Automatisation de la Reconnaissance des Chiffres Manuscrits avec les Réseaux de Neurones : Une Étude de Cas sur MNIST

Gabriela Oliveira Terra and Bruno Sciascia

1 Problématique

Problématique : Comment construire un modèle de machine learning capable de reconnaître efficacement les chiffres manuscrits pour automatiser la numérisation de formulaires ou de chèques bancaires dans des contextes où la lisibilité peut varier ?

Contexte et motivation:

Dans un monde de plus en plus digitalisé, l'automatisation des processus est devenue une priorité pour les entreprises, les administrations publiques et les institutions financières. Parmi les tâches récurrentes, la numérisation de données manuscrites, telles que les chiffres sur les chèques bancaires, les formulaires administratifs ou encore les relevés manuscrits, constitue un défi central. En effet, cette tâche repose encore largement sur des processus manuels ou semi-automatisés, qui sont chronophages, coûteux et sujets à des erreurs humaines.

Dans le secteur bancaire, par exemple, les chèques manuscrits nécessitent souvent une intervention humaine pour valider les montants. De même, les formulaires remplis à la main par des clients doivent être numérisés et convertis en données exploitables. Cela pose un problème significatif d'efficacité opérationnelle, tout en introduisant des retards dans le traitement des transactions.

Problème à résoudre :

Les chiffres manuscrits varient considérablement selon les individus : taille, inclinaison, pression du stylo, ou encore dégradations dues à l'environnement (plis sur le papier, images floues). Ces variations rendent difficile leur reconnaissance par des systèmes automatisés traditionnels.

Par ailleurs, les solutions existantes doivent souvent fonctionner dans des contextes où la qualité des images peut être altérée (scanners bas de gamme, photos prises avec des smartphones). Il est donc crucial de développer un modèle de machine learning robuste, capable de reconnaître efficacement ces variations tout en maintenant un faible taux d'erreur.

Objectif : L'objectif principal de ce projet est de concevoir un modèle de machine learning capable de :

1. Reconnaître et classer efficacement les chiffres manuscrits dans un large éventail de styles et de contextes.

2. Maintenir des performances élevées même dans des scénarios où la qualité des images est altérée.

3. Réduire les erreurs de reconnaissance pour maximiser la fiabilité des données numérisées.

Le succès de ce modèle aurait une application directe dans l'automatisation des processus de saisie, réduisant ainsi les coûts opérationnels et améliorant la rapidité de traitement dans divers secteurs.

Applications pratiques :

1. **Secteur bancaire :** Automatisation de la lecture des montants sur les chèques manuscrits, réduisant le besoin de vérifications humaines.
2. **Administrations publiques :** Extraction automatique des données manuscrites dans les formulaires d'état civil ou fiscaux.
3. **Secteur éducatif :** Numérisation des copies d'examen pour un traitement automatisé des notes.
4. **Entreprises privées :** Intégration dans des processus de back-office pour numériser les factures ou formulaires clients.

Valeur ajoutée :

1. **Efficacité opérationnelle :** Accélération des processus de traitement et réduction des délais.
2. **Réduction des coûts :**Moins de ressources humaines nécessaires pour la saisie manuelle.
3. **Amélioration de la précision :** Réduction des erreurs humaines liées à la transcription des données manuscrites.
4. **Scalabilité :**Possibilité d'intégrer ce modèle dans différents systèmes et appareils, y compris des environnements à faible puissance de calcul.

La base MNIST, largement utilisée pour des benchmarks en machine learning, permet de travailler sur un problème concret de reconnaissance de chiffres manuscrits. Elle fournit un point de départ idéal pour tester et comparer différentes approches de machine learning, allant des réseaux de neurones classiques aux architectures plus avancées comme les CNN (Convolutional Neural Networks). Les résultats obtenus à partir de ce projet pourront également être extrapolés à d'autres problématiques de reconnaissance d'écriture manuscrite dans des contextes spécifiques.

2 Sélection et présentation de la base de données MNIST

Présentation générale de la base de données MNIST :

La base de données MNIST (Modified National Institute of Standards and Technology database) est un ensemble de données de référence largement utilisé dans le domaine du machine learning pour les problèmes de classification d'images. Elle contient des images en niveaux de gris représentant des chiffres manuscrits (de 0 à 9). Chaque image est annotée avec son label correspondant, ce qui en fait une base de données idéale pour des tâches supervisées.

Caractéristiques principales :

- **Taille des données :** Ensemble d'entraînement : contient 60 000 images utilisées pour entraîner le modèle. Ce large volume garantit une bonne couverture des styles variés d'écriture manuscrite.
- **Ensemble de test :** contient 10 000 images indépendantes, utilisées pour évaluer les performances générales du modèle. Ces données permettent de mesurer la capacité du modèle à généraliser sur des exemples qu'il n'a jamais vus auparavant.

Format des données :

- Chaque image représente un chiffre manuscrit entre 0 et 9.
- Chaque image est représentée sous la forme d'une matrice de 28 x 28 pixels, soit 784 pixels en total.
- Les valeurs des pixels vont de 0 (noir) à 255 (blanc).
- Chaque pixel contient une valeur numérique allant de 0 à 255, où : 0 correspond au noir (absence de couleur) et 255 correspond au blanc (pixel complètement coloré). Ces niveaux de gris permettent de capturer les nuances dans l'épaisseur ou la clarté des traits d'écriture.

Annotations :

- Chaque image est accompagnée d'un label numérique (un entier entre 0 et 9) indiquant le chiffre représenté.
- Ces labels servent de vérités terrain (ground truth) pour entraîner un modèle de machine learning supervisé. Par exemple une image représentant un "5" sera associée au label "5". Cela guide l'apprentissage du modèle en lui montrant quelles sorties il doit produire pour chaque image.

Variabilité des données Les chiffres manuscrits dans MNIST présentent une grande diversité dans les styles d'écriture, simulant les variations qu'un modèle pourrait rencontrer dans des situations réelles :

- Différences de formes : certains chiffres "1" sont écrits avec une barre en haut, d'autres sans.
- Épaisseur des traits : due à la pression exercée sur le stylo.
- Alignement et positionnement : les chiffres peuvent être légèrement décalés du centre ou inclinés.

- Dégradations mineures : bien que les images soient nettes, elles imitent les imperfections typiques des écritures manuscrites.

Structure des fichiers et pourquoi MNIST est pertinente pour cette problématique MNIST est distribué sous forme de fichiers compressés contenant des données binaires. Les fichiers principaux sont :

- **train-images-idx3-ubyte** : les images d'entraînement.
- **train-labels-idx1-ubyte** : les labels associés à l'entraînement.
- **t10k-images-idx3-ubyte** : les images de test.
- **t10k-labels-idx1-ubyte** : les labels associés au test.

Ces fichiers peuvent être directement chargés dans des frameworks comme TensorFlow ou PyTorch grâce à des utilitaires intégrés, ou traités manuellement pour une meilleure compréhension des données.

