|  |
| --- |
| HE-Arc |
| Rapport Traitement d’image |
| CardScan |

|  |
| --- |
| Jeremy Wirth, Wassim Ajjali, Tom Rey  10/05/2016 |

Table des matières

[Introduction 2](#_Toc450571961)

[But 2](#_Toc450571962)

[Technologies 2](#_Toc450571963)

[Fonctionnalités 2](#_Toc450571964)

[Exemple 3](#_Toc450571965)

[Maestro 3](#_Toc450571966)

[PostCard 4](#_Toc450571967)

[Permis de conduire 4](#_Toc450571968)

[Difficultés rencontrées 5](#_Toc450571969)

[Conclusion 5](#_Toc450571970)

[Références 5](#_Toc450571971)

# Introduction

Dans le cadre du cours de traitement d’image nous avons choisi de traité des images de carte de crédit ou des permis de conduire. Le principe étant de pouvoir identifier automatiquement les informations principales présentes sur une carte de crédit en utilisant une image de la carte avec certain traitement.

# But

Le but de cette application est d’utiliser l’image d’une carte de crédit/permis de conduire et d’effectuer des traitements pour pouvoir extraire directement les informations utiles tels que le nom et prénom du possesseur de la carte, le numéro de compte ainsi que la banque.

# Technologies

Pour reconnaître les caractères sur la carte nous utiliserons un OCR libre de droit (Tess4J) et la libraire OpenCV pour Java pour effectuer les fonctions de traitement.

# Fonctionnalités

Dans notre application, l’utilisateur n’aura que peu de fonctionnalités à sa disposition. En effet, le but est de pouvoir extraire des données prédéfinies de carte bancaire. Les données extraites sont définies à l’avance et l’utilisateur ne peut pas les changer. On verra aussi qu’avec les techniques utilisées, les types de cartes devront être elles aussi prédéfinies et l’utilisateur ne pourra donc pas essayer avec n’importe quelle carte.

Pour pouvoir extraire ces données, nous avons dû effectuer un traitement sur l’image avec la librairie OpenCV. Les différentes étapes du traitement sont :

1. Transformation de l’image en niveau de gris
2. Seuillage (La valeur du seuil dépend du type de carte)
3. Redimensionnement de l’image
4. Erosion (dépend du type de carte)

Une fois les traitements effectués sur l’image, nous découpons des rectangles aux zones qui contiennent l’information qui nous intéresse. Ces zones sont mesurées sur des cartes physique et décrite dans l’application.

# Exemple

## Maestro

Voici un premier exemple avec le traitement sur une carte de crédit Maestro. On remarque que la qualité de la photo n’est pas forcément excellente avec un reflet lumineux au centre. Ceci prouve que les traitements effectué avant l’utilisation de l’OCR sont suffisamment bon même pour une image avec des reflets.



Pour pouvoir obtenir une image capable d’être analyser par l’OCR, nous devons commencer par effectuer différents traitements sur notre image de base. Nous commençons par découper le tour de la carte pour ne pas garder le fond. Pour ce faire nous utilisons un seul qui dépendra de la couleur de la carte qui permettra de passer l’image en noir/blanc et ne garder que le texte.



On va ensuite découper l’image du dessus avec des rectangles pour s’occuper uniquement de l’information dont nous avons besoin. Les zones sont calculées en suivant un pourcentage depuis la distance calculé en vrai. Nous utilisons ensuite l’OCR Tess4J sur ces rectangles pour obtenir les informations comme sur l’image ci-dessous.



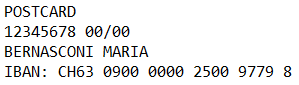
Voilà le résultat obtenu à la sortie du programme qui correspond aux informations de l’image de base.

## PostCard

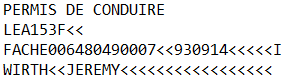
Le même procédé que pour une carte de crédit Maestro sera utilisé pour une PostCard. Les seules différences seront que le seuil devra être ajusté pour une carte de fond jaune et que les zones ne seront pas placé au même endroit.







## C:\Users\tom.rey\Documents\INF3\TraitementImage\Projet\CardScan\images\carte2.jpgPermis de conduire



# Difficultés rencontrées

Les deux grandes difficultés de ce projet sont de trouver automatiquement le seuil pour l’image. Il est en effet facile de calculer le seuil une fois que l’on connaît la carte et sa couleur mais plus difficile de le calculer automatiquement.

La position des zones est aussi un problème car elles varient suivant le type de carte utilisé. On ne peut donc pas prendre une image de n’importe quel carte de crédit mais seulement celle que nous aurons défini.

# Conclusion

Pour conclure ce projet, les fonctionnalités principales et l’idée que nous voulions mettre en place sont disponibles et fonctionnels. Nous sommes capables d’extraire les informations que nous avons jugées importantes à partir d’une image d’une carte de crédit. Même si l’image n’est pas d’une bonne qualité l’information est toujours correcte.

Cependant, des améliorations sont envisageables. Le point qui serait à améliorer en premier est le fait que nous traitons chaque carte individuellement. Il serait pratique de pouvoir traiter des images de nouvelle carte (par exemple : UBS) et de ne pas devoir modifier le code. Il faudrait donc trouver un code générique qui va traiter n’importe quel type de carte.

# Références

<http://docs.opencv.org/2.4/doc/tutorials/introduction/java_eclipse/java_eclipse.html>

<http://opencv-java-tutorials.readthedocs.io/en/latest/>

<http://tess4j.sourceforge.net/>

<http://stackoverflow.com/questions/22464503/how-to-use-opencv-to-calculate-hsv-histogram-in-java-platform>