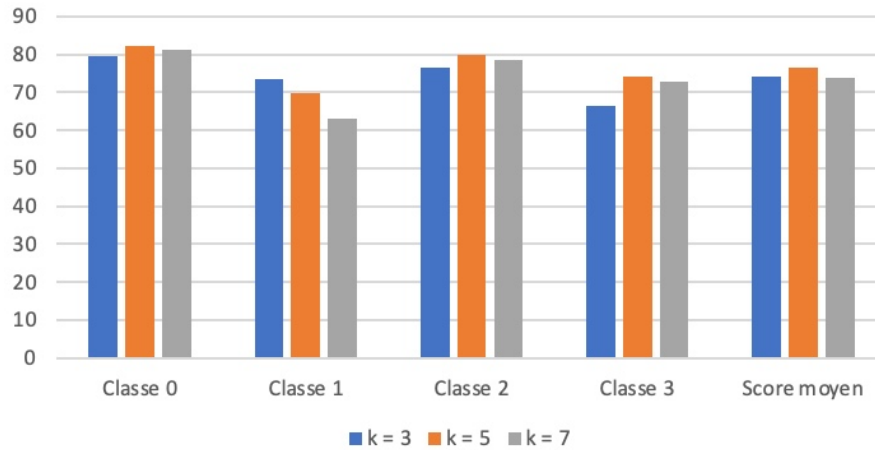


Figure 3. F_1 score pour chaque classe avec un ratio 90/10(%)

En particulier pour $k = 5$ et ratio = 90/10 :

Le calcul du F_1 score micro (*intra-classe*) donne les résultats suivants :

- classe 0 : 82,39% (marshmallows non cuits)
- classe 1 : 69,92% (marshmallows peu cuits)
- classe 2 : 79,80% (marshmallows bien cuits)
- classe 3 : 74,05% (marshmallows trop cuits)

On calcule également le F_1 score macro qui correspond à la moyenne des scores de chaque classe:

- F_1 score moyen = 76,54%

Ces résultats sont donc cohérents, les meilleures classifications sont obtenues pour des ratio de **90/10** d'entraînement puisque l'algorithme de classification possède plus de données d'entraînement et moins de données d'essais. Cette proportion favorise un **apprentissage plus complet des caractéristiques** des différentes classes, permettant au modèle de mieux généraliser lors de la phase de test. La classification est **optimale** pour un $k = 5$ puisqu'il s'agit d'un nombre suffisant de voisins pour correctement effectuer la classification **sans trop être bruité** comme le devient le cas du $k = 7$. Un k plus élevé tend à lisser les frontières de décision, ce qui peut être préjudiciable lorsque ces frontières sont intrinsèquement complexes comme dans notre cas.

Le F_1 score moyen de **76%** est également satisfaisant, la classification est globalement correcte mais pas parfaite. Ceci peut venir de divers facteurs, il est possible que les images dans la base de données soient **semblables** et sont donc plus difficiles à classer de façon adéquate, ou que les **caractéristiques choisies** ne sont pas suffisamment bonnes pour atteindre un score de classification plus élevé. Notons que la **classe 1** (peu cuit) présente le F_1 score **le plus faible (69,92%)**, suggérant une plus grande difficulté à distinguer les marshmallows légèrement cuits des autres catégories, probablement en raison de la **subtilité** des changements visuels dans les premiers stades de cuisson.

2. CNN

Après 15 tests réalisés sur des enrichissement $\in \{1, 5, 10, 20, 30\}$ et ratio $\in \{90/10, 80/20, 70/30, 60/40, 50/50\}$ différents (*voir Annexes*), nous constatons les meilleures performances pour un **enrichissement x30** et pour un **ratio 60/40**, jugé pertinent pour une application au CNN. En effet, comme notre base est restreinte, nous préférons faire de l'enrichissement sur moins de valeurs d'entraînement. L'enrichissement des données (data augmentation) est une technique

cruciale qui permet d'**augmenter** artificiellement la taille du jeu de données d'entraînement en appliquant diverses transformations aux images originales, telles que des rotations, des retournements horizontaux ou verticaux, et des changements d'échelle. Cette approche **améliore la robustesse** du modèle et **réduit le risque de surapprentissage**.

