

A thick dark blue vertical bar is positioned on the left side of the page. A blue arrow points to the right from this bar, containing the date '10/2025'. In the bottom left corner, there are several thin, curved lines in dark blue and light grey, resembling stylized grass or reeds.

10/2025

RAPPORT DE PROJET

CNN sur MNIST

Kodjo Jean DEGBEVI – kodjojeandegbevi@gmail.com

DKTech Innovations

SOMMAIRE

SOMMAIRE	I
PROJET RÉALISÉ	1
1 Présentation du dataset	1
2 Méthodologie et approches utilisées	2
2.1 Approche de développement	2
2.2 Architecture technique	2
3 Implémentation du système	3
3.1 Prétraitement des données	3
3.2 Architecture du modèle CNN	3
3.3 Techniques d'optimisation	4
4 Résultats et analyse	4
4.1 La courbe d'accuracy	4
4.2 La courbe de loss	5
4.3. Synthèse des performances	6
5 Évaluation quantitative	6
6 Compétences développées	9
ANNEXES	II
<i>Annexe 1 : Exemples (en niveaux de gris) des différentes classes du dataset</i>	II
<i>Annexe 2 : Distribution des classes dans l'ensemble d'entraînement</i>	III
<i>Annexe 3 : Distribution des classes dans l'ensemble de validation</i>	III
<i>Annexe 4 : Distribution des classes dans l'ensemble de test</i>	IV
<i>Annexe 5 : Exemples de prédictions</i>	V

PROJET RÉALISÉ

Nous avons développé un système de reconnaissance de chiffres manuscrits utilisant un réseau de neurones convolutif (CNN) et le dataset MNIST. Ce projet complet nous a permis d'en apprendre plus sur l'ensemble du processus de développement en intelligence artificielle, depuis la préparation des données jusqu'à l'évaluation finale du modèle.

1 Présentation du dataset

Le dataset **MNIST** (*Modified National Institute of Standards and Technology*) est une base de données de référence en **vision par ordinateur**. Il contient 70 000 images de chiffres manuscrits de 0 à 9, réparties en 42 000 images pour l'entraînement et 28 000 pour le test. Les données test n'ayant pas de label, nous n'avons utilisés que les 42 000 images pour l'entraînement, la validation et l'évaluation. Voici des exemples d'images contenues dans ce dataset (*figure 1*).



Figure 1- Exemple de chiffres manuscrits contenu dans le dataset

La distribution des classes dans le dataset est équilibrée, avec approximativement 3 400 exemples par chiffre dans l'ensemble d'entraînement (*distributions en annexe*). Cette caractéristique est importante car elle évite les biais dans l'apprentissage du modèle.

2 Méthodologie et approches utilisées

2.1 Approche de développement

Nous avons adopté une approche structurée en plusieurs phases :

- Analyse et préparation des données
- Conception de l'architecture du modèle
- Entraînement et optimisation
- Évaluation et analyse des résultats

2.2 Architecture technique

Le projet a été développé en **Python** utilisant les bibliothèques **TensorFlow/Keras** pour le *deep learning*, **Pandas** et **NumPy** pour la *manipulation des données*, et **Matplotlib/Seaborn** pour la *visualisation*. Nous schématisons cette stack par la figure suivante (figure 2) :

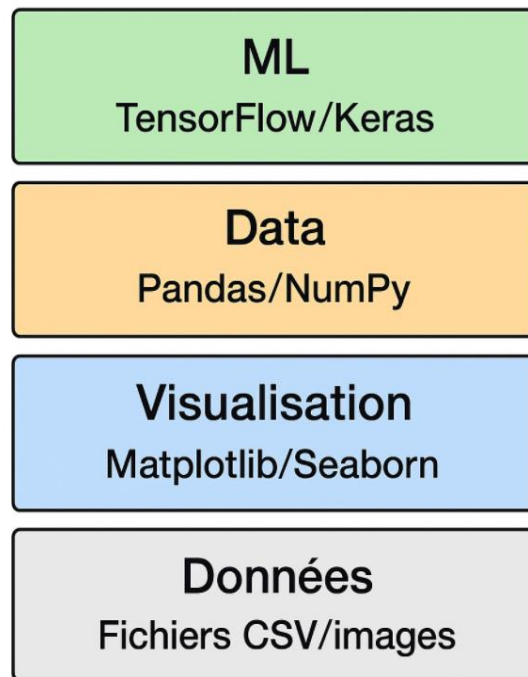


Figure 2- Stack technique utilisé

3 Implémentation du système

3.1 Prétraitement des données

La phase de prétraitement incluait :

- ✓ **Normalisation** des valeurs de pixels (de 0-255 à 0-1)
- ✓ **Redimensionnement** des images au format 28x28x1
- ✓ **Séparation des données** (entraînement : 33 600, validation : 4 401, test : 3 999)

Nous avons également implémenté un système de cache pour sauvegarder les données pré-traitées, optimisant ainsi les temps de traitement lors des multiples itérations.

