

[208, 12
[207, 12
[210, 12

[[112, 14
[127, 15
[145, 17

LA CLASSIFICATION D'IMAGES

MOHAMED ABI & HACINE HOUSSAM

REPONSES AUX QUESTIONS

CRÉATION DE LA BASE DE DONNÉES :

01

MNIST (MODIFIED NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY): IL S'AGIT D'UN ENSEMBLE DE DONNÉES COMPRENANT 70 000 IMAGES EN NOIR ET BLANC DE CHIFFRES MANUSCRITS DE 0 À 9. CET ENSEMBLE EST LARGEMENT UTILISÉ EN APPRENTISSAGE AUTOMATIQUE COMME RÉFÉRENCE POUR ÉVALUER LES PERFORMANCES DES MODÈLES DE CLASSIFICATION, EN RAISON DE SA TAILLE MODESTE ET DE SA SIMPLICITÉ.

02

FORMAT DES DONNÉES (X_TRAIN ET X_TEST): LES VARIABLES X_TRAIN ET X_TEST CONTIENNENT DES IMAGES REPRÉSENTÉES SOUS FORME DE MATRICES. CHAQUE IMAGE EST UNE MATRICE 28X28, OÙ CHAQUE ÉLÉMENT DE LA MATRICE CORRESPOND À LA VALEUR DU PIXEL EN NIVEAUX DE GRIS, NORMALISÉE DANS LA PLAGE [0, 1].

03

ÉTIQUETAGE DES IMAGES (Y_TRAIN ET Y_TEST): LES VARIABLES Y_TRAIN ET Y_TEST REPRÉSENTENT LES ÉTIQUETTES DES IMAGES D'ENTRAÎNEMENT ET DE TEST. CHAQUE ÉTIQUETTE EST UN CHIFFRE DE 0 À 9, INDiquant LE CHIFFRE MANUSCRIT REPRÉSENTÉ DANS L'IMAGE. LES ÉTIQUETTES SONT ENCODÉES SOUS FORME DE NOMBRES ENTIERS.

04

DIMENSIONS DES DONNÉES (X_TRAIN ET X_TEST): LES DIMENSIONS DE X_TRAIN SONT (NOMBRE D'IMAGES D'ENTRAÎNEMENT, 28, 28), ET CELLES DE X_TEST SONT (NOMBRE D'IMAGES DE TEST, 28, 28). CHAQUE IMAGE EST AINSI REPRÉSENTÉE PAR UNE MATRICE 28X28, REFLÉTANT LES DIMENSIONS DES IMAGES DE CHIFFRES MANUSCRITS EN NOIR ET BLANC.

CHARGEMENT DES DONNÉES :

01

LA FONCTION `LOAD_DATA` UTILISÉE PROVIENT GÉNÉRALEMENT DES BIBLIOTHÈQUES COMME TENSORFLOW OU KERAS, PAR EXEMPLE `TF.KERAS.DATASETS.MNIST.LOAD_DATA()`. ELLE PEUT ACCEPTER D'AUTRES PARAMÈTRES, COMME PATH. RÉUSSIR LE CHARGEMENT DES DONNÉES EST CRUCIAL AVANT L'ENTRAÎNEMENT POUR ASSURER L'INTÉGRITÉ DES DONNÉES.

02

POUR VISUALISER DES EXEMPLES D'IMAGES, UTILISEZ DES BIBLIOTHÈQUES COMME MATPLOTLIB. LA COMPRÉHENSION VISUELLE DES DONNÉES EST ESSENTIELLE POUR IDENTIFIER DES MOTIFS. IL EST IMPORTANT DE COMPRENDRE LES DONNÉES AVANT DE CONSTRUIRE UN MODÈLE.

03

APRÈS LE CHARGEMENT DES DONNÉES MNIST, DES ÉTAPES DE PRÉTRAITEMENT PEUVENT ÊTRE NÉCESSAIRES, COMME LA NORMALISATION DES VALEURS DE PIXEL ET LE REDIMENSIONNEMENT DES IMAGES. PRÉTRAITER LES DONNÉES EST CRUCIAL POUR AMÉLIORER LA STABILITÉ ET LA PERFORMANCE DU MODÈLE, ÉVITANT DES PROBLÈMES POTENTIELS TELS QUE LA CONVERGENCE LENTE.



