

Composantes du Rapport de Projet

1. Définition de la Tâche

Description de la tâche: Expliquez ce que fait votre système (entrée et sortie).

Le système qui sera développé devra prendre en entrée une police de caractère ainsi qu'une image textuel ayant cette même police de caractère. Après traitement, Elle retournera une chaîne de caractère correspondant au texte donner en entrée sous forme imagée. Le modèle doit constamment apprendre de nouvelles polices de caractère. Elle doit s'adapter aux nouvelles polices de caractères utilisées et nommer par l'utilisateur.

Délimitation: Assurez-vous que la tâche n'est ni trop large ni trop étroite.

Pertinence IA: Justifiez l'utilisation de l'intelligence artificielle pour cette tâche.

Dans les années précédentes, un programme algorithmique était utilisé pour reconnaître les écritures et les différentes polices de caractère. Cependant, ces programmes manquaient de fiabilité. il était parfois nécessaire d'écrire un nouveau programme et un nouvel algorithme pour des police de caractère plus récente. ces outils informatiques manquait d'adaptabilité. L'utilisation de l'intelligence artificielle basée sur un modèle s'avère une solution efficace, puisqu'elle permet une fonctionnalité d'apprentissage qui lui permet d'intégrer des nouvelles polices de caractère. Ainsi les modèles peuvent être ajusté lorsque de nouvelles police de caractère font leur apparition.

2. Brève Revue de la Littérature

Dans la littérature, nous pouvons citer deux. Implémentations du OCR :

- La première consistait à simplifier la documentation des transactions dans le commerce électronique . Le OCR servait à numériser les transactions sous format papier. Ainsi, il était plus facile de parcourir l'historique des transactions. Le modèle atteignait un score F1 de 0,7703 lors de ses tests
- Le deuxième utilisait le OCR afin de lire des caractères sur des interfaces graphiques comme des boutons, des textes et des champs de texte. Ce modèle utilise un réseau de neurones dans sa méthode d'apprentissage. Il a été implémenter sur deux architectures. Soit l'architecture en cascade et l'architecture "bout-en-bout". Ce modèle a eu un score F1 de 94%

3. Matériel et Méthodes

infrastructure

Lors de ce projet, nous utiliserons le langage de programmation Python afin d'entraîner le modèle pour chaque police de caractère entrée par l'utilisateur. Ce langage de programmation est très utilisé pour ce qui est de l'intelligence artificielle de l'analyse de données et du machine Learning. Sa syntaxe est aussi très simpliste et très simple d'utilisation.

Nous utiliserons aussi un modèle de base conçu pour du OCR. Le modèle Tesseract est un modèle d'OCR qui est facile à agrémenter. Il est ainsi possible d'apprendre au modèle à mieux reconnaître certaines polices d'écriture. Nous utilisons ce modèle afin de comparer les solutions apportées par notre agrémentation. Nous utiliserons aussi des documents produits manuellement afin de tester le modèle de base et le modèle avec agrémentation.

méthode

Afin de parvenir à l'objectif mentionné plutôt, nous utiliserons les entrées de l'utilisateur afin de générer aléatoirement du contenu textuel imagée et structuré dans la police de caractère entrée par l'utilisateur. Les chaînes de caractère utilisées pour générer ces images seront aussi conservées. Ensuite, le modèle sera entraîné sur les images de manière supervisée. Les couples (images, chaînes de caractère) formeront dataset utilisé pour entraîner le modèle. Ensuite, le modèle sera testé sur le texte écrit manuellement par l'utilisateur.

De cette manière, nous assurons que le modèle s'entraîne de manière supervisée en requérant un minimum d'interaction avec l'utilisateur.

évaluation

Le but de ce projet est de comparer le modèle avec notre agrémentation avec le modèle de base. Ainsi, il sera possible de mesurer le gain ou la perte de précision du modèle. Nous comparons ainsi les scores F1, du modèle de base et du modèle avec notre agrémentation. Un modèle avec un score F1 moins élevé et moins précis et vice-versa. Il suffit donc de trouver le modèle avec le score F1 le plus élevé.

Pour l'évaluation, il sera donc nécessaire d'écrire un programme Python simple qui mesurera la précision et le rappel, dépendant eux-mêmes de quatre données

- faux positif (FP)
- faux négatif (FN)
- vrai positif (TP)
- vrai négatif (TN)

Nous avons le choix de calculer ces données parmi différentes grandeurs textuelles. Nous pouvons les calculer à partir :

- Des caractères correctement reconnus
- Des mots correctement reconnus
- Des lignes correctement reconnues
- Des phrases correctement reconnues

4. Résultats

- **Présentation des résultats:** Affichez les résultats obtenus sous forme de tableaux ou graphiques.
- **Analyse:** Interprétez les résultats et discutez de leur signification.

5. Conclusion