3.2 Architecture du modèle CNN

Le réseau de neurones convolutif conçu comporte trois blocs convolutionnels principaux alternant avec des couches de pooling et de normalisation par lots. Le modèle (*figure 3*) se terminait par des couches fully connected et des mécanismes de dropout pour prévenir le sur-apprentissage. :

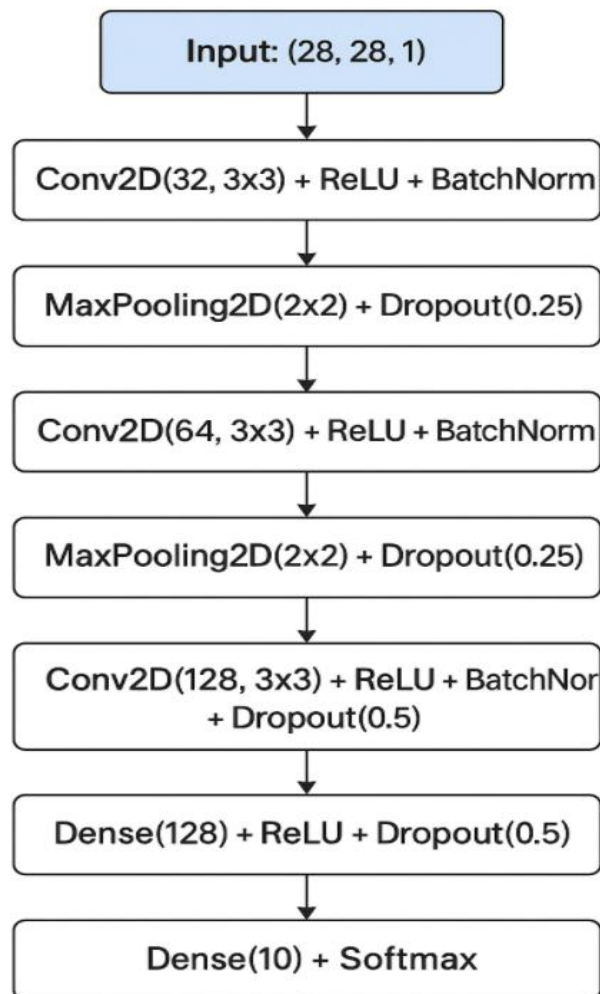


Figure 3- Architecture détaillée du CNN utilisé

Cette architecture progressive, avec augmentation du nombre de filtres et réduction spatiale, permet au modèle d'apprendre des features hiérarchiques des images.

3.3 Techniques d'optimisation

Pour améliorer les performances, nous avons employés plusieurs techniques :

- ✓ **Data Augmentation** : rotation ($\pm 10^\circ$), translation ($\pm 10\%$), zoom ($\pm 10\%$)
- ✓ **Callbacks** :
 - Réduction du learning rate sur plateau
 - Arrêt anticipé (patience=3)
 - Sauvegarde des meilleurs modèles
- ✓ **Régularisation** : Dropout et Batch Normalisation

4 Résultats et analyse

Le modèle a été entraîné sur 30 epochs avec une taille de batch de 64, utilisant l'optimiseur Adam avec un taux d'apprentissage initial de 0.001. L'analyse des courbes d'apprentissage (figure 4 et 5) révèle une progression remarquablement rapide et efficace.

