

EFREI PS2

Projet Transverse – BrainPower

Adrien Michel
Louis Potterie
Benjamin Saffar
Adin Hrelja
Ahmed Doghri

SOMMAIRE

Présentation du projet

I- La Segmentation

- Mise en niveau de gris
- Seuillage
- Délimitation de contours pour la détection
- Tri des images
- formatage

II- La Reconnaissance des caractères

- Le réseau de neurone
- L'apprentissage

Conclusion

Le projet, dans sa finalité, consiste à créer une application et un logiciel qui permettent respectivement de prendre en photo et de scanner un document au format manuscrit comme une prise de note et de le transcrire en format numérique en conservant la mise en page. La quantité de travail étant trop élevée pour un semestre, nous avons, si l'état du projet le permet, l'ambition de le prolonger en 3ème année.

Nous ciblons les milieux scolaire et professionnels qui utilisent encore l'écriture manuscrite. Des élèves ont souvent des prises de note brouillonnes, qui méritent une mise en page propre pour faciliter l'apprentissage. Une formule mathématique, par exemple, n'est ni simple ni rapide à écrire dans un logiciel de traitement de texte. Par ailleurs, les professeurs trouveront leur intérêt dans l'élaboration de cours s'ils ne sont pas à l'aise avec un clavier d'ordinateur ou s'il serait plus rapide de les écrire à la main. L'application rendrait aussi plus simple la correction de certaines copies (par exemple, des programmes écrits à la main en les faisant tourner sur ordinateur directement, ou tout simplement des textes à trous). On pourrait encore prolonger ces applications à la copie de vieux documents manuscrits (pour les musées ou archives)

L'application fonctionnerait par divers algorithmes de reconnaissance d'image à base de neurones artificiels. Le principe est de reconnaître les types d'objets manuscrits présents sur une page (texte brut, formules, schémas, tableaux...) pris en photo, puis de les traiter séparément. Pour chaque type, le programme rescindera les éléments successivement (texte brut → mot → lettre, formule → opérateur/fonction → chiffres/lettres... par exemple). Pour cela nous avons entraîné nos algorithmes sur une base d'exemples créés par nous-même. Nous utilisons le langage python qui nous propose divers outils open source additionnels simplifiant le traitement d'image comme Open CV et Numpy pour le calcul mathématique. Par la diversité des fonctionnalités et leur difficulté, nous nous sommes concentrés sur une d'entre elles : la reconnaissance de phrases écrites en majuscule. Celle-ci nécessite la détection de lettres majuscules. C'est l'objectif principal à atteindre pour ce semestre.

I – La Segmentation

Une fois le document pris en photo ou scanné, il faut séparer tous les caractères et les mettre dans l'ordre pour pouvoir ensuite les reconnaître. Pour cela nous faisons une segmentation.

Tout d'abord nous mettons l'image à niveau de gris. Le programme analyse tous les pixels et en fonction de leur couleur (RVB). Il va ainsi pouvoir remplacer ces trois valeurs par une seule représentant la luminosité. Cela permet d'avoir une image qui varie du blanc au noir avec différents niveaux de gris.

Ensuite, en utilisant une certaine valeur correspondant à un niveau de gris, le programme va pouvoir faire un seuillage. En effet, tout pixel ayant un niveau supérieur au seuil va être mis en blanc et ceux inférieur en noir.

Cela permet ensuite de délimiter les contours des formes. Cette méthode consiste à détecter les points de changement brutal d'intensité des pixels de l'image. On peut alors procéder à une segmentation horizontale pour détecter toutes lignes manuscrites, puis une segmentation verticale pour obtenir les images de chaque mot écrit. Seulement elles ne sont pas stockées dans l'ordre de la phrase. En effet l'ordinateur les classe tout seul en fonction du

premier pixel détecté en ordonnée. Pour régler ce problème nous n'utilisons les lignes créées lors de la segmentation horizontale. Le programme « prend note » des deux ordonnées correspondant au haut et bas de chaque phrase en commençant par la première. Une fois qu'il est focalisé sur la première, il utilise les abscisses de manière croissante pour classer les images des mots. Nous avons donc finalement un fichier où se trouve les images des mots dans l'ordre.

