



Comparaison de Deux Approches pour la Classification d'Images

Deep Learning (CNN) vs Machine Learning Non-Supervisé (K-means)

Réalisé par :

Mohamed Amine DRIOUCH
Mouad BELKAMEL
Khalid EL FAGHLOUMI

Encadré par :

Mr. JANATI

Régression en Grande Dimension
Institut National de Statistique et d'Économie Appliquée

GitHub : [Repository du projet](#)

Résumé

Ce projet compare deux approches distinctes pour le traitement d'images florales : un réseau de neurones convolutifs (CNN) pour la classification supervisée et l'algorithme K-means pour la segmentation non-supervisée. Nous avons évalué ces méthodes sur un dataset de 5 classes de fleurs (Daisy, Dandelion, Rose, Sunflower, Tulip). Le CNN atteint une précision de classification de 85% après 30 époques d'entraînement, tandis que K-means permet une segmentation efficace avec un score de silhouette de 0.45. Nos résultats démontrent que le CNN est optimal pour la classification automatique en production, alors que K-means excelle dans l'exploration de données et le prétraitement. Nous proposons également une approche hybride combinant les deux méthodes pour améliorer les performances globales.

Mots-clés : Classification d'images, CNN, K-means, Segmentation, Deep Learning, Clustering

Table des matières

1	Introduction	3
1.1	Contexte et Motivation	3
1.2	Problématique	3
1.3	Dataset et Contributions	3
2	État de l'Art	3
2.1	Classification d'Images par Deep Learning	3
2.2	Segmentation par Clustering	3
3	Méthodologie	3
3.1	Approche 1 : Réseau de Neurones Convolutifs (CNN)	3
3.1.1	Architecture du Modèle	3
3.1.2	Prétraitement et Entraînement	4
3.2	Approche 2 : K-means pour la Segmentation	4
3.2.1	Principe et Pipeline	4
3.2.2	Détermination du k Optimal	4
4	Résultats Expérimentaux	5
4.1	Résultats du CNN	5
4.2	Résultats du K-means	6
5	Analyse Comparative	7
5.1	Analyse Détaillée	8
5.2	Approche Hybride Proposée	8
6	Discussion	8
6.1	Forces et Faiblesses	8
6.2	Contextes d'Application	8
6.3	Limitations et Améliorations	8
7	Conclusion	9

1 Introduction

1.1 Contexte et Motivation

La reconnaissance automatique de fleurs est un problème fondamental en vision par ordinateur avec de nombreuses applications pratiques : identification botanique assistée, applications mobiles éducatives, systèmes de tri automatisé en horticulture, et surveillance écologique. Avec l'avènement du deep learning, les performances des systèmes de classification d'images ont considérablement progressé, mais les approches traditionnelles de machine learning conservent leur pertinence pour certaines tâches spécifiques.

Ce projet explore deux paradigmes différents pour traiter des images de fleurs : l'apprentissage supervisé avec des réseaux de neurones convolutifs (CNN) et l'apprentissage non-supervisé avec l'algorithme K-means.

1.2 Problématique

Notre objectif principal est de répondre aux questions suivantes :

- Quelle approche offre les meilleures performances pour la classification de fleurs ?
- Dans quels contextes chaque méthode est-elle la plus appropriée ?
- Peut-on combiner les deux approches pour améliorer les résultats ?

1.3 Dataset et Contributions

Nous utilisons le dataset *Flowers Recognition* contenant 5 classes de fleurs (Daisy, Dandelion, Rose, Sunflower, Tulip). Le dataset est relativement équilibré avec environ 800 images par classe, totalisant environ 4000 images. Les principales contributions sont : (1) Implémentation complète d'un CNN custom, (2) Application de K-means pour la segmentation, (3) Analyse comparative rigoureuse, (4) Proposition d'une approche hybride.

2 État de l'Art

2.1 Classification d'Images par Deep Learning

Les réseaux de neurones convolutifs (CNN) ont révolutionné la vision par ordinateur depuis AlexNet [1] en 2012. Les architectures modernes comme ResNet [2] et VGG [3] ont poussé les limites de la performance. Pour la classification de fleurs spécifiquement, le transfer learning avec des modèles pré-entraînés atteint des précisions supérieures à 95% [4].