Signature.
**MOHAMED ABI & HACINE
HOUSSAM**

EXTRACTION DES CARACTÉRISTIQUES AVEC COUNTVECTORIZER

01

NORMALISATION DES VALEURS DES PIXELS DANS LA PLAGE [0, 1]: LA NORMALISATION DES VALEURS DES PIXELS DANS LA PLAGE [0, 1] EST EFFECTUÉE POUR METTRE À L'ÉCHELLE LES DONNÉES. CELA A L'AVANTAGE DE RENDRE LES VALEURS COMPATIBLES AVEC LES FONCTIONS D'ACTIVATION COURAMMENT UTILISÉES (COMME LA FONCTION SIGMOÏDE) DANS LES MODÈLES DE MACHINE LEARNING. LA NORMALISATION FACILITE ÉGALEMENT LA CONVERGENCE DES ALGORITHMES D'OPTIMISATION EN RÉDUISANT LES VARIATIONS D'ÉCHELLE ENTRE LES CARACTÉRISTIQUES.

02

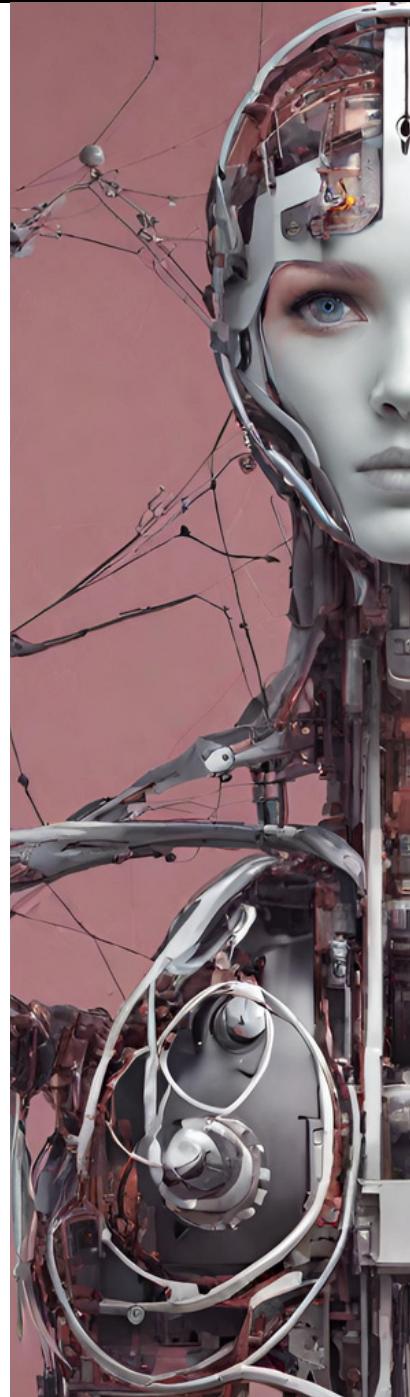
CONVERSION DES DONNÉES EN TYPE 'FLOAT32': LES DONNÉES SONT CONVERTIES EN TYPE 'FLOAT32' POUR UTILISER DES NOMBRES À VIRGULE FLOTTANTE AU LIEU D'ENTIERS. CELA PERMET D'EFFECTUER DES CALCULS PLUS PRÉCIS ET DE BÉNÉFICIER DES AVANTAGES DE LA VIRGULE FLOTTANTE POUR LES OPÉRATIONS NUMÉRIQUES, CE QUI EST PARTICULIÈREMENT IMPORTANT LORS DE L'ENTRAÎNEMENT DE MODÈLES COMPLEXES.

03

DIVISION DES VALEURS DES PIXELS PAR 255.0: LA DIVISION PAR 255.0 EST EFFECTUÉE POUR NORMALISER LES VALEURS DES PIXELS DANS LA PLAGE [0, 1]. LA VALEUR 255.0 EST CHOISIE CAR ELLE PRÉSENTE L'INTENSITÉ MAXIMALE DES PIXELS DANS UNE IMAGE EN NIVEAUX DE GRIS. EN DIVISANT PAR CETTE VALEUR, ON RAMÈNE LES INTENSITÉS DE PIXEL À UNE ÉCHELLE NORMALISÉE.

04

IMPACT SUR LA STABILITÉ ET LA CONVERGENCE: LA NORMALISATION DES PIXELS CONTRIBUE À LA STABILITÉ ET À LA CONVERGENCE DE L'ENTRAÎNEMENT EN ASSURANT QUE LES GRADIENTS CALCULÉS PENDANT LA RÉTROPROPAGATION NE DEVIENNENT PAS TROP GRANDS. DES VALEURS ÉLEVÉES PEUVENT ENTRAÎNER DES PROBLÈMES TELS QUE LA DIVERGENCE DE L'OPTIMISATION. LA NORMALISATION FAVORISE UNE CONVERGENCE PLUS RAPIDE ET STABLE EN MAINTENANT LES VALEURS DANS UNE PLAGE GÉRABLE.



