

EPITA – S3# – Projet

Cahier des charges

Marwan BURELLE

1 Cadre

Le projet est à réaliser en groupe de quatre personnes (maximum). Il s'étend de Février à Juin. Le présent document présente les objectifs du projet, les différentes parties attendues ainsi que le planning de réalisation de celles-ci. Ce document présente également les différentes contraintes de réalisation.

2 Le sujet

L'objectif de ce projet est de réaliser un logiciel de type O.C.R. (*Optical Character Recognition*) dont la fonctionnalité principale est d'extraire depuis une image (*bitmap*) le texte contenu dans celle-ci.

Votre application prendra donc une image en entrée dans un format *standard* et produira un texte reconnu. Dans sa version finale, votre application disposera d'une interface graphique permettant de charger l'image, visualiser et éventuellement corriger le texte produit et enfin sauver celui-ci. Votre application aura également un aspect apprentissage, qui pourra être séparé de la partie principale, permettant d'entraîner votre réseau de neurones et de sauver et/ou recharger le résultat de l'apprentissage.

2.1 Description détaillée

Conceptuellement, le flux de traitement de votre OCR sera plus ou moins le suivant :

1. chargement de l'image ;
2. suppression des couleurs (niveau de gris, puis noir et blanc) ;
3. pré-traitement (*optionnel*) ;
4. détection des blocs de textes ;
5. détection des caractères ;
6. reconnaissance des caractères découverts ;
7. reconstruction du texte.

Les phases initiales (chargement, suppression des couleurs et pré-traitement éventuel) permettent de rentrer dans la partie principales de l'OCR. Le découpage de l'image en blocs puis en caractères (en passant éventuellement par un découpage des blocs en lignes) est nécessaire pour envoyer à la reconnaissance les morceaux d'images correspondants logiquement à des caractères.

On notera qu'aucune information de couleur n'est nécessaire pour la reconnaissance de caractères et qu'il est même fortement conseillé de travailler en noir et blanc.

2.2 Reconnaissance de caractères : réseau de neurones

La reconnaissance de caractères est la partie centrale de votre OCR. Cette partie nécessite une phase d'apprentissage pendant laquelle votre réseau de neurones va apprendre à reconnaître les différents caractères.

Un réseau de neurones est outil permettant d'apprendre une fonction (possiblement non linéaire) à l'aide d'exemples : dans la phase d'apprentissage du réseau vous allez fournir des entrées déjà identifiées (ici des blocs d'image) auxquelles votre réseau répondra, la marge d'erreur de ces réponses sera alors utilisée pour *corriger* les poids internes du réseau. Avec un certains nombre d'exemples, votre réseau finira par fournir les bonnes réponses

dans la majorité des cas. La sortie usuelle pour ce genre de problème est une probabilité par symbole du jeu de caractères, le symbole avec la plus forte probabilité sera retenu comme le symbole reconnu.

Pour implémenter votre réseau de neurones, voici quelques termes à rechercher : **apprentissage supervisé, perceptron multi-couches, rétro-propagation, descente de gradient.**

Le site <http://neuralnetworksanddeeplearning.com/> (en anglais) fournit beaucoup d'informations théoriques et pratiques qui pourraient vous être utiles (notamment pour la partie *paramétrage* de votre réseau dans le chapitre 3.)

2.3 Partie optionnelle : pré-traitement

L'objectif de cette partie est d'améliorer la qualité des images traitées. Cette partie n'est pas *nécessaire* dans l'ensemble de la chaîne de traitements, mais elle permettra d'accepter en entrée des documents directement issus d'un scanner ou d'une photo. Voici quelques exemples de traitements pertinents pour votre projet :

- redressement de l'image (détection d'angle de rotation et rotation de l'image),
- élimination des bruits parasites (grain de l'image, taches...),
- renforcement des contrastes...