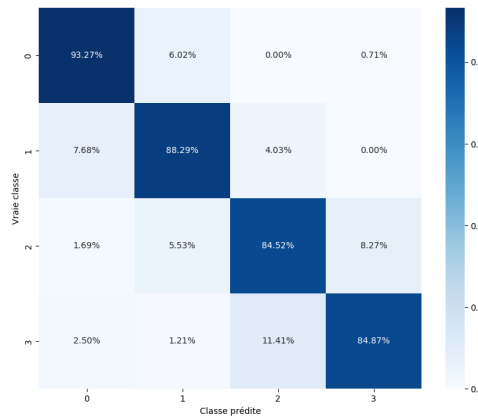


Figure 4. Carte de confusion : enrichissement = 30, ratio = 60/40

Avec ces paramètres, on obtient un taux de réussite moyen de **87,74%**, ce qui représente une amélioration **significative** par rapport au k-NN. Cette performance supérieure s'explique par plusieurs facteurs. En effet, en comparaison avec le k-NN, le CNN est moins sensible au **biais** introduit par le choix manuel des caractéristiques et comporte un nombre important de **paramètres ajustables** (poids des connexions entre neurones) qui lui permettent de modéliser des relations complexes et non linéaires dans les données. Cette architecture s'avère donc bien plus sophistiquée que la simple utilisation des **11 caractéristiques** de couleur et textures que nous avons en entrée du k-NN.

Calcul du F_1 score global :

- F_1 score moyen = 87,72%

Ce score, nettement supérieur à celui obtenu avec le k-NN, confirme la capacité du CNN à mieux discriminer entre les différentes classes de marshmallows. Il est important de noter que ce score élevé a été obtenu malgré une proportion de données d'entraînement plus faible (60% contre 90% pour le k-NN), ce qui souligne davantage l'**efficacité** de l'approche CNN **combinée à l'enrichissement des données**.

La figure 5 illustre l'**impact** des différents niveaux d'enrichissement et des ratios d'entraînement/test sur les performances du CNN. On observe clairement que l'enrichissement des données améliore les performances du modèle, avec un pic pour l'**enrichissement x30** et un **ratio de 60/40**. Cette combinaison offre le meilleur équilibre entre la quantité de données disponibles pour l'entraînement et la diversité introduite par l'enrichissement. L'amélioration progressive des performances avec l'augmentation du niveau d'enrichissement confirme l'importance de cette technique dans le contexte de notre base de données **limitée**.

Les CNN se montrent particulièrement efficaces pour notre tâche de classification de marshmallows, surpassant significativement le k-NN grâce à leur capacité à extraire automatiquement des caractéristiques pertinentes à partir des images brutes, sans nécessiter une ingénierie manuelle des descripteurs. Cette supériorité est d'autant plus remarquable que la tâche implique la distinction de nuances subtiles de texture et de couleur qui caractérisent les différents niveaux de cuisson des marshmallows.