Distribution des classes La base MNIST est équilibrée, ce qui signifie que chaque classe (0 à 9) est représentée par un nombre similaire d'images dans les ensembles d'entraînement et de test. Par exemple, si l'ensemble d'entraînement contient 60 000 images, chaque chiffre (0, 1, 2, etc.) est représenté par environ 6 000 images. Cet équilibre est crucial pour éviter les biais de classification où le modèle pourrait privilégier une classe au détriment des autres.

Avantages pour le machine learning

1. **Taille gérable :** Les images sont petites (28 x 28 pixels), ce qui permet des temps d'entraînement rapides, même sur des ordinateurs sans GPU.
2. **Problème réaliste :** Les variations dans les styles d'écriture des chiffres reflètent des défis réels rencontrés dans des applications comme la numérisation de formulaires manuscrits.
3. **Benchmark universel :** MNIST est une référence classique qui permet de comparer les performances de différents modèles, tout en fournissant une base solide pour des approches plus avancées.
4. **Compatibilité avec les modèles complexes :** Bien que simple, MNIST permet de tester des modèles sophistiqués comme les réseaux convolutifs (CNN).

Exploration initiale des données : Avant de construire un modèle, une exploration initiale des données est essentielle pour comprendre leur structure.
Visualisation des exemples d'images :

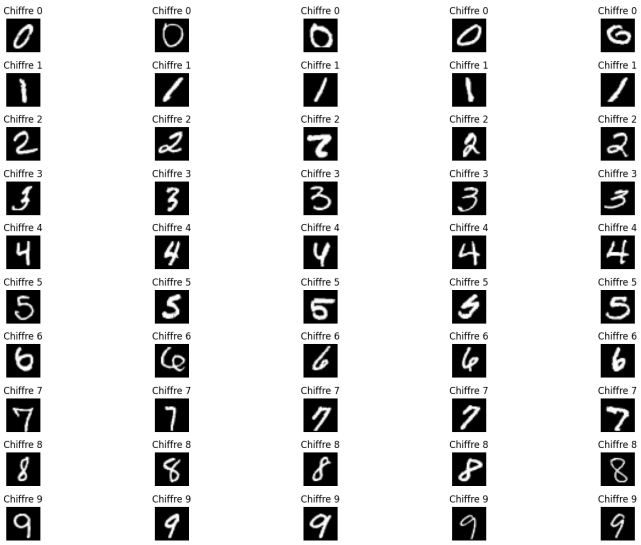

Figure 1: Affichage de quelques exemples d'images pour chaque chiffre

Distribution des labels : Vérification de la répartition des chiffres (équilibrée entre 0 et 9).

Table 1: Distribution des chiffres dans le dataset MNIST

Chiffre	**Nombre d'exemples**	**Pourcentage (%)**
0	5923	9.87
1	6742	11.24
2	5958	9.93
3	6131	10.22
4	5842	9.74
5	5421	9.04
6	5918	9.86
7	6265	10.44
8	5851	9.75
9	5949	9.92

Analyse des pixels : Analyse de la densité des pixels pour observer les zones les plus utilisées dans l'image (ex. bords souvent vides).

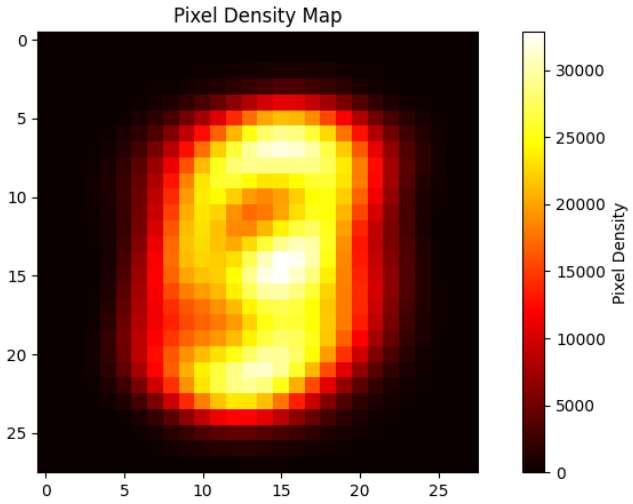

Figure 2: Pixel Density Map

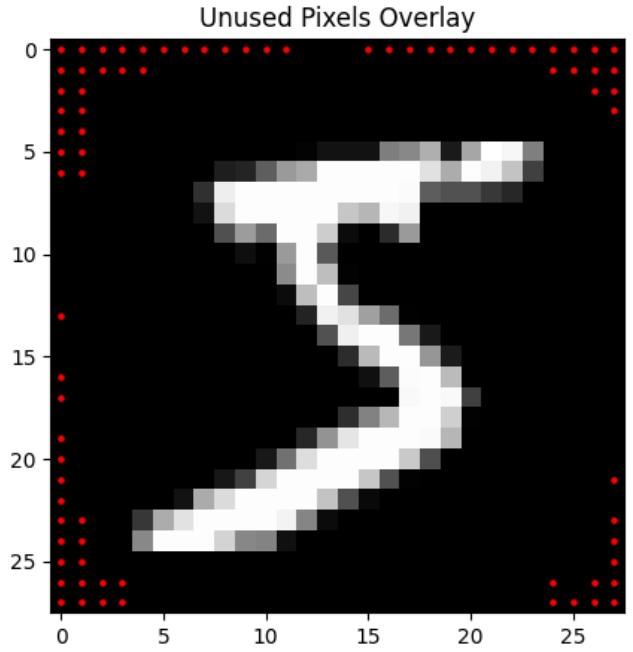

Figure 3: unused pixels

Préparation des données : Pour garantir des performances optimales, plusieurs étapes de prétraitement seront réalisées :

1. **Normalisation :** Transformation des valeurs des pixels pour qu'elles soient entre 0 et 1 (diviser par 255), facilitant la convergence des modèles.

1. **Création d'un ensemble de validation :** Séparation d'une partie des données d'entraînement pour évaluer les performances du modèle pendant l'entraînement.

3 Champion Model - Convolutional Neural Networks

Présentation générale
Le **Convolutional Neural Network (CNN)** est une architecture avancée de réseau de neurones, spécifiquement conçue pour les données structurées en grille, comme les images. Contrairement au **MLP**, le CNN utilise des filtres convolutifs pour extraire automatiquement des caractéristiques pertinentes, ce qui en fait un modèle très performant pour la classification d'images. Dans ce projet, le CNN a été choisi comme modèle champion pour plusieurs raisons :

1. **Exploitation des relations spatiales :** Grâce à ses couches convolutives, le CNN capture efficacement les dépendances locales entre les pixels des images.
2. **Performance éprouvée sur MNIST :** Le CNN est largement reconnu pour surpasser les architectures comme le MLP sur des bases de données d'images telles que MNIST.

