SAYF BEJAOUI
CONSULTANT DATA ENGINNER-SCIENTIST EN ENTREPRISE
FORMATEUR BIG DATA, IA & CLOUD

Digit Recognition

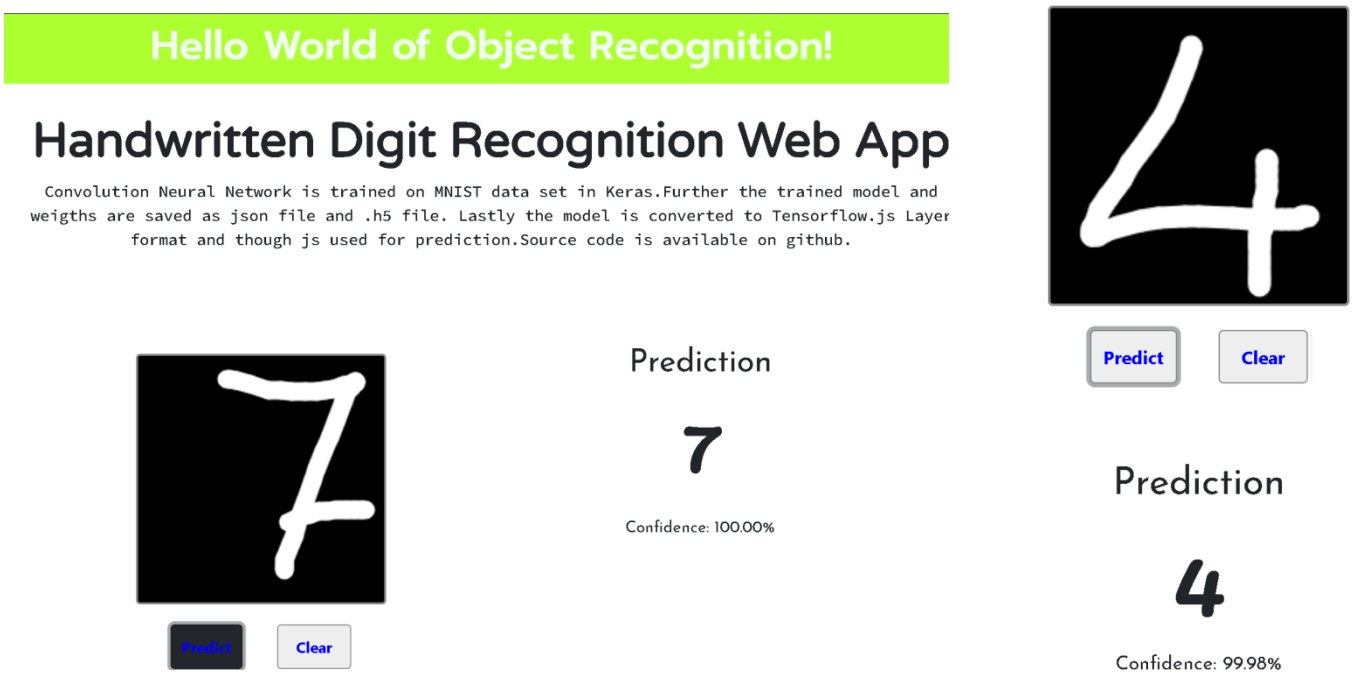
Motivation :

Les exemples suivants montrent des applications pratiques de l'OCR de chiffres manuscrits utilisées dans divers secteurs pour *automatiser la lecture*, *la reconnaissance et la gestion de documents*, ce qui peut entraîner des *gains d'efficacité*, de *productivité* et d'*exactitude* dans les processus administratifs et opérationnels :

- ***Secteur de la poste* :** trier et acheminer automatiquement les lettres et les colis en lisant les codes postaux sur les enveloppes
- ***Secteur bancaire* :** Lire et traiter les chèques manuscrits
- **…**
- Un exemple plus fun : Solving Sudoku in Seconds (or Less!) With Python part1 et part 2

Dans ce projet :

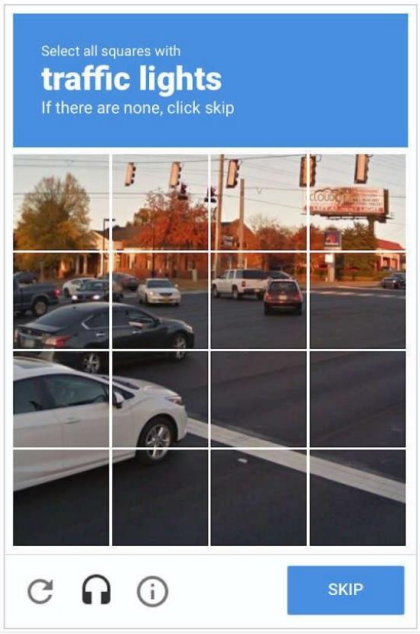
- Vous développerez une WebApp dont le fonctionnement est identique à : https://maneprajakta.github.io/Digit_Recognition_Web_App/



- Elle permettra aux users de dessiner des chiffres manuscrits sur la page d'accueil ; pas besoin de développer un système d'authentification/connexion.

