



## Processus de dématérialisation PROJET Dématérialisation

*Réaliser par :*  
Mohammed BENAOU  
Mohammed RASFA  
Anass ABDELLAOUI

---

### Liste des exercices

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>choix de technologie</b>	<b>2</b>
<b>3</b>	<b>Gestion du projet</b>	<b>3</b>
3.1	Outil de l'équipe et découpage de tâches . . . . .	3
3.2	Mise en place de l'équipe . . . . .	3
<b>4</b>	<b>Fonctionnement</b>	<b>4</b>
<b>5</b>	<b>Catégorisation des documents</b>	<b>4</b>
<b>6</b>	<b>Extraction de champs prédéfinis sur les factures et exportation en fichier csv</b>	<b>6</b>
<b>7</b>	<b>Mode d'emploi</b>	<b>9</b>
<b>8</b>	<b>Code source</b>	<b>12</b>
<b>9</b>	<b>Conclusion</b>	<b>14</b>

## 1 Introduction

Le présent rapport a pour but de décrire le déroulement de l'implémentation d'un programme de processus de dématérialisation de documents en utilisant la technologie Matlab. L'architecture logicielle du système est basée sur l'intégration générique des outils de reconnaissance de caractères du commerce appelés OCR ( Optical Character Recognition ).

Ce rapport contient l'ensemble des éléments du projet, le plan du projet s'articule sur trois piliers fondamentaux, dans la première on expliquera la gestion du projet, ensuite on procédera à clarifier le processus permettant de catégoriser les images, finalement on se concentrera à la méthode permettant localiser et de reconnaître les 4 informations **Nom image, N facture, Date, Nom, Montant**, par la suite ces données seront alors enregistrées dans un fichier *CSV*.

## 2 choix de technologie



FIGURE 1 – Technologie

**MATLAB** est un acronyme pour «matrice laboratoire» et fait référence à un langage de programmation de haut niveau et un environnement de programmation développé par The MathWorks société de logiciels informatiques techniques.

**OCR** est l'abréviation de " reconnaissance optique de caractères " et se réfère au processus de lecture des données sous forme imprimée et en identifiant les modèles optiques qui correspondent à des lettres, des chiffres et autres caractères.

Toutes les étapes OCR - segmentation, extraction caractéristiques et la classification ont été réalisées sous Matlab 2016. En effet, outre ses possibilités de traitements d'images, cette version possède un OCR intégré utile pour l'extraction des champs sur les factures.

### 3 Gestion du projet

Nous avons réalisé une répartition des tâches pour chaque membre de l'équipe une tâche(s) à faire, à travers l'outil [Trello](#), ce dernier est un outil qui nous permet de gérer tous les aspects de la gestion de projet.



Nous avons 5 jours pour rendre notre travail.  
=> Voici les deux parties qu'on va développer :

1. CATEGORISATION DES DOCUMENTS
2. RECONNAISSANCE DU FICHIER
3. ENVOI à La BASE DE DONNEES

#### 3.1 Outil de l'équipe et découpage de tâches



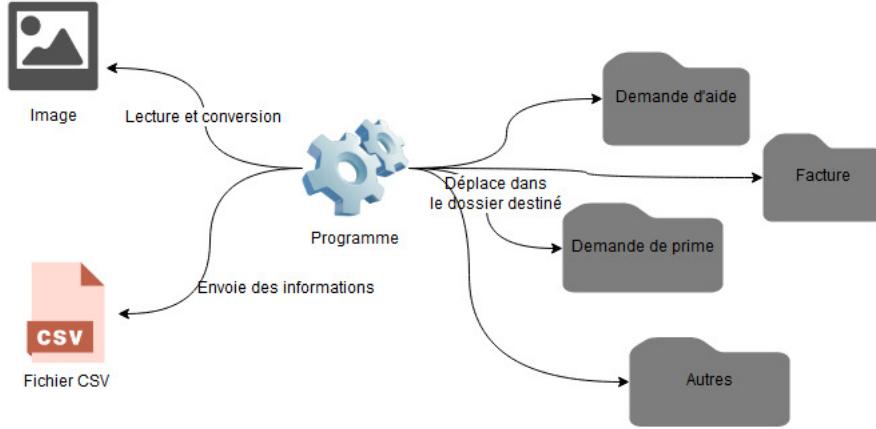
Afin de pouvoir travailler en équipe efficacement, nous utilisons l'outil GitLab. Cela nous permet de partager notre travail, de le récupérer instantanément, et de travailler en parallèle sur plusieurs tâches.