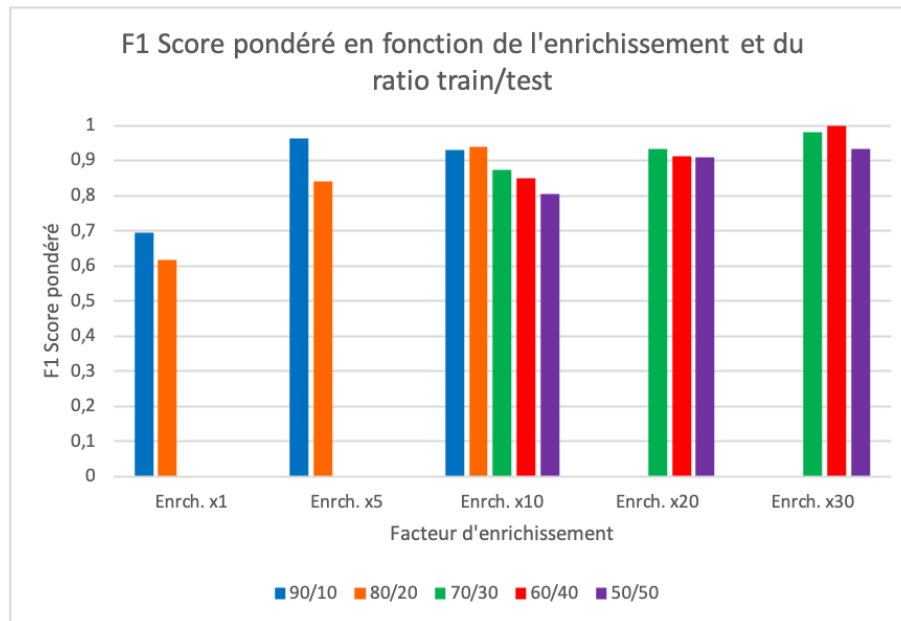


Figure 5. Comparaison des F_1 Scores Pondérés en fonction de l'Enrichissement et du Ratio d'entraînement/test

III. Conclusion

Nos données sont **équilibrées** comme le confirme l'excellente entropie de Shannon (0.993), et nous constatons un bon taux de classification avec les deux méthodes, cependant quelques difficultés pour l'identification des marshmallows de **classe 1** ("peu cuit") pour le k-NN. Cette différence peut s'expliquer par une **frontière plus étroite** avec les classes voisines et une base de données annotée de manière très **subjective**. Les marshmallows peu cuits présentent des caractéristiques visuelles intermédiaires qui les rendent plus difficiles à distinguer des marshmallows non cuits (classe 0) ou bien cuits (classe 2).