DIVISION DES DONNÉES :

01

FORME ORIGINALE DE X_TEST AVANT LE REMODELAGE : LA FORME ORIGINALE DE X_TEST AVANT LE REMODELAGE DÉPEND DES DIMENSIONS SPÉCIFIQUES DU JEU DE DONNÉES, MAIS GÉNÉRALEMENT, ELLE EST DE LA FORME (NOMBRE D'IMAGES DE TEST, HAUTEUR, LARGEUR). LE REMODELAGE CHANGE LA STRUCTURE DES DONNÉES POUR LES PRÉPARER À ÊTRE UTILISÉES DANS UN MODÈLE.

02

SIGNIFICATION DU PARAMÈTRE "-1" DANS L'OPÉRATION DE RESHAPE : DANS L'OPÉRATION DE REMODELAGE, LE PARAMÈTRE "-1" EST UTILISÉ POUR INDICER À NUMPY (OU UNE AUTRE BIBLIOTHÈQUE SIMILAIRE) DE CALCULER AUTOMATIQUEMENT LA TAILLE DE CETTE DIMENSION. CELA FACILITE LE REMODELAGE EN FONCTION DES AUTRES DIMENSIONS SPÉCIFIÉES, SANS AVOIR À PRÉCISER MANUELLEMENT LA TAILLE.

03

FORME RÉSULTANTE DE X_TEST_FLAT APRÈS LE REMODELAGE : APRÈS L'OPÉRATION DE REMODELAGE, X_TEST_FLAT EST GÉNÉRALEMENT UNE MATRICE À DEUX DIMENSIONS OÙ UNE DIMENSION CORRESPOND AU NOMBRE D'IMAGES DE TEST ET L'AUTRE À LA CONCATÉNATION DES PIXELS DE CHAQUE IMAGE. CELA PEUT ÊTRE IMPORTANT POUR LE MODÈLE CAR IL S'ATTEND À DES ENTRÉES SOUS FORME DE VECTEURS UNIDIMENSIONNELS PLUTÔT QUE DE MATRICES BIDIMENSIONNELLES.

04

CORRESPONDANCE DES DIMENSIONS AVEC LES EXIGENCES DU MODÈLE OU DE LA CONCEPTION DU RÉSEAU NEURONAL : LES DIMENSIONS DES DONNÉES REMODELÉES (X_TEST_FLAT) SONT GÉNÉRALEMENT CONFORMES AUX EXIGENCES DU MODÈLE, SURTOUT SI LE MODÈLE UTILISE UNE COUCHE D'ENTRÉE DENSE (FULLY CONNECTED) OÙ CHAQUE PIXEL EST TRAITÉ COMME UNE CARACTÉRISTIQUE DISTINCTE. LES MODÈLES DE RÉSEAUX NEURONNAUX ATTENDENT SOUVENT DES ENTRÉES SOUS FORME DE VECTEURS UNIDIMENSIONNELS, D'OÙ LA NÉCESSITÉ DE REMODELER LES DONNÉES DE MANIÈRE APPROPRIÉE POUR S'ALIGNER AVEC CETTE STRUCTURE ATTENDUE.



Signature
MACHINE LEARNING

MOHAMED ABI & HOUSSAM
HACINCE

PAGE 06

CRÉATION ET ENTRAÎNEMENT DU MODÈLE : 1. RÉSEAUX DE NEURONS :

01

ARCHITECTURE GÉNÉRALE DU MODÈLE :

- LE MODÈLE COMMENCE PAR UNE COUCHE D'ENTRÉE FLATTEN, QUI TRANSFORME L'ENTRÉE BIDIMENSIONNELLE EN UN VECTEUR UNIDIMENSIONNEL.
- SUIVI D'UNE COUCHE DENSE AVEC 128 NEURONES, UTILISANT LA FONCTION D'ACTIVATION 'RELU'.
- ENFIN, UNE COUCHE DENSE AVEC 10 NEURONES ET LA FONCTION D'ACTIVATION 'SOFTMAX' POUR LA CLASSIFICATION MULTI-CLASSES.

02

FORME DE L'ENTRÉE DU MODÈLE (INPUT_SHAPE) :

- L'ENTRÉE DU MODÈLE EST DÉFINIE AVEC `INPUT_SHAPE=(28, 28)`, CE QUI CORRESPOND À LA FORME DES IMAGES MNIST APRÈS LE PRÉTRAITEMENT. CETTE CONFIGURATION EST LIÉE À LA PREMIÈRE COUCHE FLATTEN, QUI TRANSFORME CHAQUE IMAGE EN UN VECTEUR DE TAILLE $28 \times 28 = 784$.