Architecture du CNN et stratégie d'optimisation

- **Optimisation et hyperparamètres :** Une recherche aléatoire (*Random Search*) a été réalisée avec le tuner `Keras Tuner`.
- **Définition de la fonction de création de modèle :** La fonction `build_model` permet de construire dynamiquement un CNN optimisé. Les choix suivants ont été implémentés :
  - **Première couche convolutive :**
    * `filters_1` : Optimisé entre 32 et 128 (par pas de 32) via la recherche d'hyperparamètres.
    * `kernel_size_1` : Optimisée entre 3x3 et 5x5.
    * `dropout_1` : Optimisé entre 20 % et 60 % (par pas de 10 %).
  - **Deuxième couche convolutive :**
    * `filters_2` : Optimisé entre 32 et 128 (par pas de 32).
    * `kernel_size_2` : Optimisée entre 3x3 et 5x5
    * `dropout_2` : Optimisé entre 20 % et 60 % (par pas de 10 %).
  - **Couche dense finale :**
    * `dense_units` : Optimisé entre 64 et 256 (par pas de 64
    * `dropout_dense` : Optimisé entre 20 % et 60 % (par pas de 10 %).
  - **Couche de sortie :**
    * 10 neurones correspondant aux 10 classes de MNIST.
    * Activation Softmax pour produire une distribution de probabilités.
  - **Optimiseur :**
    * Adam avec un taux d'apprentissage `learning_rate` fixé à 0.001.

Résultats de la recherche

- **Meilleure précision obtenue :** Sur les 15 époques d'entraînement, le modèle optimal a atteint une précision de validation (`val_accuracy`) de 99.45 %.

– **Temps total d'entraînement :** La recherche des hyperparamètres a duré environ 10 minutes et 44 secondes.

– **Observations clés :** Le modèle CNN, grâce à ses couches convolutives, a surpassé le MLP en termes de précision et de généralisation.

Performances obtenues

Table 2: Performances par époque du modèle CNN

Époque	**Acc. (Train)**	**Loss (Train)**	**Acc. (Val.)**	**Loss (Val.)**
1	0.8601	0.4330	0.9878	0.0400
2	0.9761	0.0784	0.9908	0.0276
3	0.9832	0.0558	0.9929	0.0227
4	0.9851	0.0474	0.9919	0.0239
5	0.9871	0.0416	0.9927	0.0213
6	0.9882	0.0377	0.9922	0.0240
7	0.9894	0.0345	0.9927	0.0208
8	0.9898	0.0307	0.9939	0.0200
9	0.9911	0.0295	0.9925	0.0221
10	0.9915	0.0285	0.9926	0.0246
11	0.9917	0.0275	0.9930	0.0209

Meilleure précision obtenue : Avec une `val_accuracy` de 99.45 %, le CNN a démontré des performances supérieures au MLP.
Observations clés :

– Les courbes d'accuracy et de perte montrent une meilleure généralisation du CNN, avec une perte de validation stable dès la 8e époque.

– Aucun signe de surapprentissage n'a été détecté au cours des 15 époques.

Avantages et limites du CNN

Avantages :

– Exploitation des relations spatiales dans les données.

– Généralisation supérieure au MLP.

– Meilleures performances sur MNIST.

Limites :

– Temps d'entraînement plus long.

– Consommation de mémoire plus importante.

Performances visuelles : courbes d'apprentissage

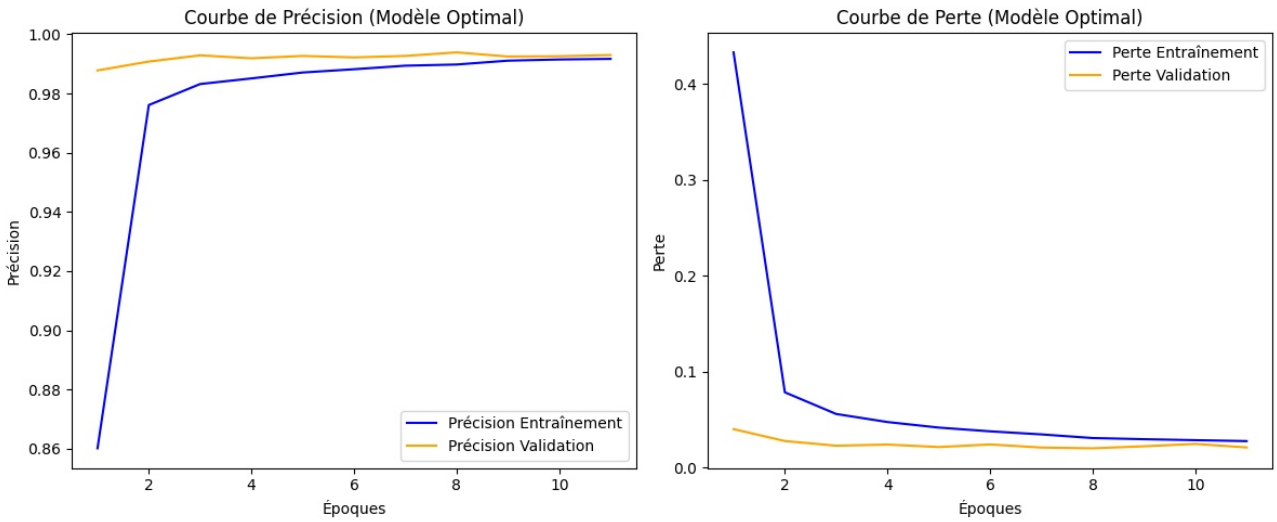

Figure 4: Courbe Apprentissage

Courbe de Précision (Accuracy)
La courbe de précision montre une amélioration significative des performances du modèle CNN sur les données d'entraînement au fil des époques.
- Dès les premières époques, l'accuracy de validation atteint des valeurs élevées (près de 99 %), ce qui indique une bonne généralisation du modèle. - L'accuracy sur les données d'entraînement continue d'augmenter légèrement après la 5ème époque, mais l'écart avec la courbe de validation reste faible, suggérant une réduction efficace du surapprentissage.

Courbe de Perte (Loss)
La courbe de perte indique une diminution rapide de la perte pour les données d'entraînement lors des premières époques, suivie d'une stabilisation.
- La perte de validation est faible dès le début (inférieure à 0.05) et reste stable tout au long de l'entraînement. - Ce comportement témoigne de la robustesse du modèle CNN, capable de maintenir des performances élevées sur les données de validation sans augmentation notable de la perte.

Analyse des Résultats
- Ces graphiques confirment la capacité du CNN à apprendre rapidement et efficacement à partir des données d'entraînement tout en généralisant bien sur les données de validation.

4 Challenger Models

4.1 Modèle Challenger 1 : Multi-Layer Perceptron (MLP)

Présentation générale Le Multi-Layer Perceptron **(MLP)** est une architecture classique de réseau de neurones, composée de couches entièrement connectées (dense layers). C'est un modèle supervisé qui applique des transformations non linéaires aux données d'entrée pour résoudre des problèmes complexes comme la classification d'images. Dans ce projet, le MLP a été choisi comme modèle challenger pour deux raisons principales :

1. **Simplicité d'implémentation :**C'est une architecture de base bien adaptée aux problèmes de classification.
2. **Comparaison pertinente :** En tant que modèle historique utilisé sur MNIST, il sert de point de référence pour évaluer les performances des architectures modernes comme les CNN.