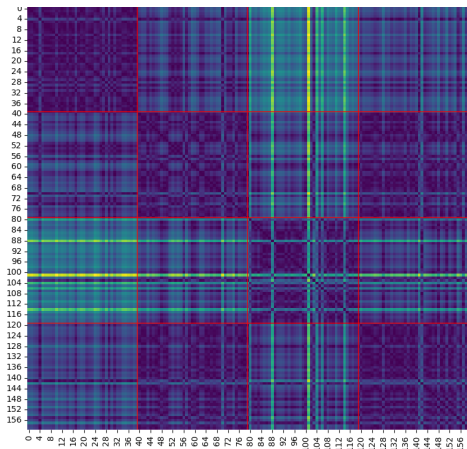
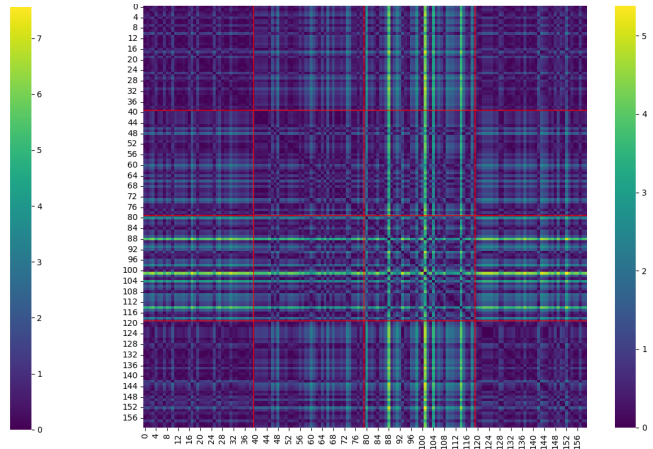
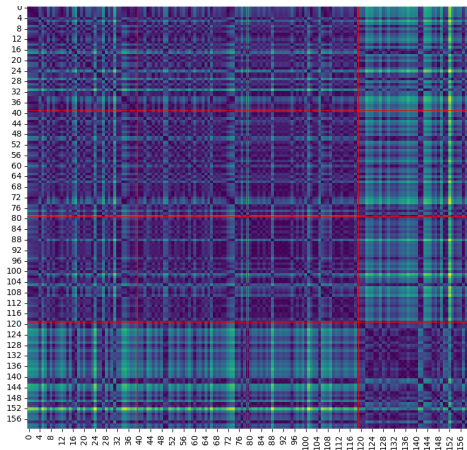
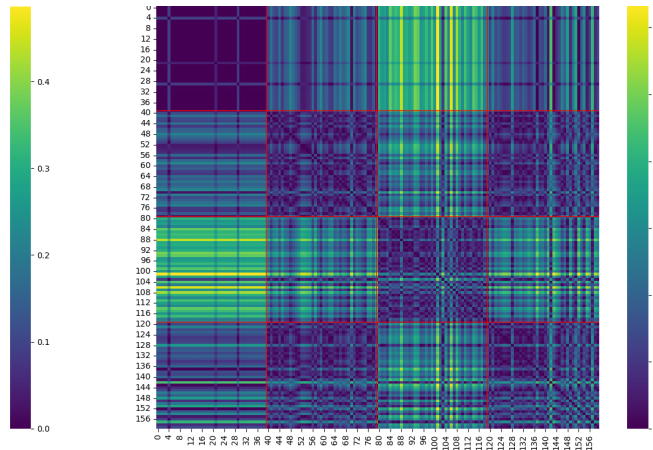
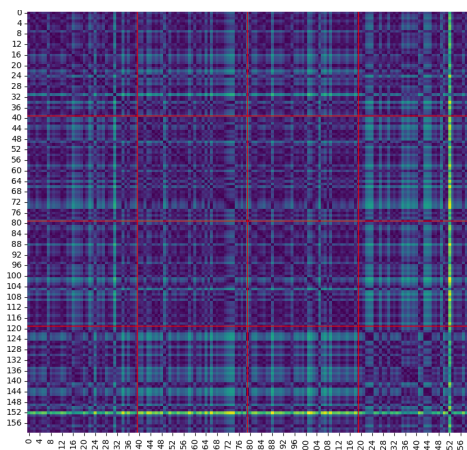
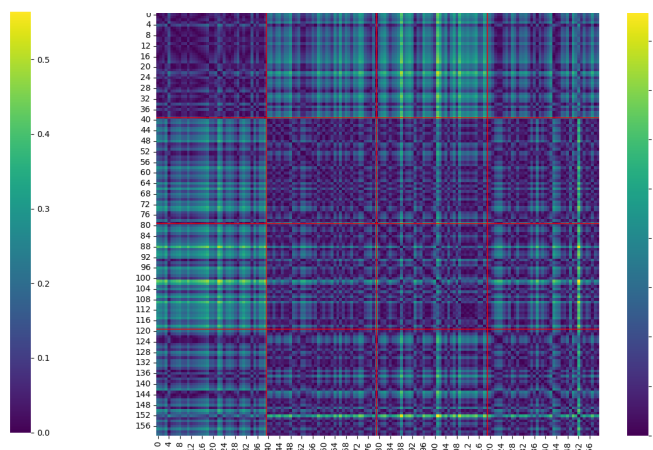
En outre, le CNN est beaucoup plus efficace sans atteindre les 100% de réussite souvent synonymes de **sur-apprentissage**. Cette performance supérieure s'explique par sa capacité à capturer des caractéristiques complexes et hiérarchiques des images, ainsi que par l'efficacité de l'enrichissement des données pour améliorer la généralisation du modèle.