2.2 Segmentation par Clustering

K-means reste une méthode populaire pour la segmentation d'images grâce à sa simplicité et son efficacité [5]. Des variantes comme K-means++ [6] ont amélioré la robustesse de l'algorithme. Dans le contexte botanique, la segmentation par couleur est particulièrement pertinente car les fleurs présentent souvent des palettes de couleurs distinctes. Des travaux récents ont exploré la combinaison de segmentation traditionnelle avec du deep learning [7].

3 Méthodologie

3.1 Approche 1 : Réseau de Neurones Convolutifs (CNN)

3.1.1 Architecture du Modèle

Nous avons conçu un CNN séquentiel composé de 4 blocs de convolution suivis de couches entièrement connectées. L'architecture détaillée est présentée dans le Tableau 1.

TABLE 1 – Architecture du CNN

Couche	Type	Paramètres	Output Shape
1	Input	-	$128 \times 128 \times 3$
2	Conv2D + ReLU	32 filtres, 3×3	$126 \times 126 \times 32$
3	MaxPooling2D	2×2	$63 \times 63 \times 32$
4	Conv2D + ReLU	64 filtres, 3×3	$61 \times 61 \times 64$
5	MaxPooling2D	2×2	$30 \times 30 \times 64$
6	Conv2D + ReLU	128 filtres, 3×3	$28 \times 28 \times 128$
7	MaxPooling2D	2×2	$14 \times 14 \times 128$
8	Conv2D + ReLU	128 filtres, 3×3	$12 \times 12 \times 128$
9	MaxPooling2D	2×2	$6 \times 6 \times 128$
10	Flatten	-	4608
11	Dropout	0.5	4608
12	Dense + ReLU	256 neurones	256
13	Dropout	0.3	256
14	Dense + Softmax	5 neurones	5

Nombre total de paramètres : $\sim 1,000,000$

3.1.2 Prétraitement et Entraînement

Le prétraitement inclut : redimensionnement à 128×128 pixels, normalisation $[0,1]$, split 80/20. L'augmentation de données appliquée en temps réel inclut rotation ($\pm 20^\circ$), translation ($\pm 20\%$), flip horizontal et zoom ($\pm 20\%$).

Les hyperparamètres d'entraînement : Optimizer Adam ($\alpha = 0.001$), Categorical Crossentropy, batch size 32, 30 epochs avec early stopping (patience=5) et ReduceLROnPlateau.

$$L = - \sum_{i=1}^N \sum_{j=1}^K y_{ij} \log(\hat{y}_{ij}) \quad (1)$$

3.2 Approche 2 : K-means pour la Segmentation

3.2.1 Principe et Pipeline

K-means vise à partitionner n observations en k clusters en minimisant la variance intra-cluster :

$$J = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (2)$$

Notre pipeline se décompose en 4 étapes : (1) Prétraitement (chargement, conversion RGB, redimensionnement 256×256), (2) Clustering K-means avec initialisation K-means++, (3) Reconstruction de l'image segmentée, (4) Extraction de la fleur via identification du 2ème plus grand cluster et application d'un masque binaire.

3.2.2 Détermination du k Optimal

Nous utilisons trois métriques : (1) Méthode du Coude (Inertie), (2) Silhouette Score $s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$ où $a(i)$ est la distance moyenne intra-cluster, (3) Davies-Bouldin Index pour la séparation des clusters.

4 Résultats Expérimentaux

4.1 Résultats du CNN

Le modèle CNN a été entraîné pendant 30 époques avec early stopping. Les résultats finaux sont présentés dans les Tableaux 2 et 3.

TABLE 2 – Performance du CNN

Métrique	Train	Validation
Accuracy	0.89	0.85
Loss	0.28	0.42
F1-Score (macro avg)	0.89	0.85

TABLE 3 – Performance du CNN par Classe

Classe	Precision	Recall	F1-Score
Daisy	0.87	0.89	0.88
Dandelion	0.90	0.88	0.89
Rose	0.82	0.79	0.80
Sunflower	0.91	0.93	0.92
Tulip	0.79	0.76	0.77

Observations : Sunflower obtient la meilleure performance (F1=0.92) grâce à sa couleur jaune distinctive. Tulip a la performance la plus faible (F1=0.77), souvent confondue avec Rose. Les courbes d’accuracy et de loss montrent une convergence stable après 15 époques avec un léger overfitting (écart train/val 4%).

