Rapport de Projet de Spécialité

GeneTeX : développement d'un système de numérisation de notes de cours

BOUKSARA Mehdi
MERLE Théo
THALGOTT Marceau

I. Introduction

Le problème de la reconnaissance d'écriture est un problème qui est encore aujourd'hui le sujet de nombreuses recherches, et demeure un problème ouvert. Cependant, bien que les algorithmes de détection et de reconnaissance des caractères s'améliorent sans cesse, trouver et implémenter un algorithme de reconnaissance d'écriture manuscrite d'un niveau équivalent à celui d'un cerveau humain est à l'heure actuelle un objectif loin d'être atteint.

Retranscrire le contenu d'un document écrit à la main sur son ordinateur est une tâche qui peut être fastidieuse : l'automatisation d'une telle tâche serait confortable pour de nombreuses personnes, en particulier des étudiants et des chercheurs. L'objectif de ce projet est de concevoir et de développer un logiciel capable de générer un document au format choisi (notamment LaTeX) à partir de pages de notes numérisées, ou du moins de jeter les fondations de ce qui sera peut-être un jour ce logiciel.

II. Objectifs du projet

Il est évident que les objectifs fixés dans le cadre du Projet de Spécialité ne sont qu'une partie infime du travail nécessaire pour créer un tel logiciel. Ainsi, nous avons déterminé un point essentiel, qui a conditionné son déroulement du début à la fin : le logiciel développé doit être capable de s'adapter le plus facilement et le plus rapidement possible à l'évolution des travaux dans le domaine de la reconnaissance d'écriture.

Cet objectif pose une contrainte forte : la conception du logiciel doit être pensée de manière à permettre des changements à n'importe quel niveau, sans nécessiter une adaptation du reste du logiciel. Bien entendu, au-delà de l'aspect conceptuel, des résultats sont attendus au niveau de la reconnaissance d'écriture en elle-même : bien qu'à terme, GeneTeX est censé pouvoir retranscrire les méta-informations (mise en page, position, style) liées au texte reconnu, ou encore des symboles mathématiques

complexes, nous nous limiterons ici à une première version de la reconnaissance d'un paragraphe de texte et sa transcription en format LaTeX, l'objectif principal étant de montrer qu'un tel logiciel peut fonctionner et être modifié sans difficulté.

III. Principe de fonctionnement

Le fonctionnement de GeneTeX repose sur un découpage en plusieurs étapes. Avant toute chose, l'image est pré-traitée pour faciliter l'extraction des caractères, puis ces caractères sont découpés en blocs, lignes puis isolés les uns des autres. Ils sont alors analysés individuellement pour être reconnus, puis intégrés à un document LaTeX qui constitue la sortie du logiciel.

3.1 Pré-traitement

Un document manuscrit est, dans la quasi-totalité des cas, susceptible de présenter divers défauts, qu'ils soient dus à la qualité de la numérisation du document ou présents sur la feuille initialement (bruit, pixels non pertinents, feuille à carreaux, taches, etc.). Afin d'éviter que ces défauts ne perturbent le processus, il est donc nécessaire de pré-traiter l'image pour en éliminer un maximum et ainsi augmenter les chances de réussite de la reconnaissance.

3.2 Découpage

Afin de reconnaître les caractères individuellement et ainsi éviter une reconnaissance mot par mot nécessitant un dictionnaire, il faut pouvoir les isoler les uns des autres tout en conservant un maximum d'informations sur eux (position, taille, couleur, mise en forme...). Les blocs de texte sont isolés, puis les lignes d'un bloc et enfin les caractères d'une ligne. Les morceaux de l'image correspondant ainsi que diverses caractéristiques sont alors stockés dans une structure adéquate.

3.3 Reconnaissance

Une fois que les caractères ont été isolés, l'étape suivante, qui constitue le cœur du logiciel, consiste à reconnaître les caractères les uns après les autres. La méthode de reconnaissance n'est pas imposée, l'objectif étant simplement de pouvoir déterminer avec une certaine précision quel est le caractère représenté par le morceau d'image analysé. Comme ce morceau d'image peut ne correspondre à aucun caractère connu (s'il s'agit d'une tache ayant passé l'étape de prétraitement, ou d'un morceau de caractère mal découpé), il doit être possible de simplement rejeter un caractère et de l'ignorer s'il ne ressemble à rien de connu. Une structure est alors générée, contenant les caractères reconnus.