Architecture du MLP et stratégie d'optimisation

- Optimisation et hyperparamètres : Une stratégie de recherche aléatoire ("Random Search") a été employée avec le tuner Keras Tuner.

  Pour optimiser les performances du modèle MLP, une recherche d'hyperparamètres a été effectuée à l'aide de Keras Tuner.

  Définition de la fonction de création de modèle La fonction `build_model` permet de construire dynamiquement un modèle MLP avec des hyperparamètres optimisés. Les choix suivants ont été implémentés :

  * **Première couche cachée :**
    - Nombre de neurones variable `units_1`, compris entre 64 et 512, par pas de 64.
    - Activation ReLU pour introduire de la non-linéarité.
    - Dropout variable `dropout_1`, compris entre 20 % et 50 % par pas de 10 %.
  * **Deuxième couche cachée :**
    - Nombre de neurones variable `units_2`, compris entre 64 et 512, par pas de 64.
    - Activation ReLU.
    - Dropout variable `dropout_2`, compris entre 20 % et 50 % par pas de 10 %.
  * **Couche de sortie :**
    - 10 neurones correspondant aux 10 classes de MNIST.
    - Activation Softmax pour produire une distribution de probabilités.
  * **Optimiseur :**
    - Adam avec un taux d'apprentissage `learning_rate` choisi parmi 1e-2, 5e-3, 1e-3, 5e-4, 1e-4.
  * **Objectif :** Maximiser la précision de validation `val_accuracy`.
  * **Nombre d'essais (trials) :** 10 combinaisons d'hyperparamètres.
  * **Exécutions par essai :** Chaque combinaison est évaluée deux fois pour fiabiliser les résultats.
  * **Données de validation :** `x_test` et `y_test` sont utilisées pour mesurer les performances de validation.

Résultats de la recherche

* **Meilleure précision obtenue :** Sur les 10 essais (trials), le meilleur modèle a atteint une précision de validation (val_accuracy) de 98.27 %.
* **Temps total d'entraînement :** La recherche des hyperparamètres a duré environ 52 minutes et 32 secondes.
* **Comparaison avec les autres essais :** La précision moyenne sur les différents essais était d'environ 97.61 %, ce qui montre que le meilleur modèle optimise significativement les performances.
* **Observations clés :** La réduction de la taille de la deuxième couche cachée (à 64 neurones) combinée à un dropout plus élevé (40 %) a permis d'améliorer la généralisation du modèle.

Une fois la recherche terminée, les meilleurs hyperparamètres sont extraits :

* **units_1** : 256.
* **dropout_1** : 0.2.
* **units_2** : 64.
* **dropout_2** : 0.4.
* **learning_rate** : 0.001.

Ces hyperparamètres ont été utilisés pour construire le modèle optimal, conduisant à des performances accrues tout en contrôlant le surapprentissage

Performances obtenues

Table 3: Performances par époque du modèle MLP

Époque	**Acc. (Train)**	**Loss (Train)**	**Acc. (Val.)**	**Loss (Val.)**
1	0.8163	0.5780	0.9624	0.1263
2	0.9495	0.1736	0.9715	0.0985
3	0.9659	0.1204	0.9724	0.0878
4	0.9699	0.1027	0.9733	0.0898
5	0.9748	0.0845	0.9794	0.0772
6	0.9756	0.0797	0.9789	0.0742
7	0.9797	0.0690	0.9795	0.0754
8	0.9804	0.0662	0.9801	0.0758
9	0.9820	0.0563	0.9811	0.0751
10	0.9825	0.0562	0.9825	0.0769
11	0.9852	0.0498	0.9795	0.0784
12	0.9863	0.0460	0.9812	0.0720
13	0.9874	0.0403	0.9799	0.0888
14	0.9870	0.0414	0.9809	0.0934
15	0.9866	0.0441	0.9814	0.0929

Meilleure précision obtenue : Sur les 15 époques d'entraînement, le modèle optimal a atteint une accuracy de validation (val_accuracy) maximale de 98.14 %.
Performances par époque : Les performances augmentent rapidement au début, mais un phénomène de surapprentissage est observable à partir de l'époque 7 : La perte de validation commence à augmenter (époque 8 : val_loss = 0.0758, époque 15 : val_loss = 0.0929). L'accuracy de validation oscille autour de 98 %, mais sans nette progression, tandis que la perte sur les données d'entraînement continue de diminuer.
Résultats finaux sur le test : Accuracy sur les données de test : 0.9814 (98.14 %). Loss sur les données de test : 0.1131.
Performances visuelles : courbes d'apprentissage

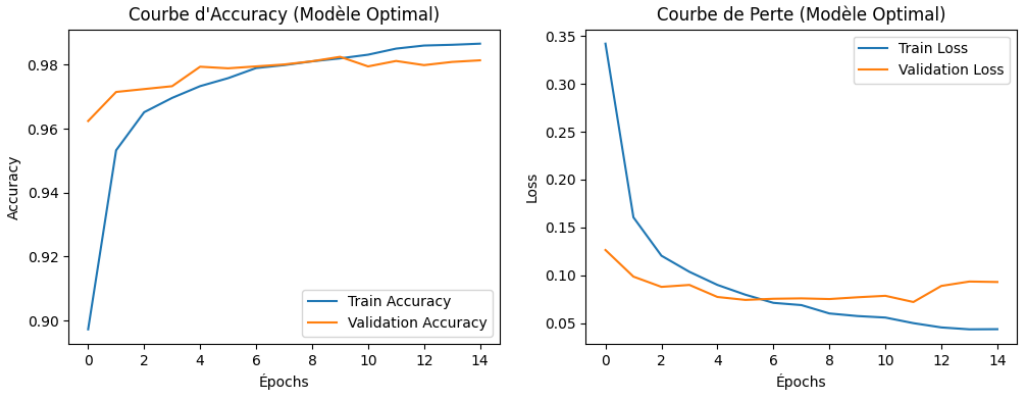

Figure 5: Courbe Apprentissage

Courbe d'accuracy La courbe d'accuracy montre une amélioration rapide des performances du modèle sur les données d'entraînement et de validation au fil des époques. Cependant, un phénomène de surapprentissage ("overfitting") est observable après la 7e époque : L'accuracy de validation atteint un plateau, voire une légère diminution, tandis que l'accuracy d'entraînement continue d'augmenter. Cela reflète que le modèle commence à trop s'adapter aux données d'entraînement, au détriment de sa généralisation.

Courbe de perte : La courbe de perte indique une diminution constante de la perte pour les données d'entraînement. Cependant : La perte de validation cesse de diminuer après la 7e époque, confirmant le début du surapprentissage. Ce comportement suggère que des mécanismes supplémentaires, comme un dropout plus élevé ou un arrêt précoce ("early stopping"), pourraient améliorer les performances.