#### 3.2 Mise en place de l'équipe

Notre équipe est constituée de trois personnes, cette dernière devait suivre le schéma-type d'une équipe utilisant la méthode Scrum :

- **Mohammed RASFA** :Un chef de projet,
- **Mohammed BENAOU, Anass ABDELLAOUI, Mohammed RASFA** :développeurs,

## 4 Fonctionnement



Suite à une phase d'acquisition du document, ce dernier va être catégorisé afin d'être envoyé directement au bon dossier.

il existe quatre types de dossiers :

- Demande de prime.
- Demande d'aide.
- Facture.
- Autres.

le dossier **autres** recueille les fichiers qui n'appartiennent pas aux autres dossiers.

## 5 Catégorisation des documents

Tout notre programme se base sur une ou plusieurs images en entrée. La première chose à faire était donc de réussir à charger une image afin de l'exploiter.

Il existe différents formats de représentation des fichiers images : TIFF, JPEG, GIF, PNG ... ,c'est la raison pour laquelle qu'on a déclaré les formats suivants :

```
1 D =[dir(fullfile('*.pdf')) ; dir(fullfile('*.png')) ; dir( fullfile('*.jpeg')) ; dir(fullfile('*.jpg'))];  
2 %D = dir(pwddir);  
3 chemin = strcat(D(1).folder,'/');
```

dépendant des propriétés de l'image, comme le chromatisme et Pour la réalisation de cette première partie, nous sommes partis de l'hypothèse que les documents fournis avaient été pré-traités et ne concernaient que les types suivants : Prime de déménagement, Aide au logement et facture. Au lancement du programme, on indique deux documents types (Aide au logement et Prime de déménagement). Ces derniers subissent un traitement de binarisation suivant les étapes suivantes :

- calcul du seuil de l'histogramme avec la méthode d'Otsu
- binarisation de l'image

La première étape consiste à lire l'image dans l'espace de travail MATLAB comme un fichier bitmap. Il s'agit d'un type de fichier graphique dans laquelle chaque élément d'image, ou pixel, correspond à un ou plusieurs chiffres binaires, ou bits, dans la mémoire. Le chargement en mémoire d'une image se fait avec la fonction **imread** :

```

1 original_prime = imread('Originaux\CourrierDev001.png');
2 seuil = graythresh(original_prime);
3 primeNB = im2bw(original_prime, seuil);
4 original_aide = imread('Originaux\CourrierDev002.png');
5 seuil = graythresh(original_aide);
6 aideNB = im2bw(original_aide, seuil); original_prime = imread(
   'Originaux\CourrierDev001.png');
7 seuil = graythresh(original_prime);
8 primeNB = im2bw(original_prime, seuil);
9 original_aide = imread('Originaux\CourrierDev002.png');
10 seuil = graythresh(original_aide);
11 aideNB = im2bw(original_aide, seuil);

```

On obtient ainsi les images binaires des trois documents étudiés.

Après avoir créé trois dossiers de classification, le programme liste tous les fichiers contenus dans le lot. Par la suite, une technique appelée " battage " est utilisé pour convertir l'image en niveaux de gris en une image binaire.