03

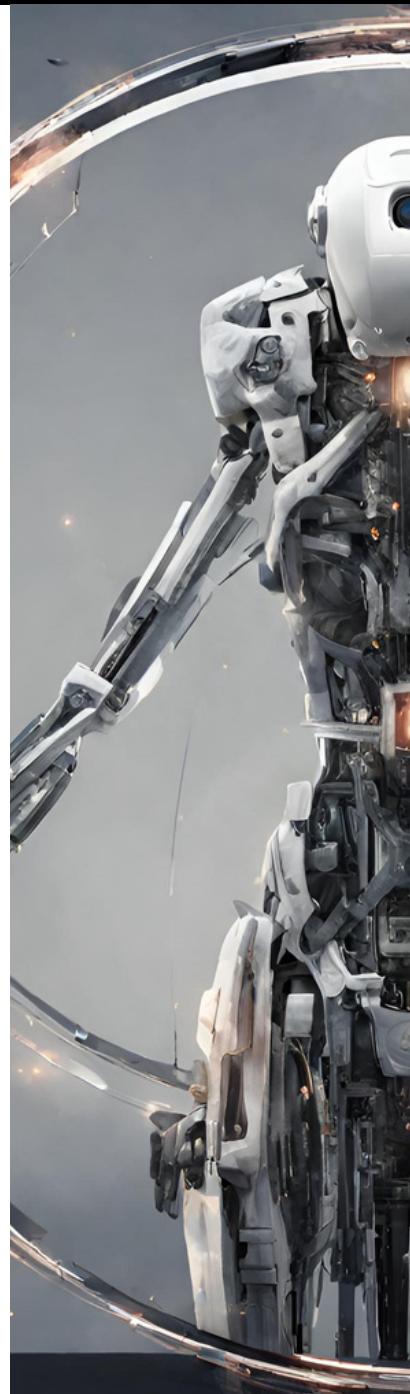
CHOIX DE LA COUCHE DENSE AVEC 128 NEURONES :

- LE CHOIX DE 128 NEURONES DANS LA COUCHE DENSE EST UNE DÉCISION DE CONCEPTION. CELA PEUT ÊTRE BASÉ SUR DES CONSIDÉRATIONS EMPIRIQUES OU DES EXPÉRIENCES ANTÉRIEURES. C'EST UNE TAILLE SOUVENT UTILISÉE POUR CRÉER UNE REPRÉSENTATION INTERMÉDIAIRE COMPLEXE DES DONNÉES.

04

RÔLE DE LA DERNIÈRE COUCHE DENSE AVEC 'SOFTMAX' :

- LA DERNIÈRE COUCHE DENSE AVEC 10 NEURONES ET LA FONCTION D'ACTIVATION 'SOFTMAX' EST UTILISÉE POUR EFFECTUER LA CLASSIFICATION MULTI-CLASSES. 'SOFTMAX' ATTRIBUE DES PROBABILITÉS AUX DIFFÉRENTES CLASSES, PERMETTANT D'IDENTIFIER LA CLASSE LA PLUS PROBABLE POUR CHAQUE IMAGE.



CRÉATION ET ENTRAÎNEMENT DU MODÈLE : 1. RÉSEAUX DE NEURONS :

05

COMPARAISON DES PERFORMANCES : LA COMPARAISON DES PERFORMANCES AVEC LE CLASSIFIEUR NAÏVE BAYES MULTINOMIAL DÉPEND DES CARACTÉRISTIQUES DU JEU DE DONNÉES. LES MODÈLES DE RÉSEAU DE NEURONES PEUVENT CAPTURER DES RELATIONS COMPLEXES, MAIS ILS NÉCESSITENT GÉNÉRALEMENT PLUS DE DONNÉES ET DE TEMPS DE CALCUL.

06

CHOIX DE L'ARCHITECTURE ADAPTÉE À LA TÂCHE :

- LE CHOIX DE L'ARCHITECTURE DÉPEND DE LA COMPLEXITÉ DU PROBLÈME. POUR LA CLASSIFICATION D'IMAGES MNIST, UNE ARCHITECTURE RELATIVEMENT SIMPLE PEUT SUFFIRE, MAIS DES TÂCHES PLUS COMPLEXES NÉCESSITERAIENT DES ARCHITECTURES PLUS PROFONDES ET COMPLEXES.