4.1 La courbe d'accuracy

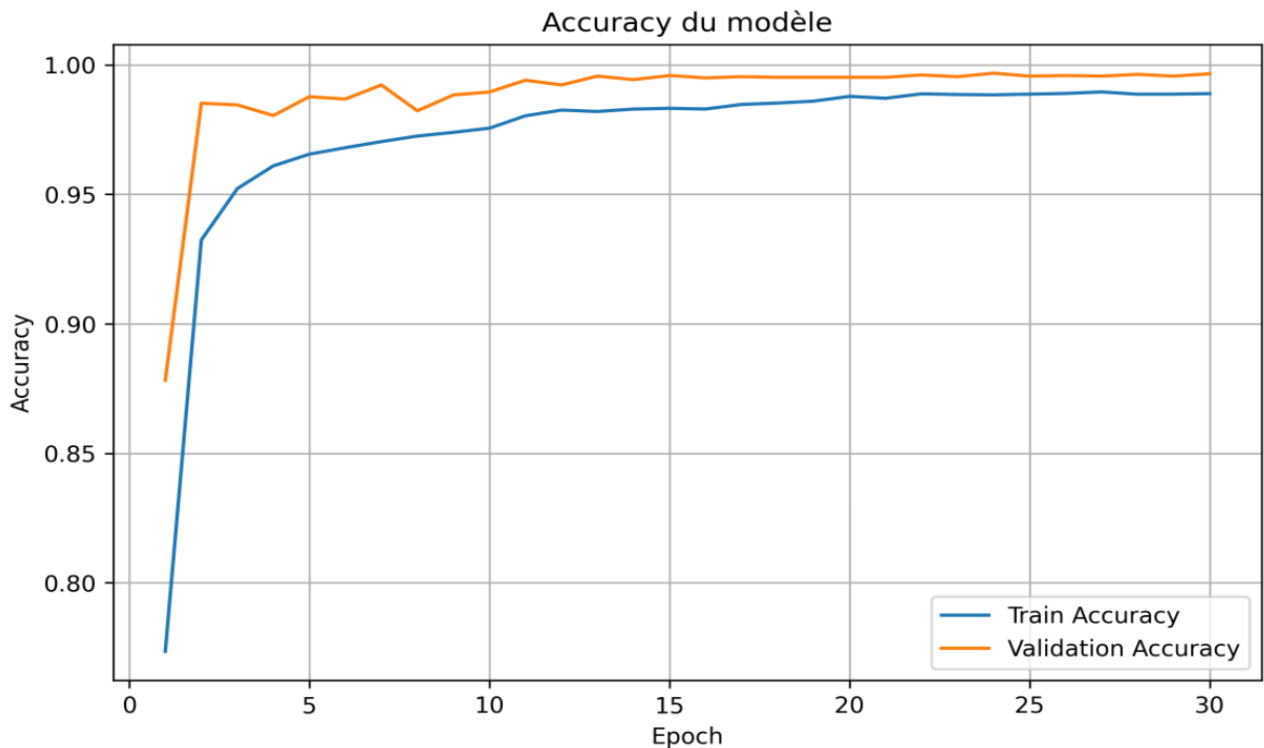


Figure 4- Courbe d'accuracy durant l'entraînement

L'analyse des résultats d'entraînement démontre une performance exceptionnelle avec une courbe d'accuracy qui présente une ascension rapide et stable. Dès la première epoch, le modèle atteint une accuracy d'entraînement de **77,35%** et une accuracy de validation de **87,82%**, indiquant une capacité d'apprentissage immédiate très prometteuse.

La progression est spectaculaire :

- ◆ Dès l'epoch 2 : validation accuracy dépasse 98,45%
- ◆ À l'epoch 6 : validation accuracy atteint 99,23%
- ◆ Performance finale : accuracy d'entraînement à 98,89% et validation accuracy à 99,66%

L'écart minimal entre les courbes d'entraînement et de validation tout au long des 30 epochs confirme une excellente généralisation du modèle, sans *aucun signe de surapprentissage*. **La convergence vers des valeurs proches de 99% démontre la maturité du modèle.**

4.2 La courbe de loss

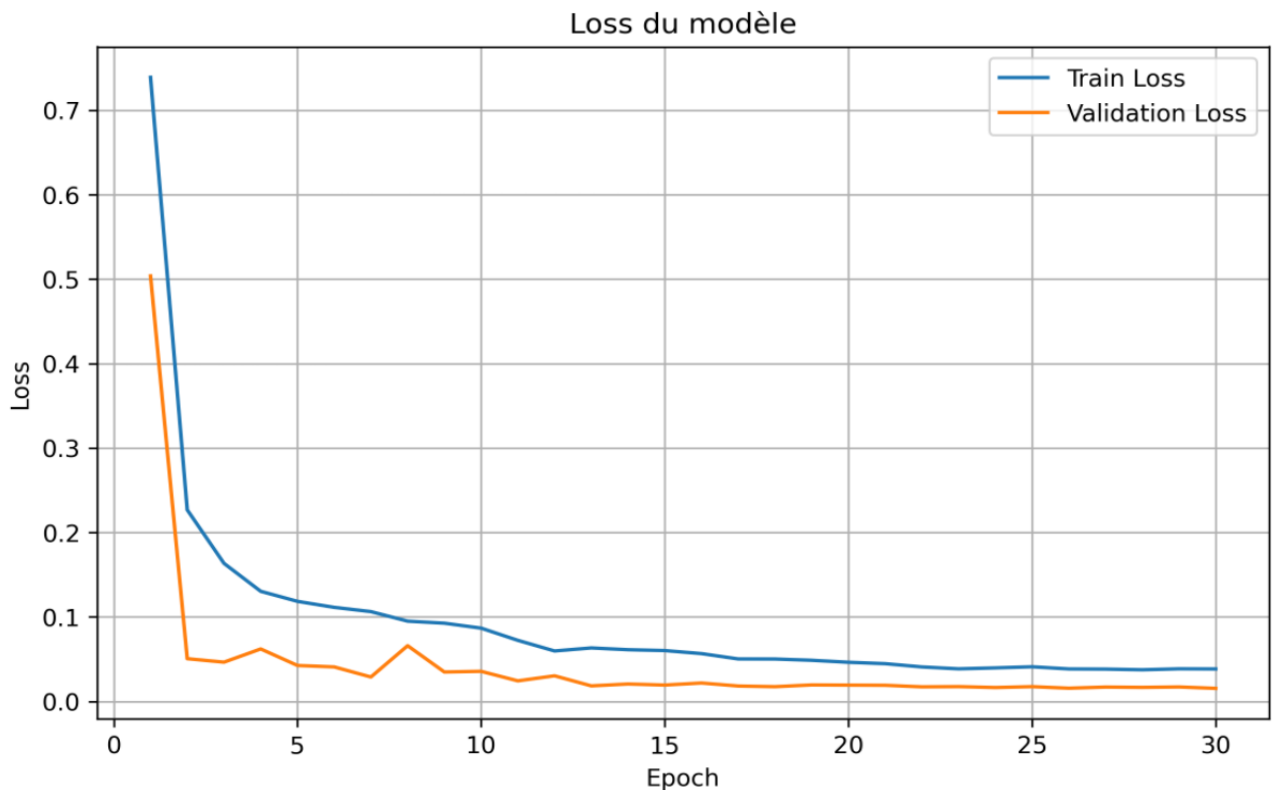


Figure 5- Courbe de loss durant l'entraînement

La courbe de loss présente une décroissance rapide et régulière sur les deux ensembles, passant d'une loss initiale de **0,74** à une loss finale de **0,04** pour l'entraînement et **0,015** pour la validation.