Une image en couleur est représentée de la manière suivante : pour chaque pixel, un niveau de rouge, de vert et de bleu est contenu en mémoire. Cette valeur va de 0 à 255. Pour passer une image en niveau de gris, il suffit de faire la somme de ces 3 valeurs divisée par 3. On obtient un nouveau nombre entre 0 et 255. La cellule de codes MATLAB nécessaires pour convertir l'image en niveaux de gris d'une image binaire ressemble à ceci :

```

1 seuil = graythresh(img);
2 imgNB = im2bw(img, seuil);

```

**im2bw** c'est l'opérateur de binarisation. Une image binaire peut être également obtenue en utilisant des opérateurs de comparaison et des opérateurs logiques. mises à l'échelle pour correspondre aux tailles des modèles calculés précédemment

```

1 imgR = imresize(imgNB, size(primeNB));
2 imgR2 = imresize(imgNB, size(aideNB));

```

calculs des différences entre les images redimensionnées et leur modèle : plus les images sont similaires, plus il y a de pixels noirs.

```

1 Z = imabsdiff(primeNB, imgR);
2 Z2 = imabsdiff(aideNB, imgR2);

```

calculs des moyennes des pixels pour les deux images résultantes.

```
1 MoyPrime=mean(Z(:));  
2 MoyAide=mean(Z2(:));
```

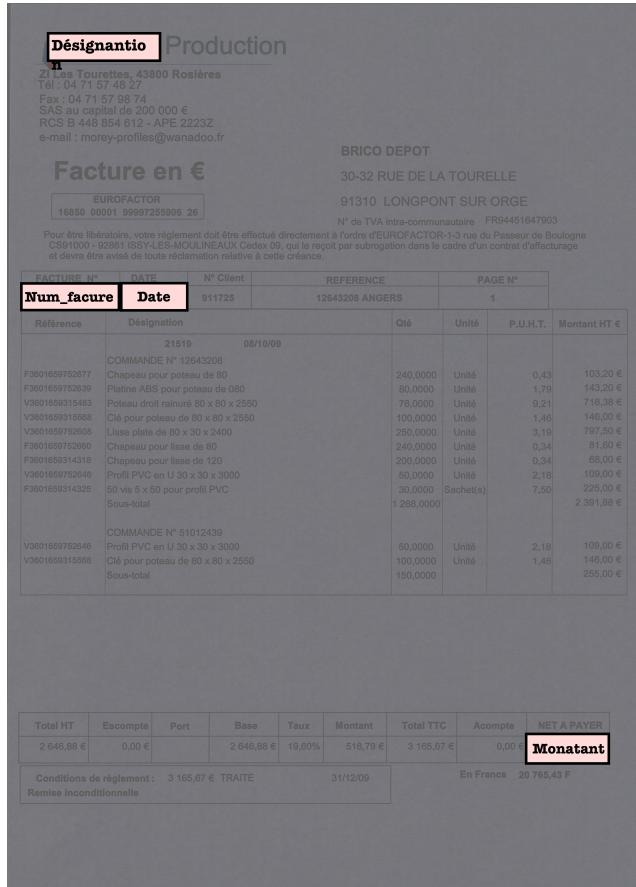
la moyenne (“prime” ou “aide”) située sous un certain seuil indique l’appartenance de l’image étudiée. Si aucune des moyennes n’est située sous ce seuil, alors le document est probablement une facture.

```
1 if MoyPrime < 0.1  
2 imwrite(img, [PrimesDem '\\' D(n).name], 'png')';  
3 elseif MoyAide < 0.1  
4 imwrite(img, [AidesLog '\\' D(n).name], 'png')';  
5 else  
6 imwrite(img, [Factures '\\' D(n).name], 'png')';  
7 end
```

**Résultat sur le lot d’image fourni :** L’exécution du programme sur ce lot présente un taux de réussite de 100%.

## 6 Extraction de champs prédéfinis sur les factures et exportation en fichier csv

Pour cette deuxième partie, nous sommes partis de l’hypothèse qu’il n’existe que trois type de factures différentes (Morey, Uniross et Alibert). En rose figurent respectivement des blocs de textes. Ces blocs qu’on veut les récupérer et les stocker dans le fichier CSV.