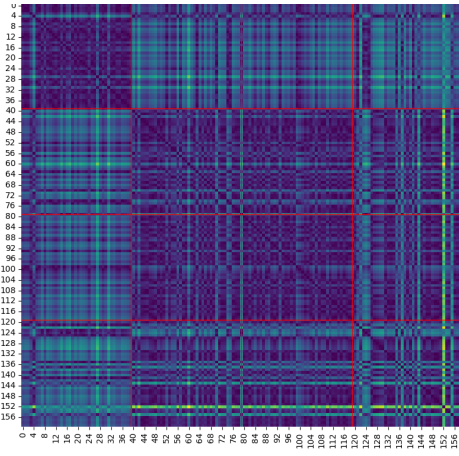
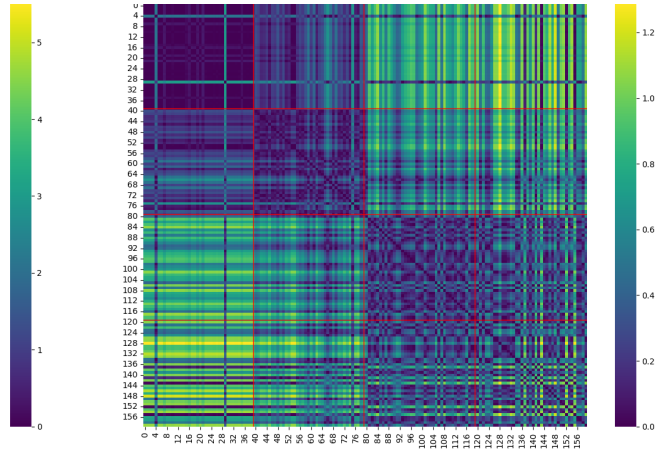
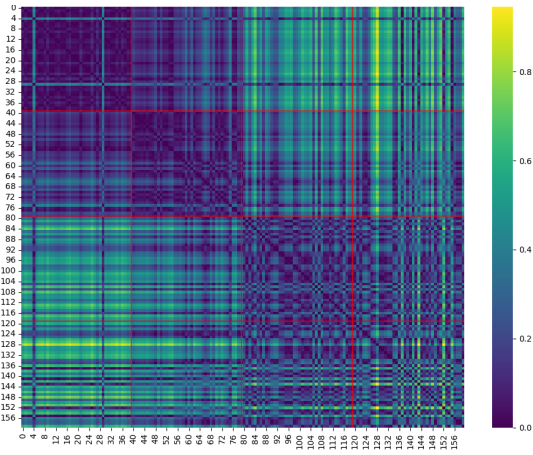
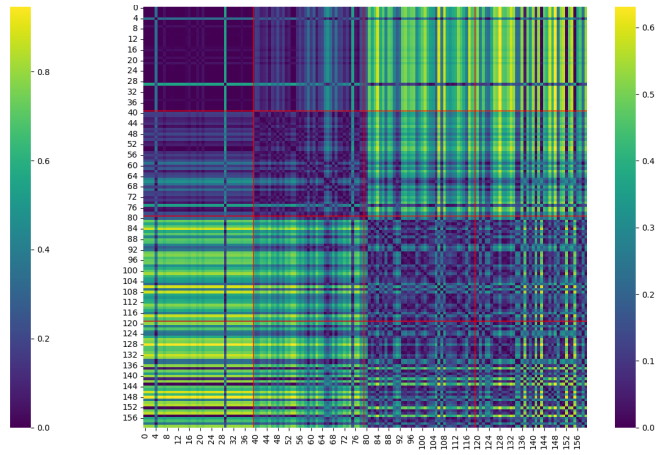
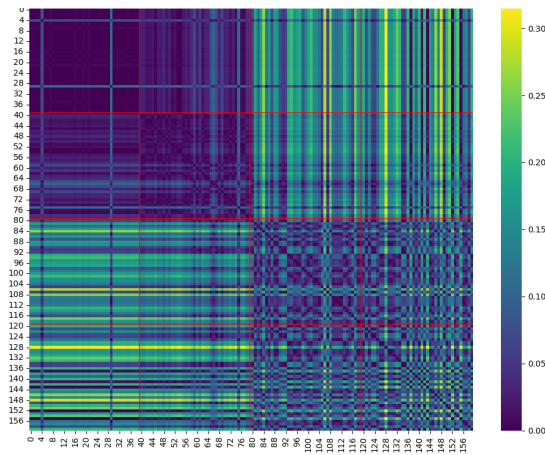
On atteint les limites de notre programme :

A défaut de trouver une base de données existante, nous avons réalisé les acquisitions nous-mêmes et ainsi nous sommes restreints à **182 images**. Une base plus complète permettrait d'améliorer les résultats de classification, particulièrement pour les cas subtils comme les marshmallows peu cuits. De plus, il est possible que les conditions d'éclairage soient un peu différentes entre les images. Cela peut jouer aussi sur les performances des algorithmes, introduisant des variations qui ne sont pas directement liées au niveau de cuisson mais plutôt aux **conditions d'acquisition**.