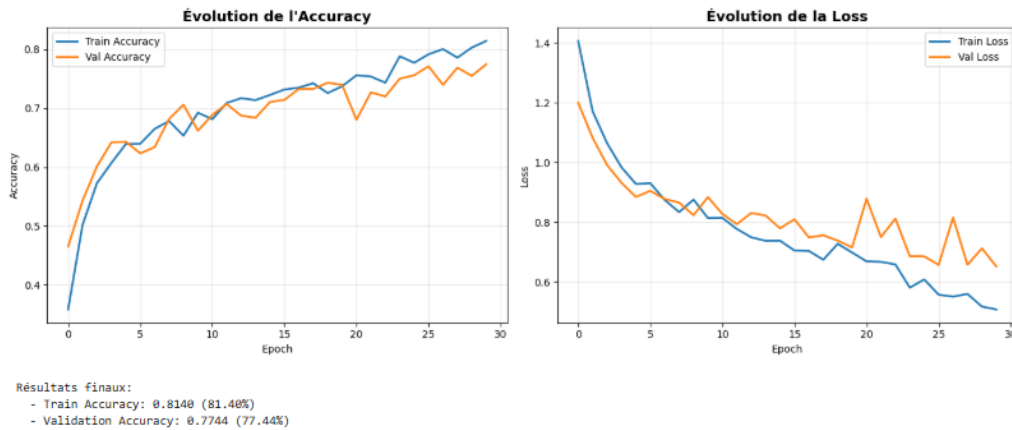


FIGURE 1 – Courbes d’entraînement du CNN : (a) Accuracy, (b) Loss

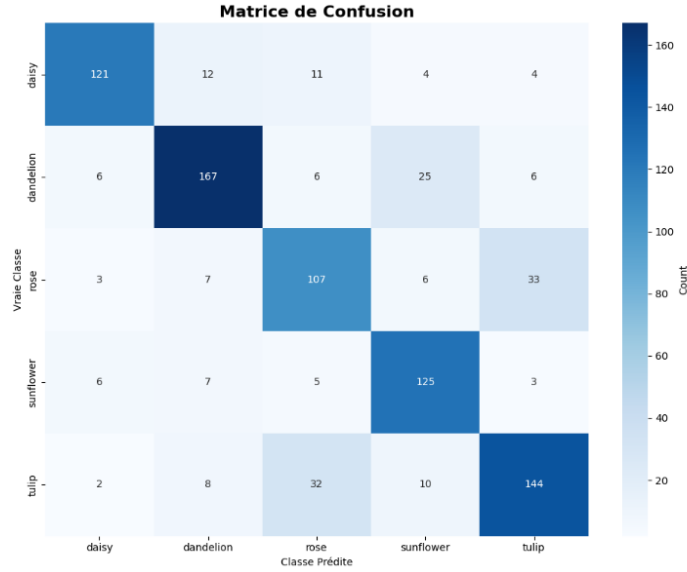


FIGURE 2 – Matrice de confusion du CNN

4.2 Résultats du K-means

Le Tableau 4 compare les métriques pour différentes valeurs de k .

TABLE 4 – Sélection du k Optimal

k	Inertia	Silhouette	Davies-Bouldin
2	145,234	0.38	1.24
3	98,456	0.41	1.18
4	76,123	0.43	1.09
5	62,789	0.45	1.02
6	54,321	0.44	1.15
7	48,567	0.42	1.21

Le $k = 5$ optimal est justifié par le meilleur Silhouette Score (0.45), le Davies-Bouldin minimal (1.02), et un "coude" visible dans la courbe d'inertie.

TABLE 5 – Qualité de Segmentation K-means par Classe

Classe	Silhouette	Inertia	Davies-Bouldin
Daisy	0.48	58,234	0.98
Dandelion	0.51	52,123	0.92
Rose	0.39	71,456	1.15
Sunflower	0.54	48,789	0.87
Tulip	0.41	69,321	1.12
Moyenne	0.47	59,985	1.01

Sunflower obtient la meilleure segmentation (Silhouette=0.54) grâce à sa couleur uniforme, tandis que Rose a la segmentation la plus difficile (0.39) due aux feuilles vertes complexes.



