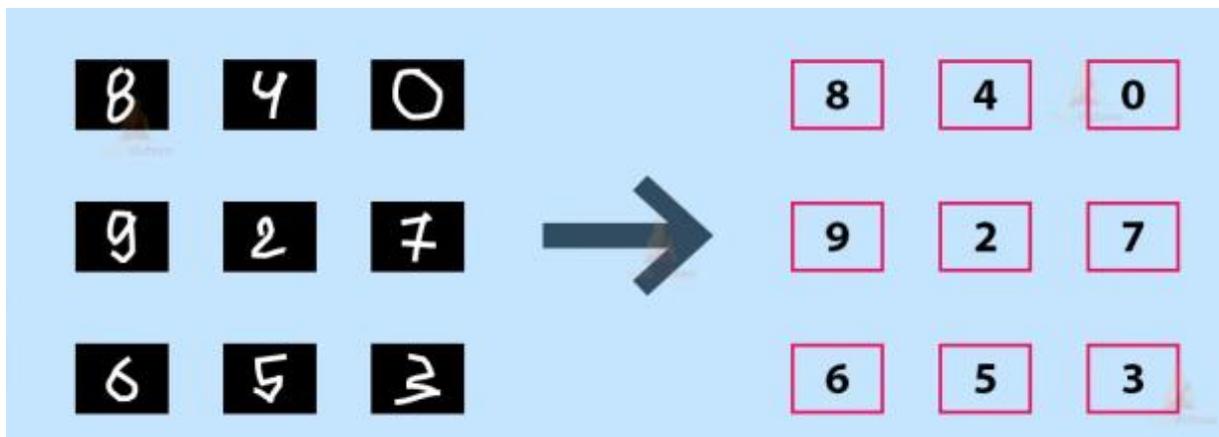


Mener un projet informatique

Description générale du projet

L'intelligence artificielle (IA) est définie comme étant la recherche de moyens susceptibles de doter les systèmes informatiques de capacités intellectuelles comparables à celles des êtres humains, ou en tout cas mimant ces capacités. Dans ce cadre, plusieurs travaux de recherche ont été réalisés durant la dernière décennie. Ces travaux ont connu une grande avancée avec l'arrivée de l'apprentissage profond, en l'occurrence, les réseaux de neurones convolutifs (CNN). Ces réseaux sont utilisés dans divers applications comme la reconnaissance d'images, l'analyse de textes, la vidéo-surveillance, l'aide au diagnostic...

Dans le cadre de ce projet, vous allez créer un réseau de neurones convolutif capable de reconnaître les chiffres écrits à la main. Pour cela et à partir d'images contenant des chiffres écrits à la main, vous allez entraîner une IA capable de dire quel chiffre est présent dans une image donnée.



Livrables

Projet individuel de développement informatique avec des livrables finaux :

- Les programmes réalisés (regroupe tous les codes que vous auriez réalisés),
- Un rapport de quelques pages retraçant les différentes étapes du projet et des résultats obtenus,

Réalisation du projet

La réalisation de ce projet se fera sur plusieurs étapes. En effet, le développement d'une IA basée sur l'apprentissage machine (machine learning) repose sur 4 notions essentielles qui sont communes à tous les projets de machine learning. Ces 4 notions étant :

1. **Le dataset** : En machine learning, tout démarre d'une base de données qui contient les données relatives à la problématique abordée. Dans l'apprentissage supervisé (notre cas), la base de données contient les observations (images dans notre cas) associées à leurs labels (chiffres dans notre cas).
2. **Le modèle et ses paramètres** : à partir de cette base de données, un modèle est créé, ce modèle est une fonction mathématique associant les observations aux labels. Les coefficients de cette fonction sont les paramètres du modèle.
3. **La fonction de coût** : un paramètre important permettant l'apprentissage du modèle. En effet, lorsque le modèle est testé sur la base de données, il donne des erreurs. L'ensemble de ces erreurs est appelé le coût qui est calculé en utilisant la fonction de coût. Ce coût est, ensuite, utilisé pour la mise à jour des paramètres du modèle.
4. **L'algorithme d'apprentissage** : c'est l'idée principale du machine learning, c'est de laisser la machine trouver quels sont les paramètres du modèle qui minimisent le coût sur la base de données.

Partie 1 : Le dataset

La première étape de la réalisation de ce projet consistera dans l'analyse et l'implémentation des différentes fonctions permettant la gestion de la base de données. Dans ce projet, vous allez utiliser la base de données MNIST contenant des images de 28x28 pixels de chiffres tracés à la main.