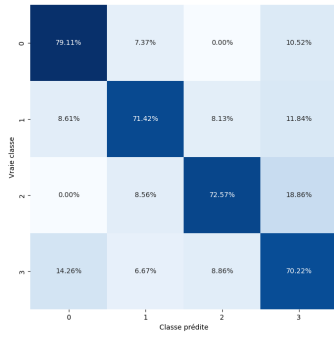
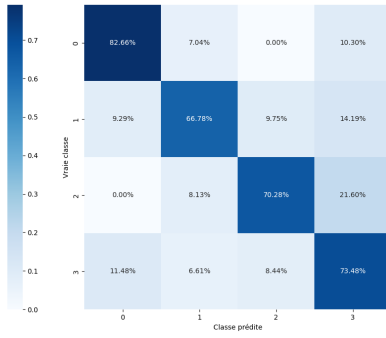
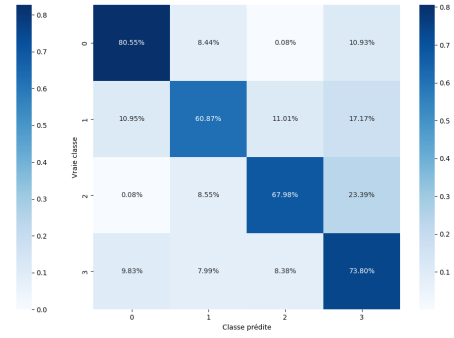
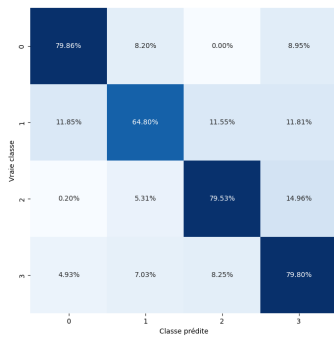
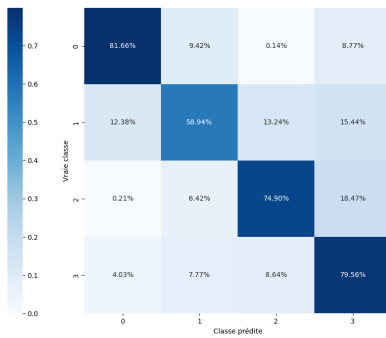
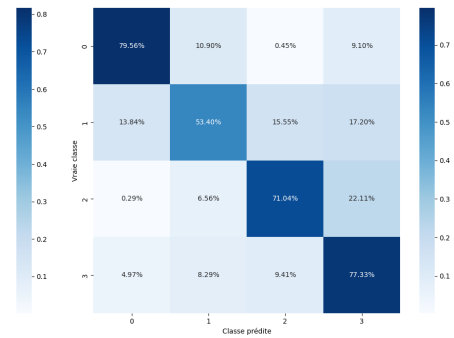
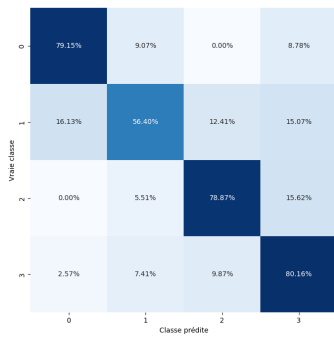
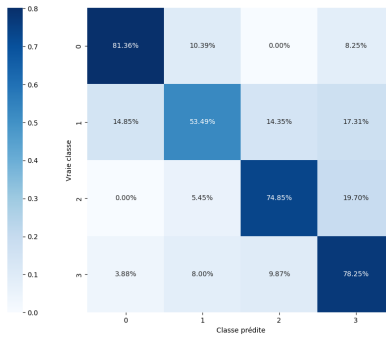
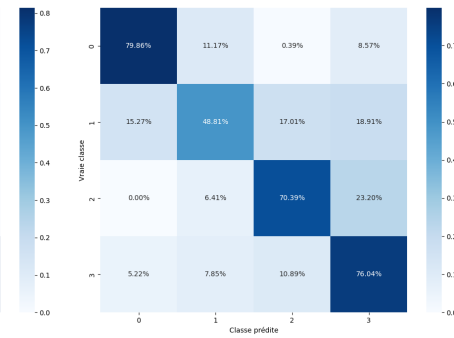
IV. Annexes