07

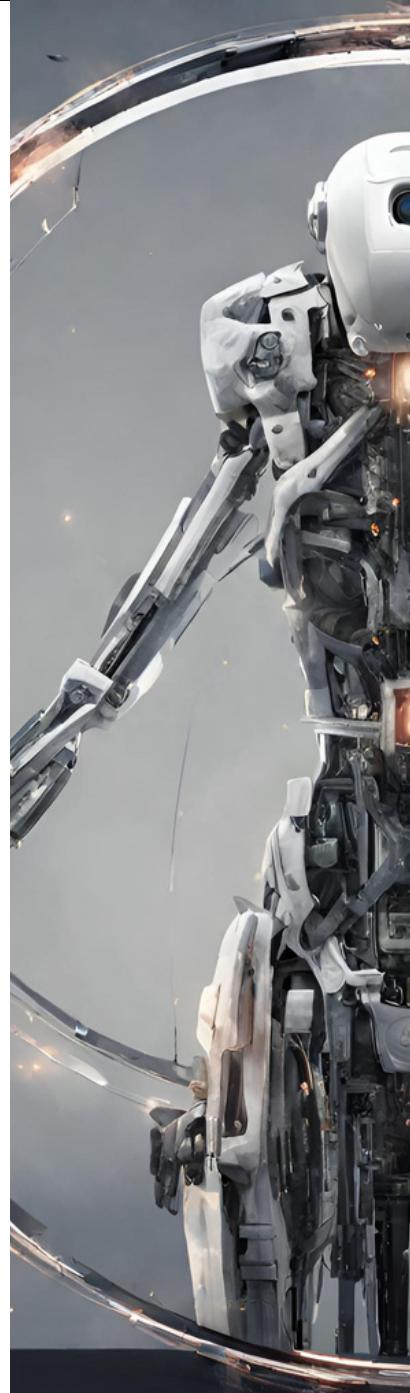
NOMBRE D'ÉPOQUES :

- ENTRAÎNER LE MODÈLE PENDANT 5 ÉPOQUES EST UN CHOIX EMPIRIQUE. LE NOMBRE OPTIMAL D'ÉPOQUES DÉPEND DE LA CONVERGENCE DU MODÈLE SUR L'ENSEMBLE D'ENTRAÎNEMENT. CELA PEUT NÉCESSITER DES ESSAIS ET DES AJUSTEMENTS.

08

SIGNIFICATION DU PARAMÈTRE 'BATCH_SIZE' :

- 'BATCH_SIZE' DÉTERMINE LE NOMBRE D'ÉCHANTILLONS UTILISÉS POUR METTRE À JOUR LES POIDS DU MODÈLE LORS DE L'ENTRAÎNEMENT. UNE TAILLE DE LOT DE 32 EST CHOISIE ICI POUR DES RAISONS DE COMPROMIS ENTRE LA STABILITÉ DE L'ENTRAÎNEMENT ET LA VITESSE DE CALCUL.



Signature .

**MOHAMED ABI &
HOUSSAM HACINE**

MACHINE LEARNING

PAGE 07

■ CRÉATION ET ENTRAÎNEMENT DU MODÈLE : 1. RÉSEAUX DE NEURONS :

09

UTILISATION DU PARAMÈTRE 'VALIDATION_SPLIT' :

- 'VALIDATION_SPLIT' EST UTILISÉ POUR DIVISER LES DONNÉES D'ENTRAÎNEMENT EN ENSEMBLES D'ENTRAÎNEMENT ET DE VALIDATION. CELA PERMET D'ÉVALUER LA PERFORMANCE DU MODÈLE SUR DES DONNÉES NON UTILISÉES PENDANT L'ENTRAÎNEMENT ET DE DÉTECTOR LE SURAJUSTEMENT.

10

SAUVEGARDE DU MODÈLE ENTRAÎNÉ :

- VOUS POUVEZ SAUVEGARDER LE MODÈLE ENTRAÎNÉ EN UTILISANT MODEL.SAVE("NOM_DU_FICHIER.H5") OU FORMAT **JOBLIB**. CELA PERMET DE RECHARGER LE MODÈLE POUR UNE UTILISATION FUTURE SANS AVOIR À LE RÉENTRAÎNER.

11

NOMBRE DE CLASSES :

- LE NOMBRE DE CLASSES EST DÉFINI EN FONCTION DU PROBLÈME. POUR MNIST, IL Y A 10 CLASSES CORRESPONDANT AUX CHIFFRES DE 0 À 9.

12

CONFIGURATION DES HYPER-PARAMÈTRES :

- LES HYPER-PARAMÈTRES TELS QUE LE NOMBRE DE COUCHES, LA TAILLE DES FILTRES, ETC., DOIVENT ÊTRE CONFIGURÉS EN FONCTION DE L'EXPÉRIENCE ET DES CARACTÉRISTIQUES DU PROBLÈME. CELA PEUT NÉCESSITER DES ESSAIS ET DES AJUSTEMENTS POUR TROUVER LA MEILLEURE CONFIGURATION.