3.4 Génération de code

La dernière étape du processus se résume à parcourir la structure des caractères reconnus, et à les insérer avec la mise en forme nécessaire dans un document LaTeX, après avoir généré un en-tête. Cette mise en forme peut consister, sans toutefois s'y limiter, en l'échappement de certains caractères, ou l'utilisation de « balises » spéciales pour les formules mathématiques.

IV. Travail réalisé

Comme expliqué précédemment, notre objectif dans le cadre du Projet de Spécialité était de jeter les fondations du logiciel, en réalisant deux tâches précises :

- Concevoir l'architecture de la manière la plus modulaire et accessible possible
- Développer des fonctionnalités basiques permettant de tester notre système dans des conditions favorables.

Autrement dit, le but n'est pas de produire un système de Reconnaissance Optique de Caractères (OCR) à la pointe de la technologie, mais plutôt de créer les bases nécessaires pour développer un logiciel capable de suivre facilement l'avancée de la recherche à ce sujet, tout en proposant une première implémentation basique d'un OCR.

Pour ce faire, nous nous sommes basés sur des travaux de recherche. Ces travaux ne sont pas très récents, mais le contenu scientifique est à notre portée, et constitue une base suffisante pour obtenir les résultats attendus.

4.1 Prétraitement

Nous avons développé un module de prétraitement réalisant le strict nécessaire pour permettre une transcription d'une fiabilité minimale. Ce module permet de « binariser » une image, c'est-à-dire la transformer en un tableau de booléens, dont les cases correspondent chacune à un pixel de l'image, indiquant si le pixel en question fait parti d'un bout de texte ou de l'arrière-plan, et dont la valeur dépend de ses niveaux RGB. Un seuil permet de déterminer si un pixel est suffisamment coloré pour apparaître dans l'image binarisée.

Cette technique est suffisante pour éliminer une grande partie des défauts dus à la qualité de la numérisation, mais ne permet pas d'éliminer certains autres défauts tels que des taches sur le document.

Nous avions également prévu de développer une technique de redressement d'image, pour gérer le cas d'un document incliné lors de la numérisation. L'inclinaison pouvant perturber fortement le découpage, cette technique permettrait de déterminer l'angle d'inclinaison de l'image, et il suffirait ensuite d'appliquer la rotation inverse pour

obtenir une image dans laquelle le texte est horizontal et donc bien plus facile à découper.

Cependant, nous avons manqué de temps pour finaliser le développement de cette fonctionnalité. Les fonctions secondaires nécessaires pour cela ont cependant été écrites et documentées.

Fig. 1: Exemple de caractère

Fig. 2: Exemple de caractère binarisé

4.2 Découpage

Dans le cadre de ce projet, nous n'avons considéré qu'un seul bloc de texte. Ainsi, nous nous sommes concentrés sur la détection de lignes, et de caractères dans une ligne.

En ce qui concerne la détection des lignes, on considère de façon très basique qu'une ligne est constituée d'un ensemble de lignes de pixels consécutives dans l'image comportant au moins K pixels colorés, K étant un seuil devant être fixé en fonction de la résolution et de la taille de l'image.

Nous avons distingué l'étape de découpage en deux parties, l'une se comportant comme un complément de l'autre.

La plus simple, que nous appellerons segmentation primaire, permet de détecter des caractères séparés, ne présentant pas de chevauchement. Cette situation constitue un cas extrêmement favorable, mais le coût extrêmement réduit d'une telle méthode de segmentation justifie son utilisation. La détection des caractères se base sur la séparation en colonnes : si on détecte au moins une colonne de pixels vide dans une ligne, on estime que le caractère en cours de détection prend fin en largeur. Si on détecte un nombre suffisant de colonnes vides consécutives, lié à la largeur des caractères déjà rencontrés, alors on en déduira qu'il s'agit d'un espace.

Cependant, cette technique de découpage est parfois insuffisante, et ne fonctionne pas si les caractères se chevauchent, ou si des caractères différents sont connectés. C'est le travail de la segmentation secondaire de résoudre ces cas plus gênants.