1. Matrices de distance des descripteurs

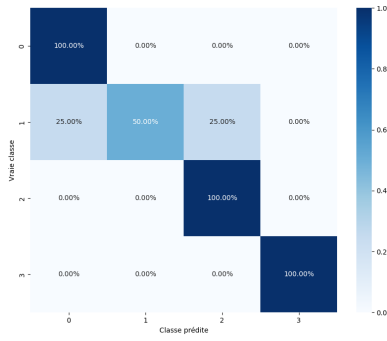
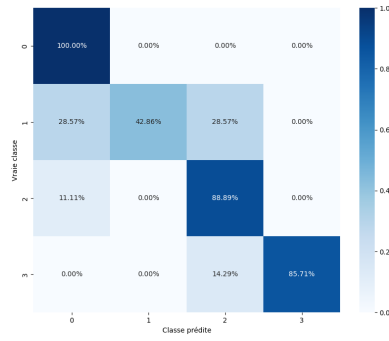
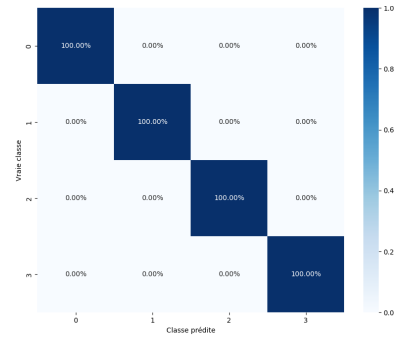
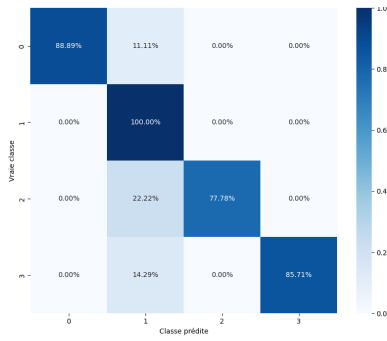
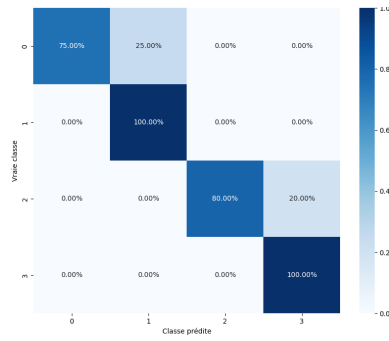
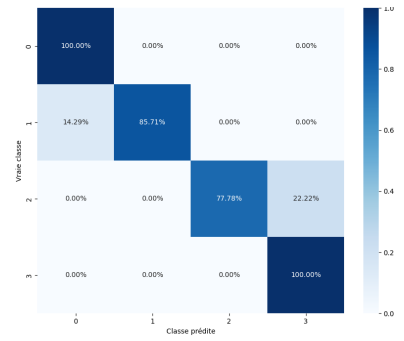
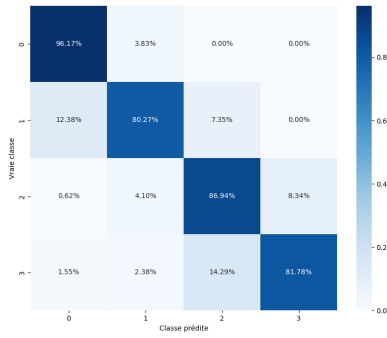
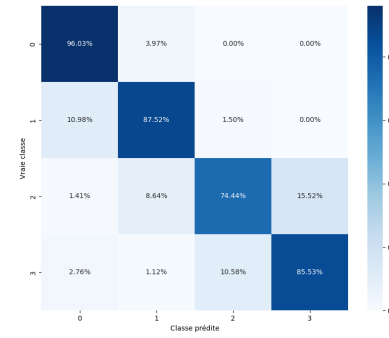
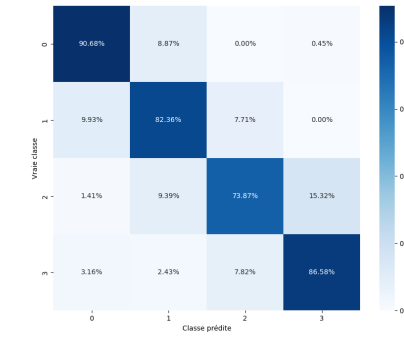
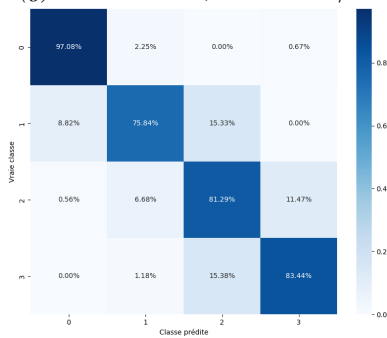
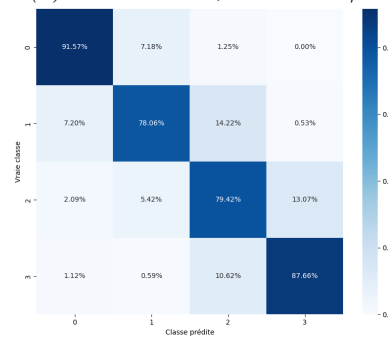
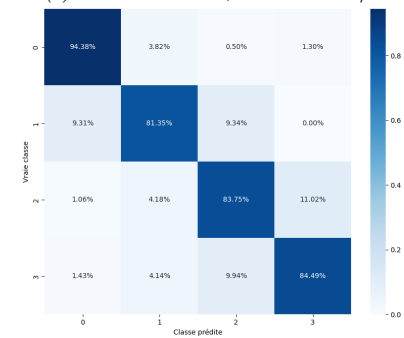
(a) Moyenne b (b) Écart-type b (c) Homogénéité b (d) Contraste b (e) Énergie b (f) Entropie b