Signature .
**MOHAMED ABI & HOUSSAM
HACINE**



KNN :

2-Adaptation du modèle KNN aux données d'entraînement :

- Le modèle KNN (k-nearest neighbors) s'adapte aux données d'entraînement en mémorisant simplement les exemples du jeu de données. Lors de la phase de prédiction, il classe un nouvel exemple en fonction de la classe majoritaire parmi ses k voisins les plus proches dans l'espace des caractéristiques

2-Aplatissement des données pour KNN:

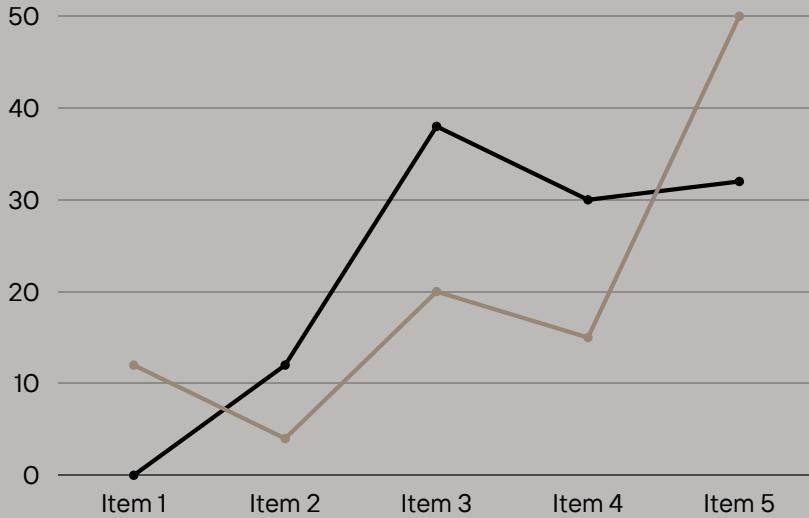
- Les données d'entraînement sont parfois aplatis avant d'être fournies au modèle KNN pour convertir les images bidimensionnelles en vecteurs unidimensionnels. Cela peut affecter la performance en modifiant la manière dont la similarité entre les exemples est calculée. L'impact dépend du problème spécifique et de la manière dont les caractéristiques sont distribuées.

3- Influence du choix de ' k ' sur la précision :

- Le choix de ' k ', le nombre de voisins considérés, peut influencer la précision du modèle KNN. Un ' k ' trop petit peut rendre le modèle sensible au bruit, tandis qu'un ' k ' trop grand peut introduire une généralisation excessive. Il existe un compromis entre la précision et la complexité du modèle, où un ' k ' optimal doit être sélectionné.

4-Détermination empirique du meilleur ' k ' :

- Pour déterminer empiriquement le meilleur ' k ', vous pouvez utiliser la validation croisée ou diviser le jeu de données d'entraînement en ensembles d'entraînement et de validation. En testant différents valeurs de ' k ', vous pouvez choisir celle qui donne la meilleure performance sur l'ensemble de validation. Il est important de surveiller la précision sur l'ensemble de test pour éviter le surajustement.



PRÉDICTIONS ET ÉVALUATION :

1-Utilisation du modèle pour faire des prédictions :

- Le modèle entraîné est utilisé pour faire des prédictions sur l'ensemble de test (X_{test}). Chaque image est classée dans une catégorie prédite en utilisant le modèle.

2-Évaluation de la précision :

- La précision du modèle est évaluée en comparant les prédictions générées par le modèle avec les étiquettes réelles de l'ensemble de test. Cela donne une mesure de la capacité du modèle à correctement classer les exemples de test.

3-Matrice de confusion :

- La matrice de confusion est une représentation tabulaire des résultats de classification. Elle montre le nombre d'occurrences pour chaque combinaison de classe réelle et prédite, ce qui permet d'évaluer les performances du modèle en termes de vrais positifs, vrais négatifs, faux positifs et faux négatifs.

Affichage des Résultats :

```
# Affichage des résultats
print(f"Neural Network Accuracy: {nn_accuracy}")
print(f"K-NN Accuracy: {knn_accuracy}")
print(f"Hybrid Model Accuracy: {hybrid_accuracy}")
```

Neural Network Accuracy: 0.979
 K-NN Accuracy: 0.9688
 Hybrid Model Accuracy: 0.0984



IMPACTE D'HYBRIDATION DES MODÈLES DES DONNÉES SUR LA CLASSIFICATION DES IMAGES

1-Utilisation de np.vstack pour empiler verticalement les prédictions :

- On utilise np.vstack pour empiler verticalement les prédictions des deux modèles afin de créer une matrice où chaque ligne correspond à la prédiction d'un modèle différent. Cela facilite la combinaison en permettant de prendre les votes majoritaires par ligne.