En premier lieu, le programme liste tous les fichiers contenus dans le dossier “Factures”. Une fois cette liste établie, on ouvre un fichier csv qui contiendra les données extraites des champs pour chaque facture. Ensuite, on définit des zones de recherches rectangulaires spécifiques aux champs recherchés pour chaque type de facture (par exemple, le montant ne sera pas localisé au même endroit sur une facture Alibert et une facture Morey). Puis, une boucle effectue un traitement sur l’ensemble des factures listées. Ce traitement est composé des étapes suivantes :

- binarisation de l’image avec la méthode d’Otsu
- lancement du calcul OCR sur la zone de recherche de l’entête

```
1 ocrResults = ocr(imgNB, zone_recherche);
```

le texte reconnu est mis en majuscule et comparé aux textes ‘MOREY’, ‘UNI-ROSS’ et ‘ALIBERT’. On détecte ainsi le type de la facture en cours.

```
1 texteReconnu = upper(ocrResults.Text);
2 if isempty(strfind(texteReconnu, 'MOREY')) == 0
3 fact = 2;
```

```
4 nom = 'MOREY';
5 elseif isempty(strfind(texteReconnu, 'UNIROSS')) == 0
6 fact = 3;
7 nom = 'UNIROSS';
8 else
9 fact = 1;
10 nom = 'ALLIBERT';
11 end
```

---

trois calculs OCR sont ensuite lancés sur les zones recherchés (date, numéro de facture et montant). Chaque résultat est stocké dans un tableau.

```
1 for i=1:1:3
2 ocrResults = ocr(imgNB, zones{i, fact});
3 item{i} = strtrim(ocrResults.Text);
4 end
```

---

le programme inscrit ensuite le nom de l'image, sa désignation et les valeurs du tableau précédent dans le fichier csv. **Résultat sur le lot d'image fourni**  
Le taux de réussite sur les factures testées est d'environ 100%. En effet, deux montants de factures sont mal détectés par l'OCR de Matlab.

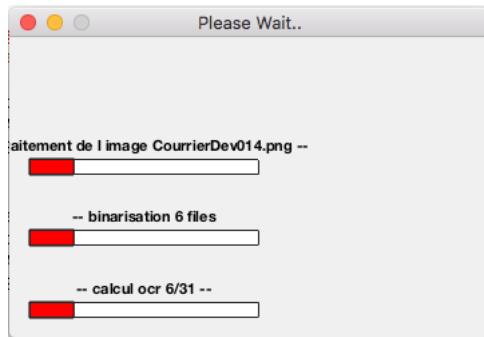
## 7 Mode d'emploi

Après avoir exécuter notre programme, on aura ce menu comme résultat :

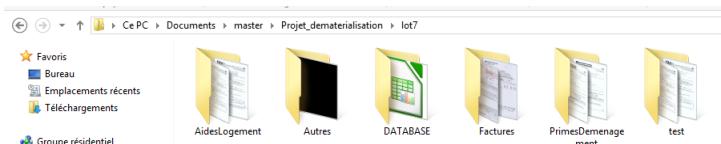
```
Trial>> projet
 1- CATEGORISATION DES DOCUMENTS
 2- RECONNAISSANCE ET BASE DE DONNEES
 3- Quitter
```

Validez votre choix |

- Le choix numéro 1 permet de catégoriser les documents : au moment de la validation du choix une petite fenêtre apprête indiquant le déroulement du traitement des fichiers tel que la binarisation.