- **Optionnel :** Vous pouvez utilisez ce système de digit recognition pour mettre en place un **captcha**

(où l'User dessine un chiffre, si le chiffre est reconnu en tant que digit, ça débloque la session sur le browser)



- Puis de retenir l'IA la plus performante pour classifier ces images après avoir entraîné / expérimenté au moins 3 modèles (de Machine Learning / Deep Learning) sur le célèbre dataset MNIST (format csv) et MNIST (format images)

  - ✓ N'oublier pas d'optimiser la performance de différents modèles (Decision Tree, RondomForest, Support Vector Machine, CNN, …) en se basant sur les métriques adéquates (confusion matrix, accuracy, …) et en ajustant les hyperparamètres en utilisant des techniques telles que le GridSearch

  - ✓ Intégrer le modèle final en utilisant une API RESTful permettant aux images d'être envoyées au serveur pour classification en temps réel, et d'afficher les résultats

- L'application sera développée en utilisant :

  - ✓ Le framework Django (Python), éventuellement MERN
  - ✓ Le package scikit-learn pour entraîner et déployer le modèle de classification (éventuellement tensorflow)
  - ✓ MongoDB-Atlas et pymongo pour stocker :

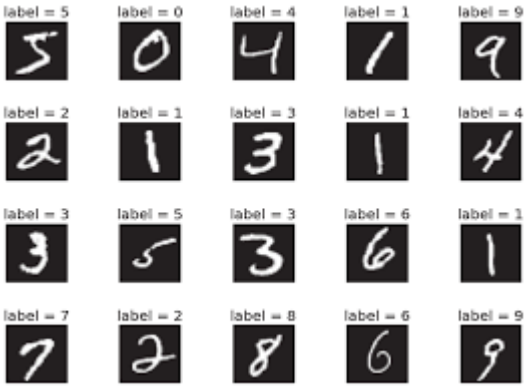
    - Le jeu de données MNIST (format initial : CSV) dans 2 collections : une 1ère pour le ***train*** et une 2ème collection pour le ***test*** ;
    - Dans une 3è collection : on stockera les images dessinées par les users Et les résultats des prédictions corresp. à ces images

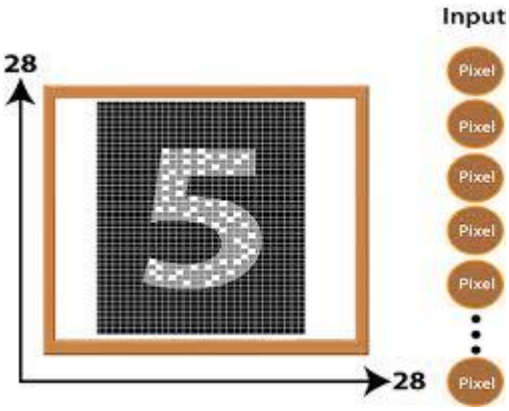
  - ✓ React JS (ou autre framework) pour créer l'interface utilisateur interactive

Data

- Le jeu de données MNIST est un ensemble d'images en noir et blanc de chiffres manuscrits de 0 à 9, où chaque image est de taille 28x28 pixels. Ces images ont déjà été traitées pour obtenir le format **CSV**. Pour les curieux, vous pouvez consulter ce lien : https://pjreddie.com/projects/mnist-in-csv/. Il s'agit alors d'images bitmap transformées en vecteur de données.

- L'objectif est de reconnaître ces caractères manuscrits : chiffres de 0 à 9 (votre ***target*** "***y***" possède alors K = 10 modalités). Ce problème est le ***Hello World*** de la classification d'image connu sous le nom d'***Optical Recognition of Handwritten Digits***



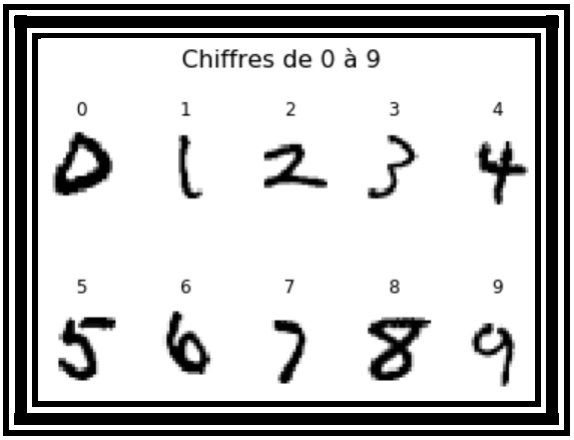


Questions guidées :