2_Fonctionnement du vote majoritaire :

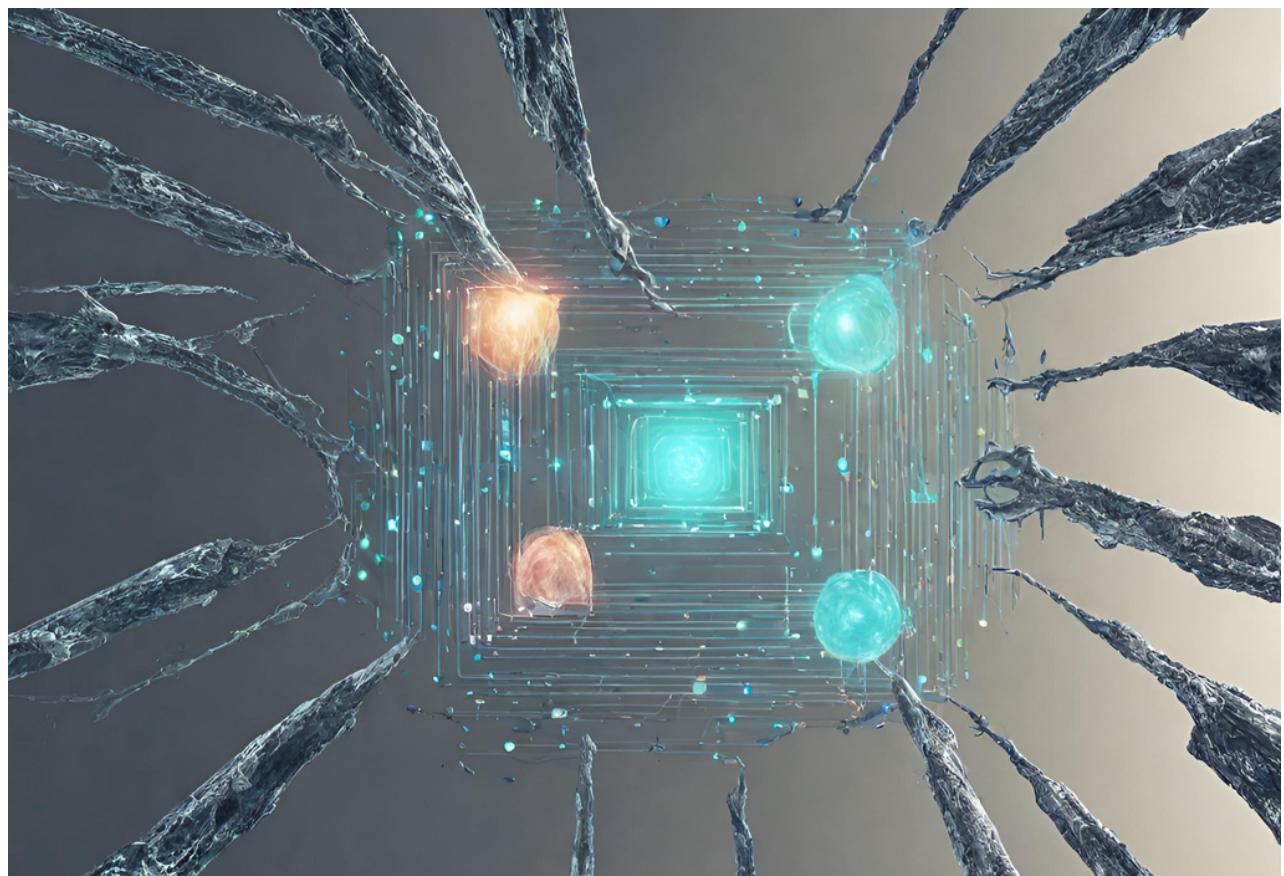
- Dans le contexte de ce vote majoritaire, les prédictions des deux modèles pour chaque exemple sont prises en compte. La classe finale prédite est celle qui obtient le plus grand nombre de votes parmi les deux modèles. C'est la prédiction finale qui sera considérée pour cet exemple.

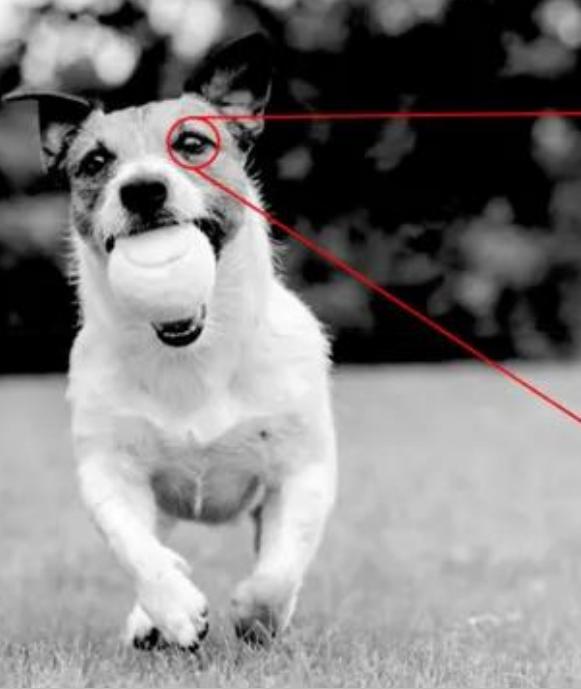
3-Différences potentielles entre les prédictions des modèles (NN et KNN) :

- Les prédictions des deux modèles peuvent différer en raison de leurs approches distinctes pour la classification. Par exemple, le modèle neuronal (NN) peut capturer des relations complexes, tandis que le modèle KNN peut être plus sensible aux caractéristiques locales. La combinaison cherche à exploiter ces différences pour améliorer la robustesse et la performance globale.