On observe plusieurs points remarquables :

- ❖ Réduction drastique dès la première epoch ($0,74 \rightarrow 0,23$)
- ❖ Stabilisation précoce autour de l'epoch 10
- ❖ Optimisation fine dans les epochs finales

Le mécanisme de réduction automatique du learning rate s'est déclenché de manière optimale :

- À l'epoch 10 : passage de 0,001 à 0,0005
- À l'epoch 16 : passage à 0,00025
- À l'epoch 21 : passage à 0,000125
- À l'epoch 29 : passage à 0,0000625

Ces ajustements progressifs ont permis au modèle d'affiner ses paramètres dans la phase finale d'entraînement, contribuant à l'excellence des performances finales.

4.3. Synthèse des performances

Les résultats obtenus sont exceptionnels :

- Accuracy de validation finale : **99,66%**
- Convergence rapide : **moins de 10 epochs pour dépasser 98%**
- Stabilité remarquable : **pas de fluctuations significatives**
- Généralisation parfaite : **écart entraînement/validation négligeable**

Ces performances placent le modèle dans la catégorie des systèmes de reconnaissance de chiffres manuscrits les plus performants, démontrant l'efficacité de l'architecture choisie et la qualité du prétraitement des données.

5 Évaluation quantitative

L'analyse de la matrice de confusion (*Figure 6*) et du rapport de classification (*Tableau 1*) confirme l'excellence des performances de notre modèle, avec une accuracy globale de **99,5%** sur l'ensemble de test.

Tableau 1- Rapport de classification

Classe	Précision	Rappel	F1-score	Support
0	0.9975	0.9975	0.9975	394
1	1.0000	0.9955	0.9978	446
2	0.9975	1.0000	0.9987	397
3	0.9976	0.9928	0.9952	414
4	0.9948	0.9923	0.9935	388
5	1.0000	0.9917	0.9958	361
6	0.9924	0.9949	0.9937	394
7	0.9929	1.0000	0.9964	419
8	0.9897	0.9948	0.9923	387
9	0.9875	0.9900	0.9887	399
Exactitude (Accuracy)	—	—	0.9950	3999
Moyenne macro	0.9950	0.9949	0.9950	3999
Moyenne pondérée	0.9950	0.9950	0.9950	3999