3 Les contraintes du projet

Votre projet sera réalisé en C et devra compiler et fonctionner dans l'environnement de travail qui vous est fourni (Linux). Sauf cas particulier, seuls les outils déjà installés pourront être utilisés. Vous devrez respecter les points suivants :

- La version du standard C utilisée est **C99**.
- Vous pourrez utiliser **gcc** ou **clang** pour compiler votre code.
- Votre code doit compiler, sans warning avec au moins les options suivantes : `-Wall -Wextra -std=c99`
- Votre projet sera livré avec un fichier `Makefile` fournissant au moins les règles `all` et `clean`.
- Votre projet fournira un fichier `README` décrivant de manière concise comment utiliser votre application.
- Votre projet fournira un fichier `AUTHORS` contenant les logins des membres de votre groupe au format décrit plus loin.
- Tous les identifiants utilisés dans votre code ainsi que les commentaires devront être en anglais.
- Dans votre code, les lignes ne devront pas dépasser 80 colonnes.
- Il ne devra pas y avoir d'espaces en fin de ligne.

Le fichier `README` doit être au format texte et pouvoir être lu correctement dans un terminal avec une commande comme `cat` ou `less`. Il pourra éventuellement être au format *markdown*. Le fichier `AUTHORS` devra contenir une ligne par membre du groupe sous la forme : `* login (NOM Prenom)`

3.1 Contrôle de versions et rendus

Pour le projet vous allez disposer d'un dépôt `git` utilisable par tous les membres de votre groupe. Il est fortement conseillé d'utiliser les fonctionnalités de gestion de version de `git`.

Pour chaque soutenance un rendu de code vous sera demandé avant la soutenance.

Ce rendu de code s'effectuera à l'aide de `git` : les examinateurs effectueront un *clone* de votre dépôt sur la branche principale (`master`) à la date et à l'heure prévues pour le rendu. Seuls les fichiers présents sur votre dépôt au moment du *clone* seront utilisés pour montrer votre projet pendant la soutenance.

Votre dépôt devra contenir :

- Le fichier `AUTHORS`
- Le fichier `README`
- Votre code source et son fichier `Makefile`

Le dépôt ne devra pas contenir : de fichiers exécutables, de code source extérieur (bibliothèques, autres projets...), tout fichier binaire autre que des images pour la démonstration en soutenance, votre rapport (sous quelque forme que ce soit)...

4 Planning et livrables

Les dates clefs du projet sont les suivantes :

7 mars 2016 : constitution des groupes

11 avril 2016 : premières soutenances (semaine)

20 juin 2016 : soutenances finales (semaine)

4.1 Livrables pour chaque soutenance

En début de semaine de soutenance, vous devrez fournir (comme présenté dans la section 3.1) le code de votre projet qui sera utilisé pendant la soutenance. Lors de la soutenance, vous devrez fournir :

- Un rapport présentant votre travail (minimum 25 pages pour la première soutenance et 40 pour la soutenance finale) : présentation du groupe, répartition des charges, état d'avancement. . .
- Un plan de soutenance.

4.2 Découpage du projet et livrables techniques

Votre projet se découpe en plusieurs éléments. Un élément est considéré comme étant réalisé si votre programme est capable de réaliser la tâche attendue sur quelques exemples non triviaux. Bien évidemment, l'évaluation complète de l'élément dépendra de la qualité et de la robustesse face à des tests avancés.

- **Éléments devant être réalisés obligatoirement pour la première soutenance :**
 - chargement des images et suppression des couleurs ;
 - détection et découpage en blocs, lignes et caractères ;
 - *proof of concept* de votre réseau de neurones : au minimum un mini réseau capable d'apprendre la fonction *XOR*.
- **Éléments devant être commencés pour la première soutenance :**
 - sauvegarde et chargement des poids du réseau de neurones ;
 - jeu d'images pour l'apprentissage ;
 - manipulation de fichiers pour la sauvegardes des résultats
- **Éléments devant obligatoirement être réalisés pour la soutenance finale :**
 - tous les éléments obligatoires de la première soutenance ;
 - Le réseau de neurones complet et fonctionnel (apprentissage et reconnaissance) ;
 - reconstruction du texte et sauvegarde ;
 - Une interface utilisateur permettant d'utiliser ces éléments.
- **Éléments supplémentaires (évalués uniquement si la partie obligatoire est faite) :**
 - pré-traitement
 - intégration d'un correcteur orthographique ;
 - bonus : gestion du multi-colonnes, mise en page, images dans le texte. . .