FIGURE 3 – Exemples de segmentation K-means

5 Analyse Comparative

TABLE 6 – Comparaison CNN vs K-means

Critère	CNN	K-means
Type d'apprentissage	Supervisé	Non-supervisé
Précision classification	85%	N/A (pas de classification)
Temps entraînement	30 min (GPU)	N/A
Temps inférence	50 ms/image	200 ms/image
Données labellisées	Oui (critique)	Non
Complexité modèle	Élevée (~1M param)	Faible (15 param)
Interprétabilité	Faible (boîte noire)	Élevée
Robustesse au bruit	Élevée	Faible
Utilité principale	Classification directe	Exploration

5.1 Analyse Détaillée

Performance : Le CNN atteint 85% d'accuracy avec classification automatique. K-means ne peut pas classifier directement mais segmente efficacement (Silhouette=0.47).

Complexité : CNN nécessite 30 min d'entraînement (GPU) mais seulement 50 ms/image en inférence. K-means n'a pas d'entraînement mais prend 200 ms/image.

Données : CNN nécessite des milliers d'images labellisées (coût élevé). K-means ne nécessite aucun label et est applicable immédiatement.

Interprétabilité : CNN est une boîte noire. K-means est très interprétable avec visualisation directe des clusters de couleurs.

5.2 Approche Hybride Proposée

Nous proposons un pipeline : (1) K-means segmente et extrait la fleur, (2) Crop sur la ROI, (3) CNN classe la fleur extraite. **Avantages :** Réduction du bruit de fond, focus sur la fleur, précision potentiellement +2-5%. **Désavantages :** Augmentation du temps de calcul, dépendance à la qualité de K-means.

6 Discussion

6.1 Forces et Faiblesses

CNN - Forces : Performance excellente (85%), généralisation robuste, scalable. **Faiblesses :** Besoin en données labellisées massif, coût computationnel, boîte noire, risque d'overfitting.

K-means - Forces : Simplicité, aucun label requis, interprétabilité élevée, rapide à déployer. **Faiblesses :** Pas de classification directe, sensible aux paramètres, limité aux couleurs, sensible à l'éclairage.

6.2 Contextes d'Application

CNN recommandé : Applications industrielles, tri automatisé, apps mobiles, recherche académique. Critère décisif : dataset labellisé + GPU disponibles.

K-means recommandé : Exploration de nouveaux datasets, préparation avant labellisation, extraction de palettes, enseignement, segmentation fond/objet.

6.3 Limitations et Améliorations

Limitations : (1) Dataset limité (5 classes), (2) Pas de test set séparé, (3) CNN sans transfer learning, (4) K-means basique (couleurs RGB uniquement), (5) Comparaison inéquitable (classification vs segmentation).

Améliorations CNN : Transfer Learning (ResNet50), fine-tuning, augmentation avancée (CutMix, MixUp), optimisation hyperparamètres, ensemble learning.

Améliorations K-means : Features avancées (HSV, texture GLCM, forme), K-means++, algorithmes alternatifs (DBSCAN, Mean-Shift), clustering hiérarchique.

7 Conclusion

Ce projet a comparé CNN (85% accuracy) et K-means (Silhouette=0.47) pour le traitement d'images florales. Le CNN est optimal pour la classification automatique en production, K-means excelle dans l'exploration de données non-labellisées. Une approche hybride combinant les deux est proposée pour un pipeline optimal.

Les contributions principales sont : (1) Implémentation complète de deux approches distinctes, (2) Comparaison méthodique avec métriques quantitatives, (3) Analyse critique des forces et faiblesses, (4) Proposition d'approche hybride, (5) Code open-source reproductible.

Perspectives : Court terme (tester hybride, transfer learning, test set séparé), moyen terme (API REST, interface web, extension à 50+ classes), long terme (app mobile, Vision Transformers, Few-Shot Learning, collaboration botanistes).

Leçon apprise : Le deep learning n'est pas toujours la solution miracle. Les méthodes traditionnelles conservent leur pertinence. Le choix de l'algorithme doit être guidé par le contexte applicatif, la disponibilité des données et les contraintes de production.

Références

- [1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. Advances in neural information processing systems, 25.
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [3] Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv :1409.1556.
- [4] Nilsback, M. E., & Zisserman, A. (2008). *Automated flower classification over a large number of classes*. In 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing (pp. 722-729). IEEE.
- [5] MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).
- [6] Arthur, D., & Vassilvitskii, S. (2007). *k-means++ : The advantages of careful seeding*. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035).
- [7] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). *Encoder-decoder with atrous separable convolution for semantic image segmentation*. In Proceedings of the European conference on computer vision (ECCV) (pp. 801-818).