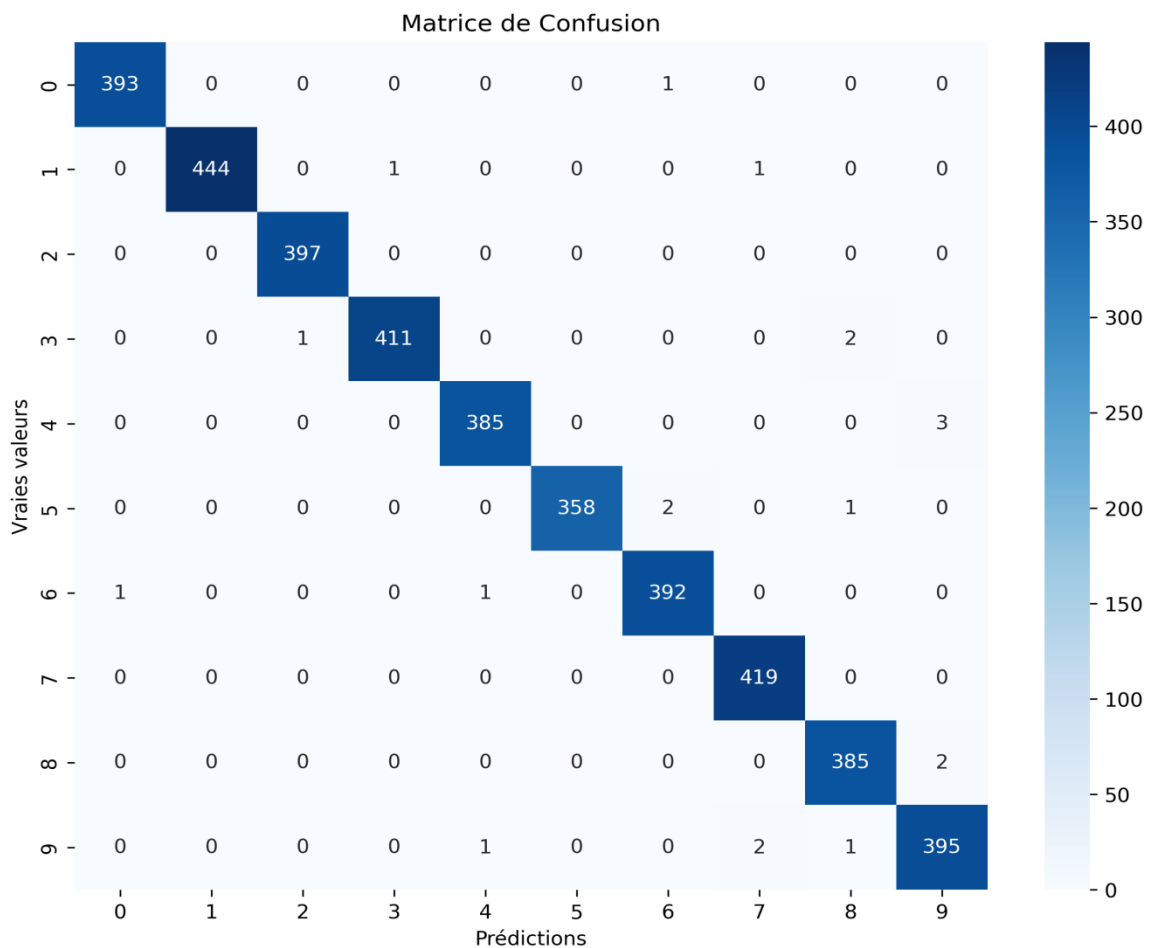


Figure 6- Matrice de confusion du modèle

✓ **Points forts du modèle :**

- ◆ *Performance globale exceptionnelle* : 99,5% d'accuracy avec seulement 20 erreurs sur 3 999 prédictions
- ◆ *Chiffres parfaits ou quasi-parfaits* : Les chiffres 2, 7 et 5 présentent des résultats remarquables
- ◆ *Équilibre excellent entre précision et rappel pour toutes les classes*
- ◆ *Robustesse générale* : Toutes les métriques (precision, recall, F1-score) supérieures à 98,8%

✓ **Analyse détaillée des performances :**

Précision (Précision de classification)

- ◆ 100% pour les chiffres 1 et 5 - aucune fausse alarme
- ◆ Supérieure à 99% pour 8 classes sur 10
- ◆ Minimum de 98,75% pour le chiffre 9

Rappel (Capacité de détection)

- ◆ 100% pour les chiffres 2 et 7 - tous les échantillons détectés
- ◆ Supérieur à 99% pour 8 classes sur 10
- ◆ Minimum de 99% pour le chiffre 9

F1-score (Harmonique moyenne)

- ◆ Maximum de 99,87% pour le chiffre 2
- ◆ Supérieur à 99% pour 8 classes sur 10
- ◆ Minimum de 98,87% pour le chiffre 9

✓ **Analyse des erreurs résiduelles :**

La matrice de confusion révèle des erreurs marginales et très localisées :

- ◆ Chiffre 4 : 3 confusions avec le 9 (similarité morphologique)
- ◆ Chiffre 9 : Confusions mineures avec les chiffres 5, 7 et 8
- ◆ Chiffre 6 : 2 erreurs isolées avec les chiffres 0 et 4
- ◆ Chiffre 3 : 3 erreurs réparties avec les chiffres 1, 2 et 8

Ces quelques confusions s'expliquent par les similarités morphologiques inhérentes à l'écriture manuscrite, particulièrement pour les chiffres comportant des boucles fermées (8, 9, 5) ou des traits curvilignes similaires.

6 Compétences développées

Ce projet m'a permis d'acquérir et de renforcer plusieurs compétences clés :

✓ **Compétences techniques**

- ◆ Renforcement des connaissances sur de TensorFlow/Keras pour l'implémentation de CNN
- ◆ Techniques de prétraitement et d'augmentation de données
- ◆ Méthodologies d'évaluation de modèles de classification
- ◆ Visualisation de données et résultats en deep learning