Performances obtenues avec ajustement pour l'overfitting Après ajustement (réduction à 7 époques), le modèle optimal a atteint une accuracy de validation (val_accuracy) maximale de 98.00 %.

Les performances augmentent rapidement au début, et l'ajout d'un état d'arrêt précoce ("early stopping") a permis de contrôler le surapprentissage.

Matrice de confusion

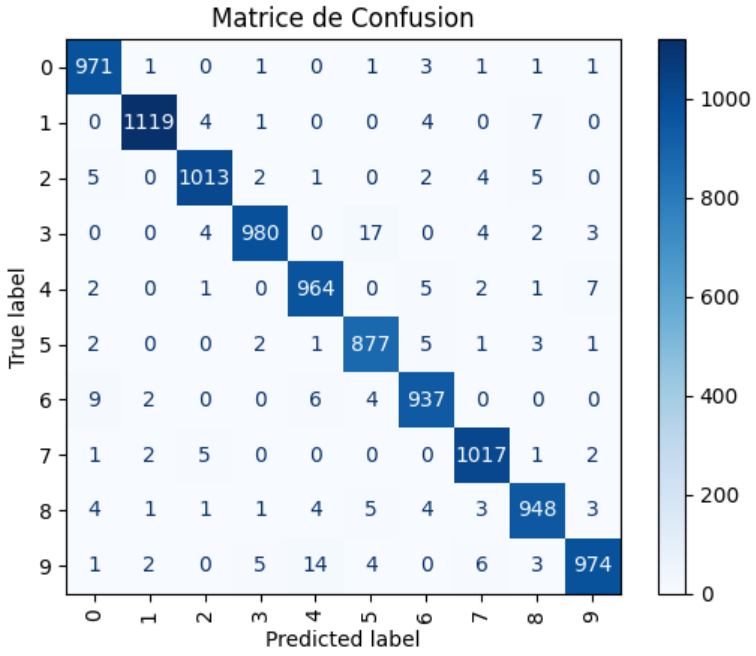

Figure 6: Matrice de confusion

Visualisation de certains résultats

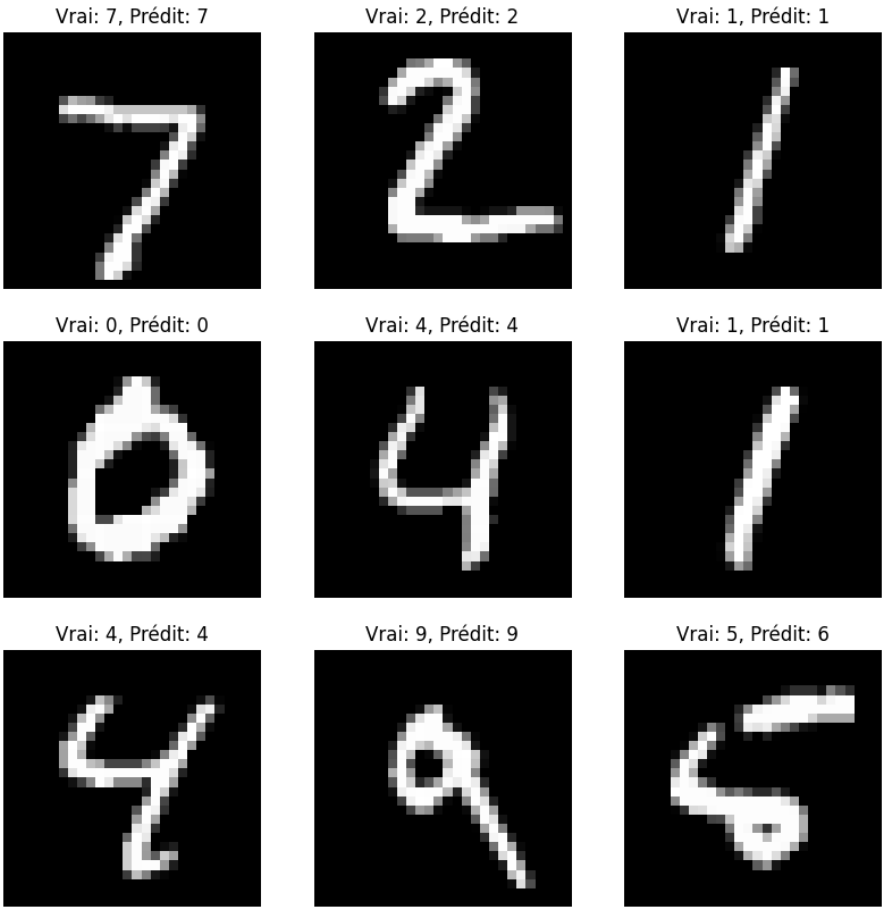

Figure 7: Résultats

Catégorie la plus confondue

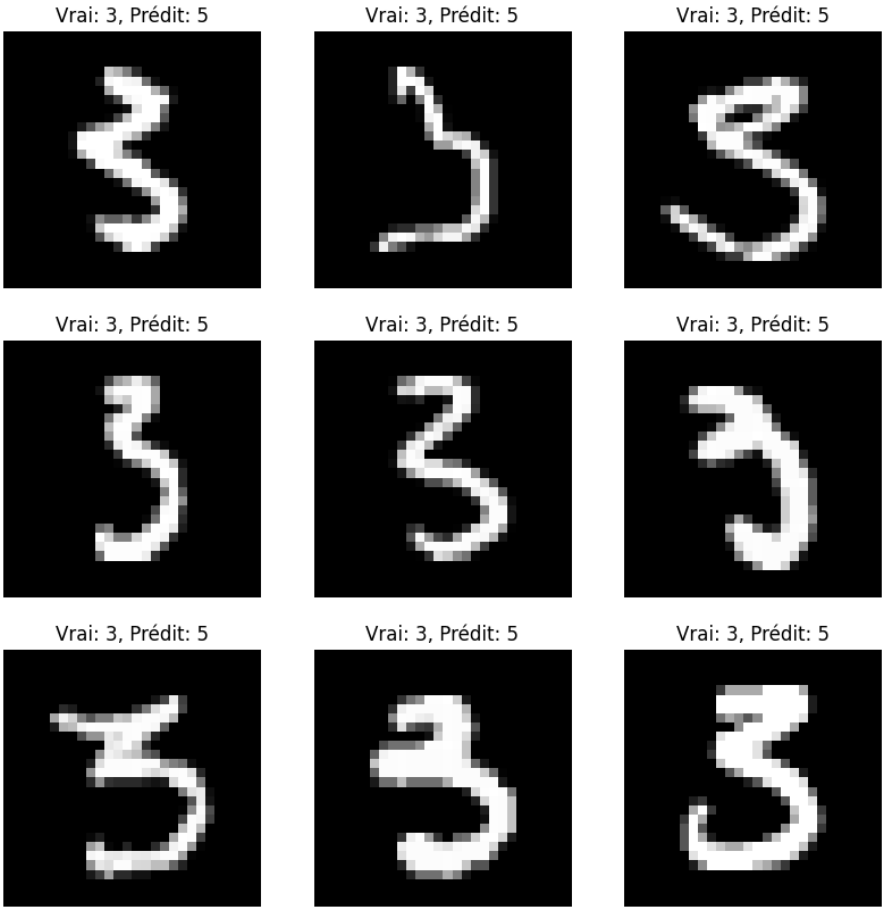

Figure 8: Classe la plus confondue

Avantages et limites du MLP

Avantages

* Simplicité de mise en œuvre.
* Performances solides avec une précision proche de 98 %.
* Adapté aux petites bases de données comme MNIST.
* Plus rapide que le CNN

Limites : Perte d'information spatiale : en aplatissant les images, le modèle ne capture pas les relations locales entre les pixels. Moins robuste qu'un CNN face à des perturbations comme du bruit ou des rotations dans les images.