1. Télécharger le fichier [dataset.zip](#), placer le dans un répertoire R508 dans votre répertoire utilisateur et décompresser le à l'aide de la commande `unzip dataset.zip`
2. Ecrire une description résumant votre compréhension du projet et les différentes étapes que vous comptez entreprendre pour sa réalisation (fonctions, temps de réalisation, ressources ...).
3. Etudiez les données de ce répertoire et faites une petite description (nombre d'images, répartition, fichiers d'annotation).
4. Ecrire une fonction python `stat_dataset(nom_fichier)` qui fait les statistiques de votre base de données. La fonction prendra en entrée le fichier csv `nom_fichier` et renverra un dictionnaire

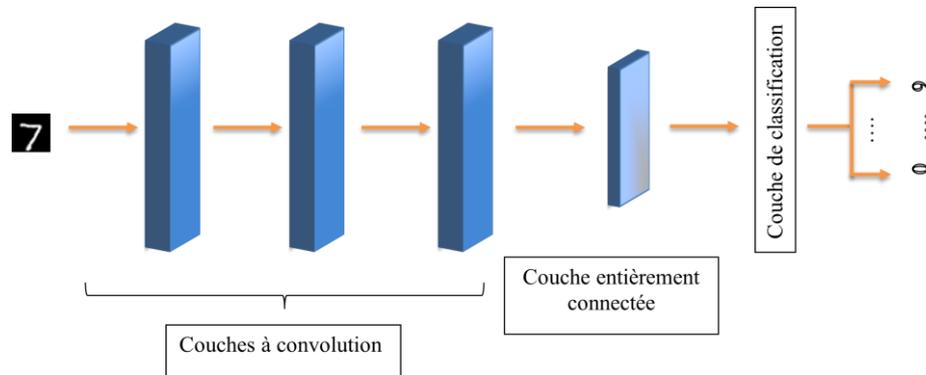
dont les clés représentent les classes de la base et les valeurs représentent le nombre d'éléments de chacune des classes. Utiliser cette fonction pour avoir les statistiques sur les données d'apprentissage et les données de test de votre base.

5. Etudier les fonctions `cv2.imread()` et `cv2.imshow()` et `cv2.imwrite()` et décrire leurs fonctionnements.
6. Ecrire une fonction python `affichage_exemple(nom_fichier, nom_repertoire, nbr)` qui permet d'afficher des exemples de la base de données, ainsi que les labels correspondants. La fonction prendra en entrée le fichier csv `nom_fichier` et le répertoire d'images `nom_repertoire`, ainsi que le nombre aléatoire d'exemples à afficher `nbr` et renverra un dictionnaire dont les clés sont les noms d'images sélectionnés aléatoirement et les valeurs sont des listes de deux éléments dont le premier est l'emplacement de l'image et le second le label correspondant. La fonction renverra également une image affichant l'ensemble des exemples sélectionnés avec les labels correspondants.
7. Ecrire une fonction python `re_organisation(nom_fichier1, nom_fichier2, nom_repertoire1, nom_repertoire2)` permettant de réorganiser la base de données. Cette fonction doit prendre en entrée les noms de fichiers d'annotation `train_data.csv` et `test_data.csv` et les noms des répertoires `train` et `test` de la base initiale et renverra une nouvelle base organisée comme indiqué dans le schéma. Chaque répertoire doit contenir toutes les images d'apprentissage/test correspondant au nom du répertoire.

```
dataset
  |----train
  |---0
  |---1
  |---2
  |---3
  |---4
  |---5
  |---6
  |---7
  |---8
  |---9
  |----test
  |---0
  |---1
  |---2
  |---3
  |---4
  |---5
  |---6
  |---7
  |---8
  |---9
```

Partie 2 : Le modèle de classification

La deuxième étape de réalisation consistera en la construction d'un modèle de réseau de neurones convolutif (CNN), son entraînement et son évaluation dans le cadre de la reconnaissance de chiffres écrits à la main. Dans ce projet, vous allez utiliser un modèle composé de 3 couches de convolution et d'une couche fully-connected.