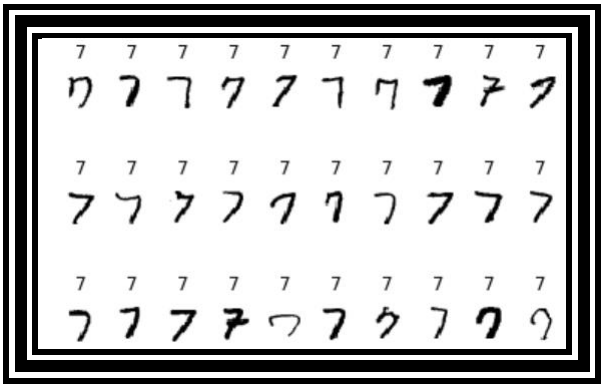
1) **Exploration du jeu de données MNIST** : Charger et visualiser les images d'entraînement et de test du jeu de données MNIST dans un notebook ?

    a. Afficher une image quelconque depuis le ***train*** en indiquant comme titre de l'image son label (càd le target associé) ?

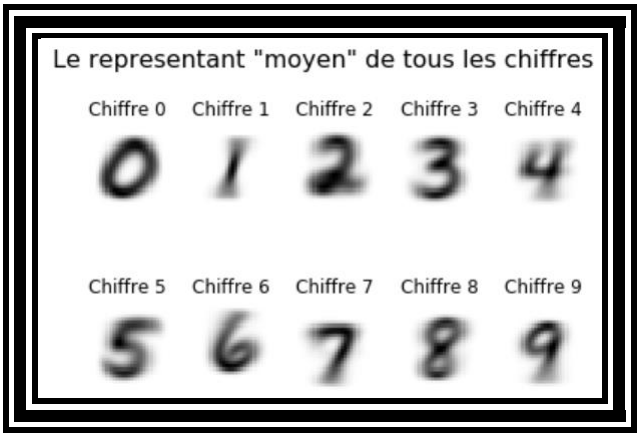
    b. Afficher dans une même figure les chiffres de 0 à 9 ?



c. Pour visualiser les différentes façons d'écrire un **7,** afficher les 9 1ères images qui correspondent au chiffre 7 en les plaçant dans une même figure ?



d. Afficher le représentant "moyen" de chaque chiffre ?



2) **Intégration de MongoDB**

3) **Entraînement et sérialisation du modèle :** Créer, entraîner, évaluer et optimiser plusieurs modèles de classification en utilisant le dataset MNIST à partir de MongoDB (et non à partir du CSV) et stocker le modèle le plus performant sous format pickle ou h5 (…)

4) **Développement de l'interface utilisateur :** Utiliser ReactJS pour développer une interface utilisateur interactive pour la WebApp, permettant aux utilisateurs de dessiner des chiffres manuscrits sur la page d'accueil, d'envoyer les images au serveur pour classification, et d'afficher les résultats de prédiction

5) **Intégration du modèle dans l'application web :** Intégrer le modèle de classification entraîné dans l'application web en utilisant une API RESTful, permettant aux images dessinées par les utilisateurs d'être envoyées au serveur pour classification en temps réel, et d'afficher les résultats sur l'interface utilisateur.

6) **Déploiement de l'application :** Déployer la WebApp sur un serveur en utilisant des outils de déploiement appropriés, et tester son fonctionnement en ligne.

7) Afficher un extrait des bonnes prédictions dans une figure ? En d'autres termes, vous affichez l'image corresp. dans le test et la prédiction trouvée par le classifieur ?

8) Afficher un extrait des prédictions erronées ? Sur quel chiffre, le classifieur se trompe le plus ? Essayer de comprendre pourquoi ?

Livrables attendus (15 pts)

- **Un repo Github** où on trouve :
  - Un fichier README où vous expliquez :
    - *Comment lancer les scripts*, …
    - *Mentionnez le lien de votre trello avec la répartition des tâches à travers les membres de votre groupe* (à faire/en cours/fait)
    - …
    - *Lien vers le support de présentation* (par exemple google slides)
  - Un notebook Python (non cleané pour comprendre votre démarche)
    - Les problèmes rencontrés sur le jeu de données
    - Comment vous avez nettoyé les données
    - Votre modélisation (du preprocessing à la prédiction)
  - Le code générant la WebApp et permettant de déployer le modèle sous forme d'API
  - L'URL de la WebApp mise en ligne et répondant au cahier des charges précisé ci-dessus.
  - Un support de présentation (environ 10 slides)

Présentation du travail (5 pts)

Votre présentation pourra prendre cette forme (à titre indicatif) :

5 min.	Rappel de la problématique, présentation du jeu de données, de l'exploration, du cleaning, …
15 min	Explication de l'approche pour mettre en place votre solution (méthodo, code, interprétation des résultats, modèle final sélectionné ainsi que des performances et améliorations effectuées, … ?) ET Démo
10 min	Séance de questions-réponses