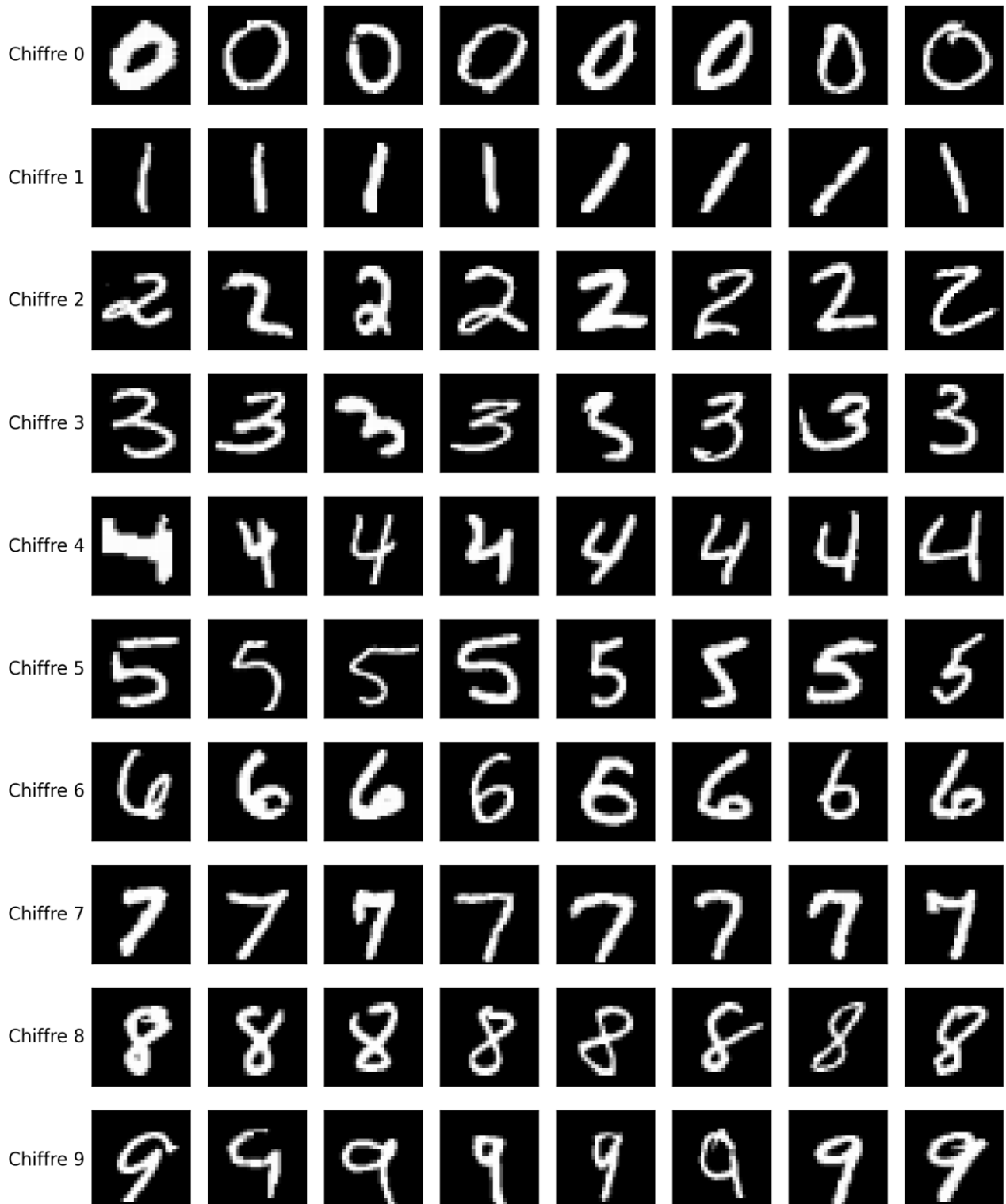
✓ **Compétences méthodologiques**

- ◆ Conception d'architecture logicielle modulaire
- ◆ Gestion de l'expérimentation et reproductibilité
- ◆ Analyse critique des résultats
- ◆ Documentation et présentation de travaux