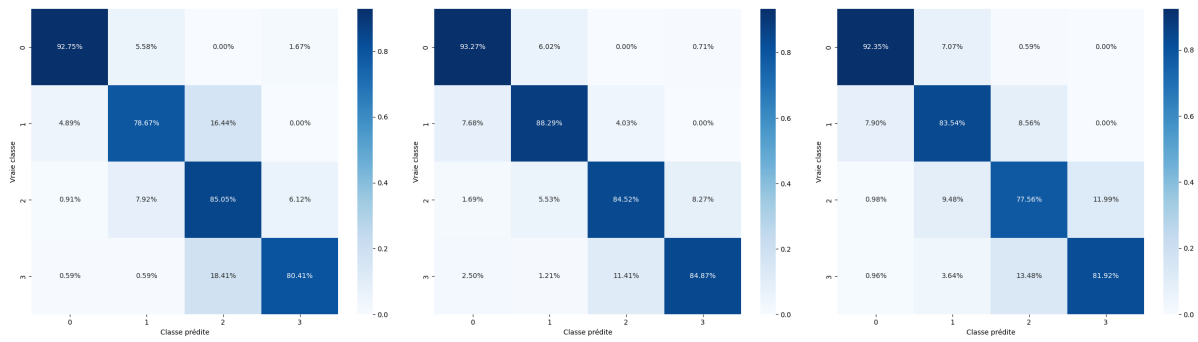
(a) Skewness b (b) Entropie L (c) Écart-type L (d) Énergie L (e) Homogénéité L

2. Cartes de confusion - k-NN

(a) $k = 3$, ratio = 90/10(b) $k = 3$, ratio = 80/20(c) $k = 3$, ratio = 70/30(d) $k = 5$, ratio = 90/10(e) $k = 5$, ratio = 80/20(f) $k = 5$, ratio = 70/30(g) $k = 7$, ratio = 90/10(h) $k = 7$, ratio = 80/20(i) $k = 7$, ratio = 70/30

3. Cartes de confusion - CNN

(a) *enrich. = 1, ratio = 90/10*(b) *enrich. = 1, ratio = 80/20*(c) *enrich. = 5, ratio = 90/10*(d) *enrich. = 5, ratio = 80/20*(e) *enrich. = 10, ratio = 90/10*(f) *enrich. = 10, ratio = 80/20*(g) *enrich. = 10, ratio = 70/30*(h) *enrich. = 10, ratio = 60/40*(i) *enrich. = 10, ratio = 50/50*(j) *enrich. = 20, ratio = 70/30*(k) *enrich. = 20, ratio = 60/40*(l) *enrich. = 20, ratio = 50/50*

(a) $enrich. = 30, ratio = 70/30$ (b) $enrich. = 30, ratio = 60/40$ (c) $enrich. = 30, ratio = 50/50$