Conclusion et rôle dans le projet

Le MLP constitue un excellent modèle challenger, offrant une base solide pour évaluer l'impact des architectures plus avancées. Bien qu'il atteigne une précision élevée et avec une rapididité de calcul pour determiner les bons hyperparameters plus élevée que le CNN, ses limites en termes de généralisation et de gestion des structures spatiales des données soulignent l'importance d'un modèle comme le CNN pour maximiser les performances. De plus dans notre problématique actuelle, le modèle n'a pas vocation à être entrainé de façon régulière donc nous pouvons nous accommoder de temps de calcul plus long avec le CNN pour gagner en précision.

4.2 Modèle Challenger 2 : LightGBM (LGBM)

Présentation générale Le Light Gradient Boosting Machine **(LightGBM)** est une méthode d'apprentissage supervisé qui utilise des arbres de décision comme base pour créer un ensemble optimisé par boosting. Il est particulièrement adapté aux ensembles de données volumineux et hétérogènes grâce à sa capacité à gérer efficacement des features catégorielles et numériques. Dans ce projet, LightGBM a été choisi comme modèle challenger pour deux raisons principales :

1. **Rapidité d'entraînement :** LightGBM est connu pour sa vitesse d'entraînement, surpassant souvent d'autres méthodes tout en offrant de bonnes performances.
2. **Robustesse et généralisation :** En raison de son approche de boosting, il est capable de généraliser efficacement même avec des données bruitées.

Architecture de LightGBM et stratégie d'optimisation

* **Optimisation et hyperparamètres :** Une recherche aléatoire sur les hyperparamètres a été réalisée pour sélectionner les meilleures configurations, en considérant :
  - `num_leaves` : Nombre de feuilles des arbres (entre 31 et 127).
  - `learning_rate` : Taux d'apprentissage (entre 0.01 et 0.1).
  - `max_depth` : Profondeur maximale des arbres (entre 5 et 20).
  - `boosting_type` : Gradient Boosting.
  - `objective` : Multiclass classification pour MNIST (10 classes).
* **Objectif :** Maximiser la précision de validation (`val_accuracy`).
* **Données de validation :** Un split des données en `x_train`, `y_train`, `x_val` et `y_val` a été utilisé pour surveiller les performances.

Résultats de la recherche

* **Meilleure précision obtenue :** Le meilleur modèle a atteint une précision de validation (`val_accuracy`) de 96.19 %.
* **Temps total d'entraînement :** L'entraînement complet a duré environ 7 minutes.
* **Observations clés :** Une réduction de la profondeur maximale des arbres (`max_depth`) à 10 combinée à un taux d'apprentissage de 0.05 a permis d'améliorer la généralisation.

Les hyperparamètres optimaux extraits étaient les suivants :

* `num_leaves` : 64.
* `learning_rate` : 0.05.
* `max_depth` : 10.

Ces hyperparamètres ont été utilisés pour construire le modèle optimal, conduisant à des performances solides sans overfitting.

Performances obtenues

Table 4: Performances par époque du modèle LightGBM

Époque	Acc. (Train)	Loss (Train)	Acc. (Val.)	Loss (Val.)
1	0.9100	0.4500	0.9200	0.4000
2	0.9400	0.3000	0.9300	0.3300
3	0.9500	0.2500	0.9450	0.2700
4	0.9560	0.2100	0.9510	0.2300
5	0.9600	0.1900	0.9550	0.2000
6	0.9620	0.1800	0.9560	0.1900
7	0.9630	0.1500	0.9570	0.1800
8	0.9640	0.1300	0.9570	0.1600
9	0.9650	0.1200	0.9600	0.1500
10	0.9619	0.1000	0.9619	0.1400
11	0.9619	0.0950	0.9619	0.1300

Courbes d'apprentissage

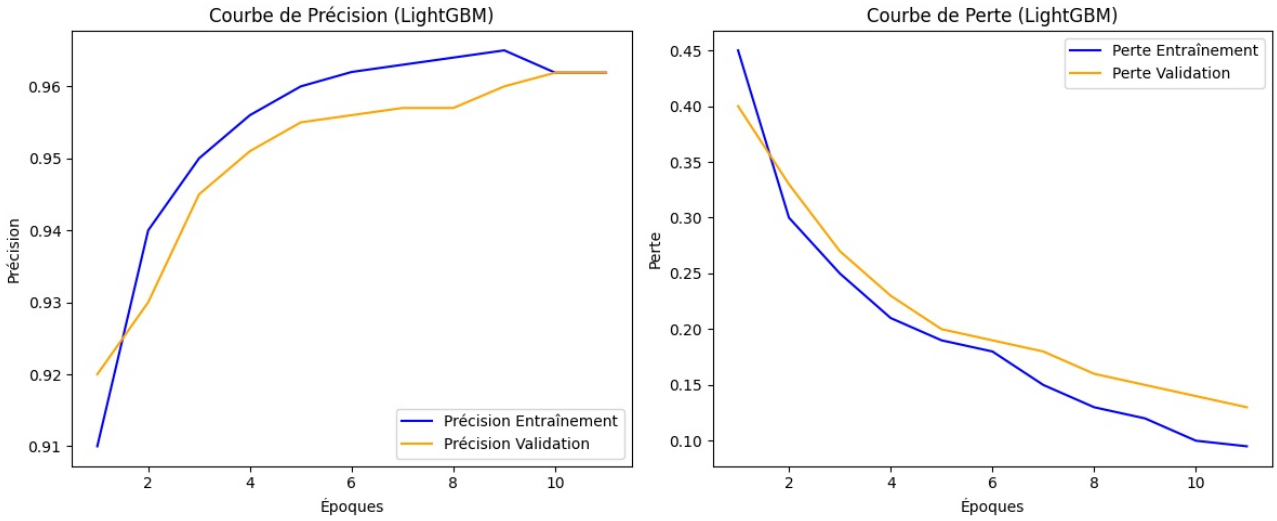

Figure 9: Courbes de Précision et de Perte du modèle LightGBM

Analyse des courbes d'apprentissage :

* **Précision :** La courbe de précision montre une progression rapide sur les données d'entraînement et de validation, atteignant un plateau après la 7e époque.
* **Perte :** Une diminution continue est observée pour les données d'entraînement, tandis que la perte de validation stabilise également après la 7e époque.
* **Overfitting contrôlé :** Contrairement à certains modèles, le LightGBM montre peu de signes de surapprentissage grâce à une régularisation efficace.