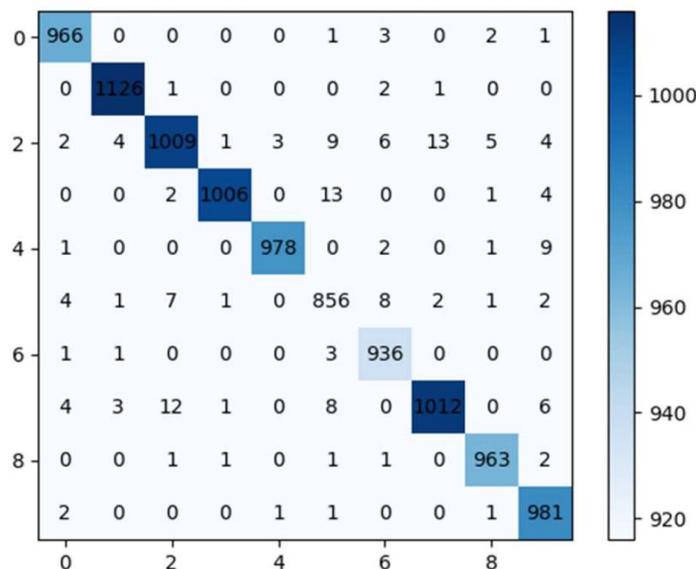


1. Faites une petite recherche sur les modèles de réseau de neurones convolutif (CNN) et expliquer le fonctionnement de ces derniers.
2. Etudier les chargeurs de données (data loader), la fonction `image_dataset_from_directory()` et ses paramètres et décrire leur fonctionnement et leur utilité.
3. Ecrire une fonction python `chargement_donnees(nom_repertoire_train, image_size, batch_size)` qui permet de charger les données qui seront utilisées dans l'entraînement du modèle. La fonction prendra en entrée le répertoire d'images `nom_repertoire_train`, la taille des images `image_size` et `batch_size` et renverra deux chargeurs de données (data loader) `train_gen` et `val_gen` pour les données d'apprentissage qui vont représenter 80% et les données de validation qui vont représenter 20% de l'ensemble des données d'entraînement, respectivement.
4. Etudier la création de modèles séquentiel avec la fonction `Sequential()` et l'ajout des différentes couches à ces modèle grâce à la fonction `model.add(layer, activation, ...)` et décrire son fonctionnement.
5. Ecrire une fonction python `creation_modele(num_classes, image_size)` qui permet de créer un modèle CNN composé d'une couche d'entrée `InputLayer`, de 3 couches de convolution `Conv2D`, d'une couche d'aplatissement `Flatten` et d'une couche fully-connected `Dense` (expliquer le fonctionnement de chacune de ces couches). La fonction prendra en entrée le nombre de classes de sorties de votre modèle `num_classes` et la taille des images `image_size` et renverra le modèle CNN.
6. Compiler votre modèle à l'aide de la fonction `model.compile()` et entraîner le à l'aide de la fonction `model.fit()`.

Partie 3 : évaluation de classification et détection

La troisième étape de réalisation consistera en l'évaluation de votre modèle de classification sur vos données de test et l'utilisation de ce modèle pour la détection de chiffres.

1. Faites une petite recherche sur l'évaluation d'un modèle de classification et expliquer la procédure dans les grandes lignes.
2. Ecrire une fonction python `eval_image(model, chemin_img)` qui permet d'évaluer votre modèle de classification entraîné précédemment sur une image donnée. La fonction prendra en entrée le modèle entraîné `model` et le chemin à votre image `chemin_img` et renverra une liste de 3 éléments : le premier étant la vraie classe de l'image, le deuxième étant la classe estimée par le modèle et le troisième étant la probabilité de cette estimation.
3. Ecrire une fonction python `eval_base(model, nom_repertoire_test)` qui permet d'évaluer votre modèle de classification entraîné précédemment sur l'ensemble des données de la base de test. La fonction prendra en entrée le modèle entraîné `model` et le répertoire d'images `nom_repertoire_test` et renverra un dictionnaire dont les clés sont les vraies classes des images et les valeurs sont des listes de 10 éléments représentant les classifications des images de cette classe par le modèle : chaque élément cette liste représentera le nombre d'images classées dans les classes 0, 1, 2...9 par le modèle respectivement.
4. Faites une petite recherche sur la construction d'une matrice de confusion pour une classification et expliquer cette dernière.
5. Modifier la fonction la question 3 pour afficher la matrice de confusion qui correspond à votre classification.



6. Faites une petite recherche sur la détection d'objets avec l'approche de fenêtre glissante (sliding window approach for objects detection), expliquer cette approche.