4-Gestion des prédictions égales dans la combinaison par vote majoritaire :

- Si les prédictions des deux modèles sont égales pour un exemple donné, cela peut se produire en cas d'accord total ou de désaccord total entre les modèles. Dans la combinaison par vote majoritaire, le choix de la classe finale est généralement arbitraire dans le cas d'une égalité. Cependant, cela peut être géré en utilisant des mécanismes de prise de décision spécifiques, tels que la sélection de la classe avec l'indice le plus bas en cas d'égalité.





```
[208, 126, 83],  
[207, 124, 82],  
[210, 126, 84]],  
  
[[112, 143, 166],  
[127, 158, 181],  
[145, 176, 199],
```

L'HYBRIDATION DE MODÈLES : RÉSEAU DE NEURONES, K-NN ET SVM

```
from sklearn.svm import SVC  
  
# Ajustement des paramètres du SVM  
svm_model = SVC(kernel='linear', C=1.0)  
svm_model.fit(x_train_flat, y_train)  
svm_predictions = svm_model.predict(x_test_flat)
```

```
hybrid_predictions_svm = np.argmax(np.vstack((nn_predictions, svm_predictions)), axis=1)  
  
# Évaluation des performances du modèle hybride avec SVM  
hybrid_accuracy_svm = accuracy_score(y_test, hybrid_predictions_svm)  
print(f"Hybrid Model (with SVM) Accuracy: {hybrid_accuracy_svm}")
```

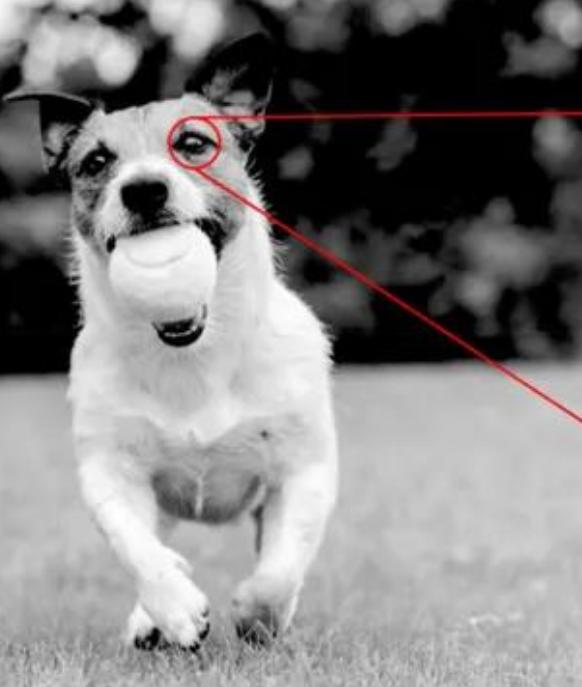
Accuracy With Hibryd Model (SVM + NN)

```
hybrid_predictions_svm = np.argmax(np.vstack((nn_predictions, svm_predictions)), axis=1)  
  
# Évaluation des performances du modèle hybride avec SVM  
hybrid_accuracy_svm = accuracy_score(y_test, hybrid_predictions_svm)  
print(f"Hybrid Model (with SVM) Accuracy: {hybrid_accuracy_svm}")
```

```
Hybrid Model (with SVM) Accuracy: 0.0981
```

+ Code

+ Markdown



```
[208, 126, 83],  
[207, 124, 82],  
[210, 126, 84]],  
  
[[112, 143, 166],  
[127, 158, 181],  
[145, 176, 199],
```

FINALY THIS IS THE COMAPRISION BETWEEN THE TWO HYBRYD MODELS

Accurency With Hibryd Model (SVM + NN)

```
hybrid_predictions_svm = np.argmax(np.vstack((nn_predictions, svm_predictions))

# Évaluation des performances du modèle hybride avec SVM
hybrid_accuracy_svm = accuracy_score(y_test, hybrid_predictions_svm)
print(f"Hybrid Model (with SVM) Accuracy: {hybrid_accuracy_svm}")


```

Hybrid Model (with SVM) Accuracy: 0.0981

Accurency With Hibryd Model (KNN + NN)

```
# Affichage des résultats
print(f"Neural Network Accuracy: {nn_accuracy}")
print(f"K-NN Accuracy: {knn_accuracy}")
print(f"Hybrid Model Accuracy: {hybrid_accuracy}")
```

Neural Network Accuracy: 0.9779
K-NN Accuracy: 0.9688
Hybrid Model Accuracy: 0.0984

+ Code

+ Markdown