Ensuite, nous répétons toute ces opérations sur toutes images de mots pour avoir des images de chaque caractère dans l'ordre.

Enfin, le programme prépare ces images de lettres à la reconnaissance en les mettant en niveau de gris, en les redimensionnant en 28×28 et en les plaçant dans l'ordre dans une matrice 3D pour les préparer à la reconnaissance. La première dimension correspond aux lignes, la deuxième aux mots et la troisième aux caractères.

II – La Reconnaissance des caractères

Une fois que nous avons les images de lettre, il faut pouvoir les reconnaître. Pour cela nous utilisons un réseau de neurones que nous avons entraîné nous-même.

Le réseau de neurones est composé de 3 couches, deux cachées de 10 neurones chacune et une de sortie.

L'entrée est un vecteur de taille 784 (nombre de pixels de l'image à évaluer) et la sortie un vecteur de taille 26 (pour les 26 lettres de l'alphabet, 0 pour le 'A' jusqu'à 25 pour le 'Z').

L'index de la valeur la plus satisfaisante de la sortie détermine le résultat (la lettre reconnue) associé à cet index.

Maintenant il faut l'entraîner. Pour cela, nous avons créé des images de lettres à partir de lettres écrites à la main et en majuscules. Pour chaque image, on lui attribue la lettre qui lui correspond. On entraîne le réseau avec l'image et le résultat attendu par rétropropagation de gradient. Le but est de minimiser l'erreur entre le résultat pratique et théorique en modifiant les poids et les biais du réseau selon l'erreur qu'ils génèrent.

Maintenant, nous pouvons transmettre les images de lettre de la phrase écrite. Le réseau va les analyser, trouver les lettres correspondantes et écrire dans un fichier le texte manuscrit.

III – L'avenir du projet

Ce projet nécessite encore beaucoup d'amélioration pour se rapprocher de l'idée de départ. C'est donc pour cela que nous espérons le poursuivre en L3.

Par le manque de données d'entraînement, le réseau de neurones ne peut pas assez apprendre et fournit donc des résultats peu concluants. C'est pour quoi les algorithmes de segmentation ne sont pas encore liés au réseau.

Les principales évolutions à court terme seraient un dictionnaire qui permettra en même temps que le réseau reconnaît les caractères de pouvoir faire une recherche afin de trouver le mot. Cela permet de renforcer le logiciel dans le but de trouver le bon mot. On va aussi entraîner beaucoup plus le réseau afin avec des écritures de nouvelles personnes, on va lui faire apprendre de nouveaux caractères et des écritures en minuscules. D'autres problèmes

sont à résoudre au niveau du traitement de l'image lors de la segmentation, la difficulté de trouver un seuillage universel impose d'écrire lisiblement sur un fond clair.

Mais la segmentation aussi est à améliorer. Il va falloir qu'elle puisse segmenter des lettres minuscules qui seront donc attachées (tout dépend de qui écrit). Et enfin nous feront écrire le texte non pas dans un .txt mais un word ce qui nous permettra aussi de pouvoir mettre en place une mise en page.

Ce projet était très intéressant. Il nous a permis de mieux nous rendre compte des types de projet que nous serons amenés à faire dans le futur, à l'école comme en entreprise. Ce projet est un vrai défi car il nous a demandé d'apprendre un nouveau langage et qui plus est nous nous sommes dans la conception dans la conception d'un réseau de neurone. C'est un projet qui est très long et difficile à faire en plus des cours. Mais nous avons grand espoir de le poursuivre en L3 et même après s'il le faut afin d'arriver à un logiciel très performant et pourquoi pas espérer le commercialiser.