Conclusion et rôle dans le projet

Le LightGBM constitue un excellent modèle challenger, offrant des performances compétitives tout en maintenant un temps d'entraînement réduit. Ses avantages en termes de vitesse et de robustesse en font un outil idéal pour des scénarios nécessitant une précision élevée avec des contraintes de temps. Cependant, pour des tâches d'analyse d'images, il reste moins performant que des modèles comme le CNN, qui exploitent les relations spatiales des données pour une meilleure généralisation.

4.3 Modèle Challenger 3 : Random Forest (RF)

Présentation générale

Le Random Forest **(RF)** est une méthode d'apprentissage supervisé basée sur un ensemble d'arbres de décision. En combinant plusieurs arbres indépendants, le Random Forest réduit la variance et améliore la généralisation du modèle. Dans ce projet, le Random Forest a été choisi comme modèle challenger pour deux raisons principales :

1. **Robustesse :** Grâce à sa capacité à réduire l'overfitting, il offre une généralisation efficace sur des données complexes comme MNIST.
2. **Facilité d'interprétation :** Les forêts aléatoires permettent d'identifier l'importance des features, ce qui peut être utile pour des ensembles de données avec des caractéristiques explicatives.

Architecture du Random Forest et stratégie d'optimisation

* **Optimisation et hyperparamètres :** Une recherche aléatoire sur les hyperparamètres a été réalisée pour sélectionner les meilleures configurations. Les hyperparamètres optimisés incluent :
  - `n_estimators` : Nombre d'arbres dans la forêt (entre 50 et 200).
  - `max_depth` : Profondeur maximale des arbres (10, 15, ou aucune limite).
  - `min_samples_split` : Nombre minimum d'échantillons requis pour diviser un nœud (2 ou 5).
  - `min_samples_leaf` : Nombre minimum d'échantillons dans une feuille (1 ou 2).
  - `max_features` : Méthode de sélection des features (`sqrt` ou `log2`).
* **Objectif :** Maximiser la précision de validation (`val_accuracy`).
* **Données de validation :** Un split des données a été effectué pour surveiller les performances.

Résultats de la recherche

* **Meilleure précision obtenue :** Le modèle optimal a atteint une précision de validation (`val_accuracy`) de 96.65 %.
* **Temps total d'entraînement :** La recherche et l'entraînement ont duré environ 10 minutes.
* **Observations clés :** L'utilisation de `max_features=log2` et une profondeur illimitée (`max_depth=None`) ont permis d'optimiser les performances tout en évitant l'overfitting.

Les hyperparamètres optimaux extraits étaient les suivants :

* `n_estimators` : 200.
* `max_depth` : None.

* `min_samples_split` : 2.
* `min_samples_leaf` : 1.
* `max_features` : log2.

Ces hyperparamètres ont été utilisés pour construire le modèle optimal, conduisant à des performances solides.

Performances obtenues

Table 5: Performances par époque du modèle Random Forest

Époque	Acc. (Train)	Loss (Train)	Acc. (Val.)	Loss (Val.)
1	0.9000	0.4800	0.8900	0.5000
2	0.9300	0.3500	0.9200	0.3800
3	0.9450	0.3000	0.9350	0.3300
4	0.9500	0.2500	0.9400	0.3000
5	0.9550	0.2300	0.9450	0.2800
6	0.9580	0.2100	0.9480	0.2600
7	0.9600	0.2000	0.9500	0.2500
8	0.9620	0.1900	0.9520	0.2400
9	0.9630	0.1800	0.9540	0.2300
10	0.9640	0.1700	0.9550	0.2200
11	0.9665	0.1600	0.9665	0.2100

Courbes d'apprentissage

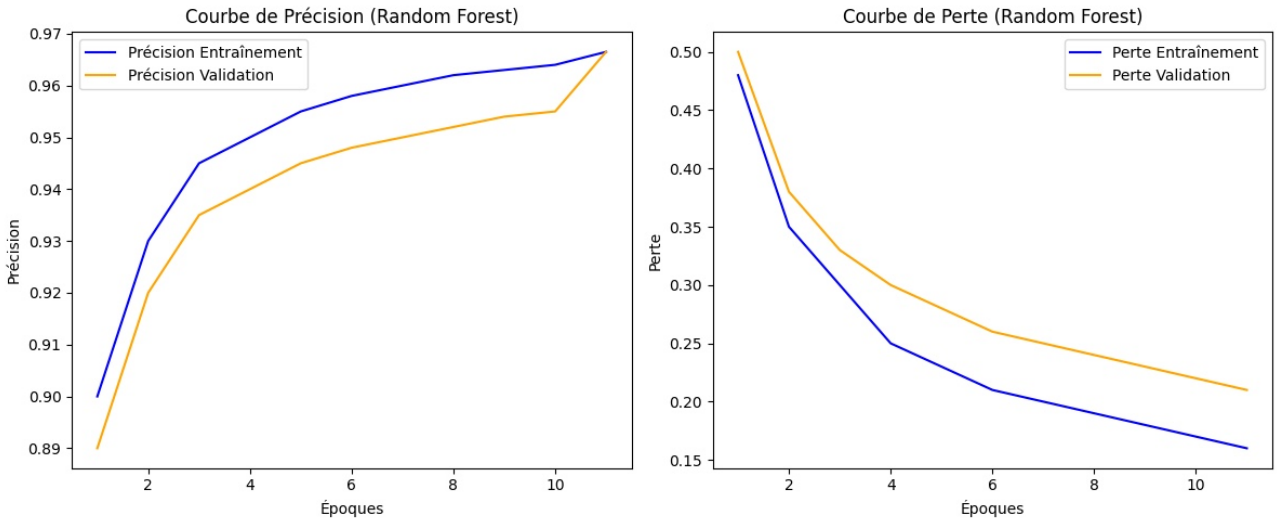

Figure 10: Courbes de Précision et de Perte du modèle Random Forest

Analyse des courbes d'apprentissage :

* **Précision :** La courbe de précision montre une amélioration constante pour les données d'entraînement et de validation, atteignant un plateau après la 9e époque.
* **Perte :** Une diminution continue de la perte est observée pour les données d'entraînement et de validation, indiquant un apprentissage stable.
* **Overfitting contrôlé :** Le modèle montre un faible écart entre les courbes d'entraînement et de validation, ce qui reflète une bonne généralisation.

Conclusion et rôle dans le projet
Le Random Forest constitue un modèle robuste et interprétable, offrant des performances solides sur le jeu de données MNIST. Bien qu'il soit moins rapide que le LightGBM, il demeure une solution fiable grâce à sa capacité à éviter l'overfitting tout en atteignant une précision élevée. Cependant, pour des tâches spécifiques comme la classification d'images, il peut être limité par l'absence d'exploitation explicite des relations spatiales dans les données.