La segmentation secondaire tente de résoudre le problème du chevauchement de caractères en recherchant simplement les ensembles de pixels connectés entre eux. C'est une méthode simple, mais qui pose des problèmes lorsqu'un caractère est composé de plusieurs parties distinctes (par exemple, la lettre « i », ou le point-virgule). Il est alors nécessaire de réunir ces parties en utilisant le recouvrement horizontal des éléments séparés : si deux éléments se recouvrent suffisamment, ils sont réunis.

La résolution du problème des caractères connectés pose encore plus de

problèmes, puisqu'il est impossible de savoir avec certitude à quel endroit se termine le premier caractère. Nous avons développé une heuristique se basant sur la largeur moyenne d'un caractère, qui découpe le caractère à l'endroit où il présente la plus faible épaisseur, dans une zone donnée. Cette technique fonctionne dans certains cas, mais nous n'avons pas réussi à corriger certains bogues.

Fig 3: Exemple de chevauchement

Fig. 4: Exemple de caractères connectés

4.3 Reconnaissance

Pour implémenter la reconnaissance d'un caractère, nous avons utilisé un réseau de neurones simple, connu sous le nom de perceptron multicouches. Bien que les techniques de reconnaissance les plus récentes utilisent également la technologie des réseaux de neurones, nous nous sommes contentés d'une version basique, permettant tout de même d'obtenir quelques résultats.

Le réseau de neurones est associé à un module d'apprentissage(sous la forme d'un programme secondaire) permettant au réseau de s'adapter aux éléments qu'il doit reconnaître. L'idée consiste à injecter les données d'apprentissage dans le réseau les unes après les autres, et d'adapter les coefficients internes du réseau en fonction de la différence entre la réponse donnée et la réponse attendue.

Le problème principal de cette étape est de réunir des données d'apprentissage. Afin de nous faciliter la tâche, nous avons développé un module d'échantillonnage (lui aussi sous la forme d'un programme secondaire) permettant de générer des données d'apprentissage à partir d'une image, connaissant les caractères qu'elle contient ainsi que l'ordre dans lequel ils apparaissent. Cependant, le nombre d'échantillons nécessaires pour constituer une base minimal implique de passer énormément de temps à réunir ces échantillons.

Comme nous n'avons pas trouvé de base de données publique de caractères pour éviter cette phase, nous avons décidé de faire nos tests en utilisant comme base d'apprentissage les caractères exacts à reconnaître, ce qui permet tout de même d'éprouver le fonctionnement du réseau de neurones et de l'apprentissage.

4.4 Génération

Le module de génération du fichier LaTeX est très simple : il consiste en un générateur d'en-tête, d'un parcours d'arbre pour récupérer les caractères reconnus et les insérer dans le fichier, et d'un générateur de pied de page.

4.5 Documentation

Afin de rendre le projet le plus accessible possible à d'éventuels repreneurs du projet, nous avons porté une attention particulière à la documentation du projet en plus des efforts de conception réalisés.

Ainsi, nous avons produit, en plus des sources :

- La Javadoc liée au code source
- La Javadoc liée au code des tests
- Les diagrammes de classe, d'états-transitions et la décomposition architecturale du système
- Une version minimaliste du manuel utilisateur

V. Bilan du projet

L'objectif pédagogique (c'est-à-dire, au-delà du travail fourni) de notre projet était d'essayer de se mettre dans des conditions relativement proches d'un projet réel.

La liberté d'action dont nous avons profité durant ce projet s'est avéré être à la fois un avantage conséquent, puisqu'on ne ressent beaucoup moins les contraintes pédagogiques liées à un projet scolaire, mais également une difficulté notable puisqu'il n'y a personne pour nous avertir d'éventuels problèmes ou de mauvais choix dans notre travail.

Malgré cela, nous avons réussi à produire un prototype capable de fonctionner dans des cas simples, même si nous n'avons pas eu le temps de développer et de débugger toutes les fonctionnalités prévues au départ, nous sommes satisfaits du résultat obtenu et restons sur une bonne impression vis-à-vis du Projet de Spécialité.