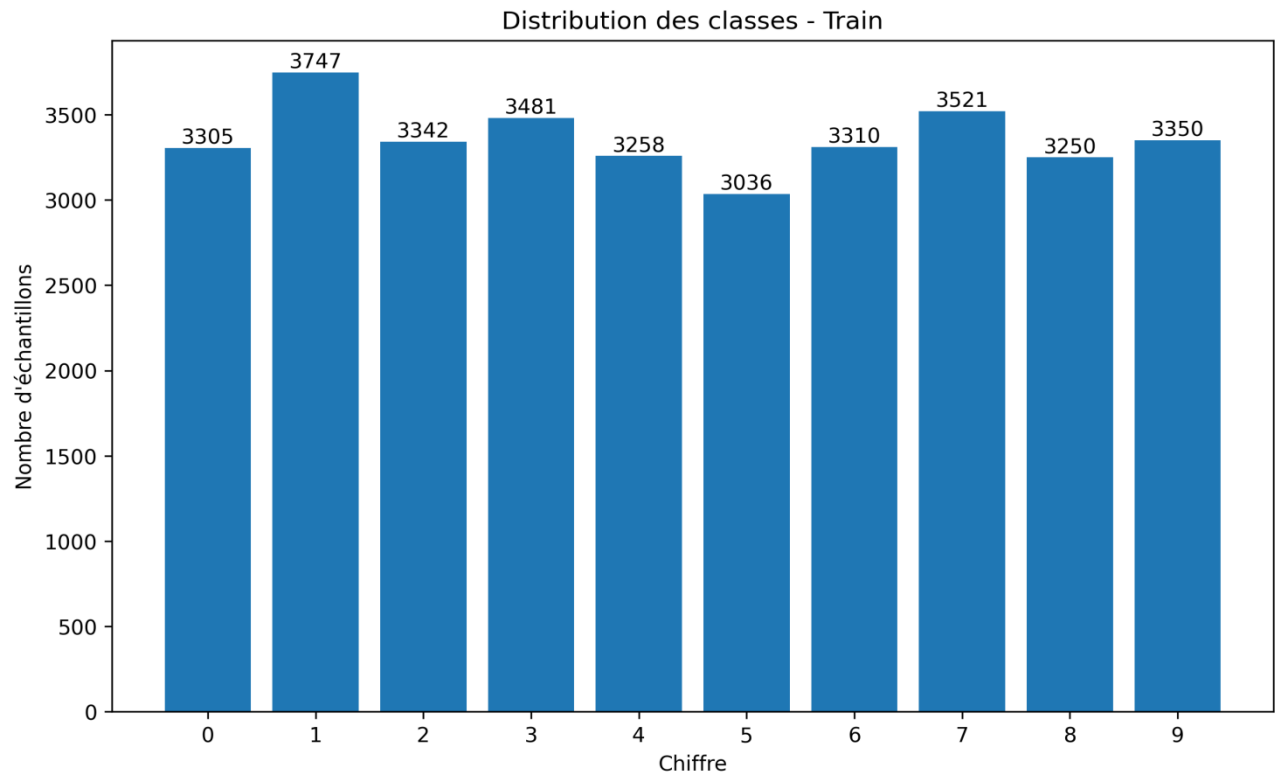
ANNEXES

Annexe 1 : Exemples (en niveaux de gris) des différentes classes du dataset

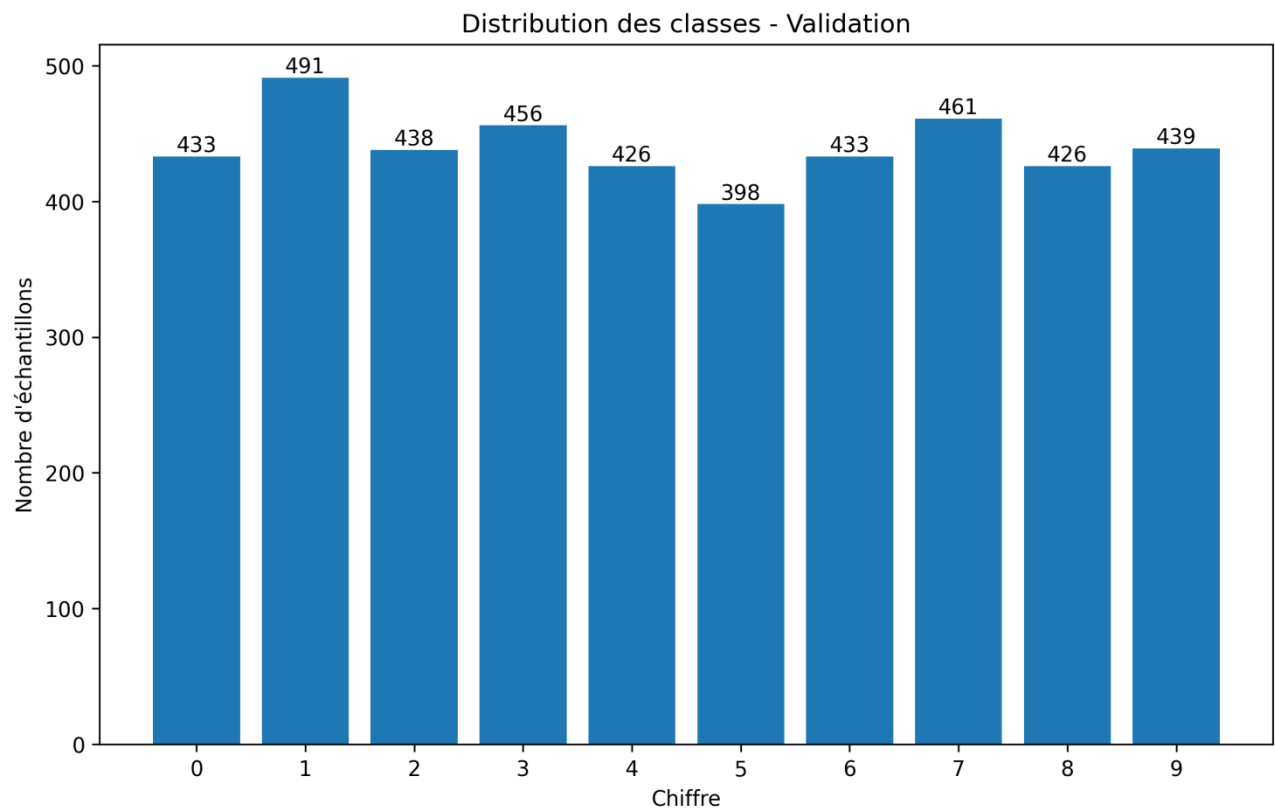
Exemples représentatifs - 8 images par classe