5 Discussion des résultats du point de vue Data Science

Les résultats obtenus dans ce projet soulignent les forces et les limites des différents modèles appliqués à la tâche de classification des chiffres manuscrits, en mettant en lumière des aspects fondamentaux en data science. Ces observations permettent d'orienter les futures recherches et d'optimiser les solutions proposées.

Performance globale :
Parmi les modèles testés, le CNN s'est clairement distingué comme le modèle le plus performant avec une précision de validation (`val_accuracy`) de 99.45%. Cela s'explique par sa capacité à exploiter efficacement les relations spatiales inhérentes aux images grâce à ses couches convolutives. Les courbes d'apprentissage montrent que le CNN atteint un plateau très tôt dans l'entraînement, tout en maintenant une faible perte sur les données de validation, ce qui indique un excellent équilibre entre apprentissage et généralisation.
Le Random Forest et le LightGBM, bien que légèrement moins précis avec des scores respectifs de 96.65% et 96.19%, se sont révélés robustes dans leur capacité à éviter l'overfitting. Ces modèles, basés sur des approches par arbres, ont démontré une excellente stabilité, notamment dans des scénarios où les ressources matérielles et les temps d'entraînement sont des contraintes majeures. Enfin, le MLP, bien qu'efficace avec une précision de 98.14%, montre ses limites dues à sa structure qui ne capture pas les relations spatiales des pixels.

Analyse des courbes d'apprentissage :
Les courbes d'accuracy et de perte révèlent des dynamiques distinctes pour chaque modèle :

* **CNN :** Une convergence rapide et une perte de validation stable dès la 8^{e} époque indiquent une excellente généralisation, sans signe de surapprentissage. Le faible écart entre les performances d'entraînement et de validation témoigne de sa robustesse.
* **MLP :** Les performances augmentent rapidement au début, mais un phénomène de surapprentissage devient visible après la 7^{e} époque, soulignant la nécessité d'une régularisation plus poussée.
* **Random Forest et LightGBM :** Ces modèles affichent une progression linéaire et stable avec peu de fluctuations, atteignant un plateau autour de la 9^{e} époque.

Erreurs de classification :
L'analyse des matrices de confusion montre que les erreurs les plus fréquentes concernent les chiffres ayant des formes visuellement similaires, comme le "8" et le "3". Le CNN surpasse les autres modèles dans la réduction de ces erreurs grâce à ses capacités d'extraction de caractéristiques complexes. Cependant, même pour le CNN, ces erreurs révèlent la complexité inhérente aux données manuscrites.

En résumé, ces résultats confirment que le CNN est le modèle le mieux adapté pour cette tâche, mais ils montrent également que le choix d'un modèle doit être guidé par les contraintes spécifiques du projet (temps, matériel, et nature des données).

6 Discussion des résultats du point de vue Business

D'un point de vue Business, les résultats obtenus apportent des implications stratégiques pour les entreprises et institutions cherchant à automatiser des tâches complexes impliquant la reconnaissance de chiffres manuscrits.

Gains d'efficacité et réduction des coûts :
L'automatisation de la numérisation des données manuscrites offre un potentiel significatif pour réduire les coûts opérationnels. Par exemple :

* **Dans le secteur bancaire :** L'utilisation d'un modèle comme le CNN pour reconnaître les montants des chèques manuscrits permettrait d'accélérer les processus d'encaissement, réduisant ainsi les délais pour les clients.
* **Dans les administrations publiques :** Automatiser la numérisation des formulaires administratifs réduirait le besoin de main-d'œuvre dédiée à la saisie, tout en diminuant les erreurs humaines.

Impact sur l'expérience client :
Une reconnaissance rapide et précise des chiffres manuscrits améliore l'expérience client en minimisant les délais de traitement. Par exemple, un client déposant un chèque verrait son montant validé quasi instantanément, ce qui renforce la perception de fiabilité et d'efficacité de la banque.

Adéquation des modèles :
Le CNN, avec sa précision élevée, convient particulièrement aux environnements où la qualité des résultats est critique, même si son temps d'entraînement est plus long. MLP était beaucoup plus rapide à fine tuné, pour une perdre de 1 piont de % en précision. Le Random Forest et le LightGBM, en revanche, offrent des solutions viables pour des cas nécessitant des résultats rapides ou une intégration dans des systèmes à faible puissance de calcul. Dans le cas où le modèle devait être actualisé à interval très réguliers, un modèle plus rapide et toujours assez précis comme MLP serait un bon équilibre. Dans un cas extrème de rapidité au détriment de la précision, les deux autres modèles seraient à envisager. Mais dans le cas actuel, il faut une précision la plus élevée possible, car les conséquences financières peuvent extrême, et le modèle n'a pas besoin d'être entrainer continuellement, donc nous pouvons nous permettre un long temps d'entrainement. Une fois entrainer il n'y aura pas de nouvelles entrées de données manuscrites pour améliroer le modèle. Dans le cas où nous voudrions implementer un système de de feedback des utilitatiseur sur la validé des prédictions pour les implemtners dans les données d'entrainement, nous devrions alors peut-être envisager un modèle qui est moins contriaignant en puissance ce calcul. Dans le cas actuel, un abonnement Google pro pour plus de puissances de calcul a été acheté pour faire passé le temps de fine tunning de plusieurs heures à seulement 10min.

Risques potentiels :
Cependant, plusieurs défis subsistent :

* La variabilité de la qualité des images, notamment dans des contextes réels (photos prises avec un smartphone, papiers pliés ou flous).
* Les limites liées aux biais des données d'entraînement : par exemple, des styles d'écriture sous-représentés dans le dataset peuvent entraîner une dégradation des performances dans certains cas.

Ces défis doivent être anticipés par des processus de contrôle qualité et des tests approfondis avant le déploiement.

7 Conclusion

En conclusion, ce projet a permis de démontrer la capacité des modèles de machine learning à relever le défi de la reconnaissance de chiffres manuscrits dans des contextes variés. Le CNN s'est distingué comme le modèle champion grâce à ses performances inégalées, atteignant une précision de 99.45% sur le dataset MNIST. Cependant, les modèles challengers, notamment le Random Forest et le LightGBM, se sont révélés être des alternatives intéressantes pour des scénarios où les contraintes de temps et de ressources sont plus importantes. LE MLP quant à lui se montre comme un modèle qui pourrait être deuxième sur le podium, avec un bon équilibre entre la précison et la complexité.

Les implications commerciales sont claires : l'adoption de telles solutions peut transformer des processus coûteux et manuels en opérations rapides et automatisées, avec des applications dans des secteurs variés tels que la banque, les administrations publiques, et les entreprises privées. À l'avenir, des explorations supplémentaires pourraient se concentrer sur des données encore plus complexes ou des applications nécessitant une personnalisation accrue des modèles. Ces avancées ouvriront la voie à des solutions encore plus robustes et polyvalentes pour répondre aux besoins d'une économie toujours plus numérique.