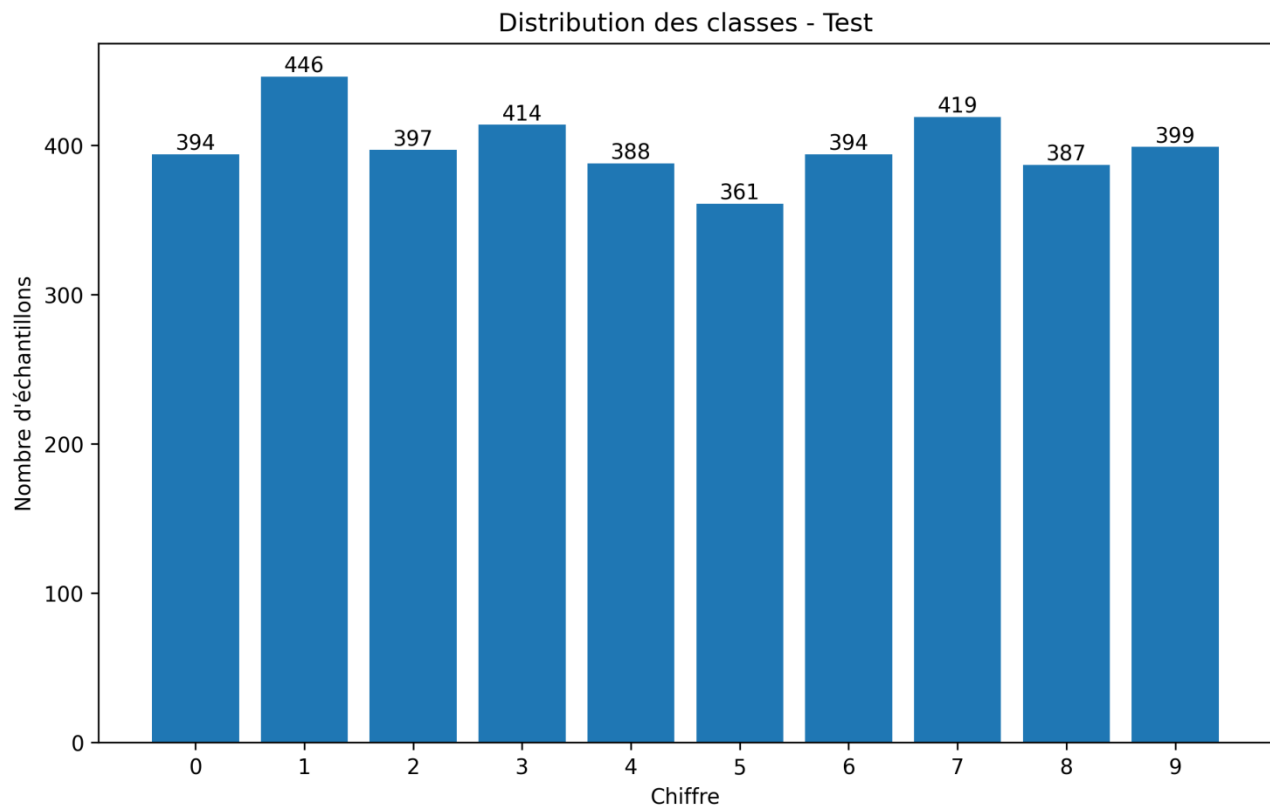
Annexe 2 : Distribution des classes dans l'ensemble d'entraînement



Annexe 3 : Distribution des classes dans l'ensemble de validation



Annexe 4 : *Distribution des classes dans l'ensemble de test*



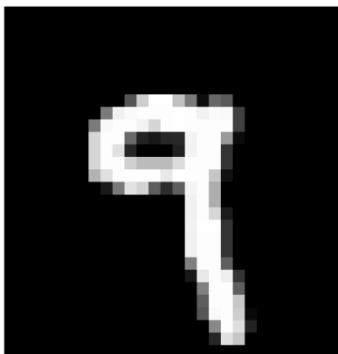
Annexe 5 : Exemples de prédictions

Exemples correctement vs incorrectement classés

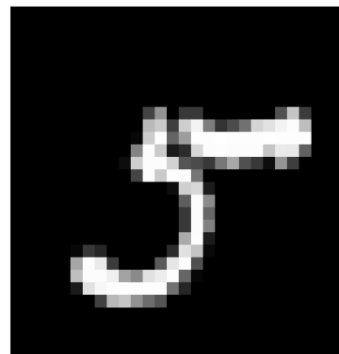
Correct: Vrai 3 → Prédit 3



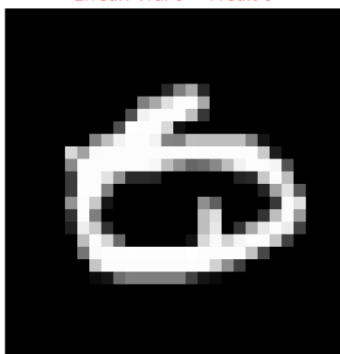
Correct: Vrai 9 → Prédit 9



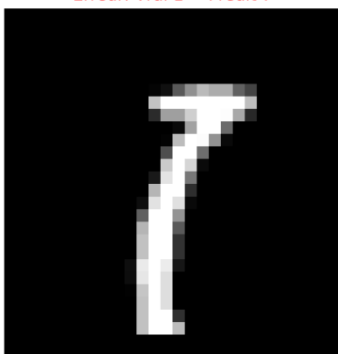
Correct: Vrai 5 → Prédit 5



Erreur: Vrai 0 → Prédit 6



Erreur: Vrai 1 → Prédit 7



Erreur: Vrai 6 → Prédit 4