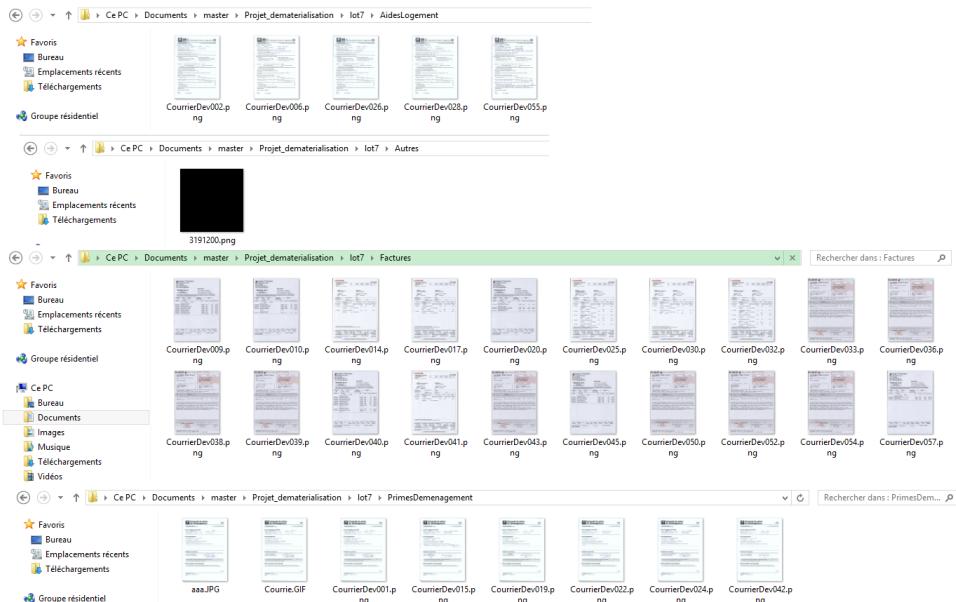


À la fin du traitement on trouve les trois dossiers générés automatiquement par notre programme (Aides de logement, Factures, Primes de déménagement) avec un dossier nommé autre contenant les documents qui ne correspondent à aucune catégorie.



Trente et un documents sont utilisés pour l'apprentissage, le tout est réparti dans les dossiers qu'on a déjà créé.

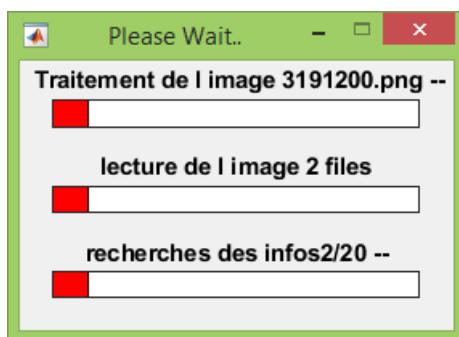
En faisant varier le contenu de ces différents dossiers, le pourcentage de reconnaissance est à 100%.



- Le choix numéro 2 permet de la reconnaissance et le stockage dans la base de données .

Au moment du démarrage du traitement, l'utilisateur aura cette vue qui indique le où les fichiers en cours de traitement.

cette vue comporte trois contrôles de chargement, chacun correspond au déroulement d'une opération.



À la fin du traitement on obtient un fichier CSV qui regroupe de différentes informations extraites des documents analysées de type facture. on remarque cinq types d'informations :

- nom\_image
- Désignation
- Numéro\_facture
- Date
- Montant

The screenshot shows a LibreOffice Calc spreadsheet window. The title bar indicates the file is named 'factures.csv'. The main area displays a table with the following data:

	Nom	Designation	Numéro_facture	Date	Montant
1	nom_image			13/10/09	3272 14
2	CourrierDev009.png	MOREY	19566	13/10/09	2787 42
3	CourrierDev010.png	MOREY	19569	13/10/09	2787
4	CourrierDev014.png	UNIROSS	19003417	20/10/09	226.28
5	CourrierDev018.png	UNIROSS	19003418	20/10/09	216.75
6	CourrierDev020.png	MOREY	19565	13/10/09	4910 38
7	CourrierDev025.png	UNIROSS	19003419	20/10/09	305.98
8	CourrierDev030.png	UNIROSS	19003420	20/10/09	267.84
9	CourrierDev031.png	MOREY	19567	13/10/09	307.77
10	CourrierDev033.png	ALLIBERT	1510043434	16.10.2009	1.829188
11	CourrierDev035.png	ALLIBERT	1510043423	16.10.2009	1.219.92
12	CourrierDev038.png	ALLIBERT	1510043424	16.10.2009	1.524
13	CourrierDev039.png	ALLIBERT	1510043427	16.10.2009	1.524
14	CourrierDev049.png	MOREY	19567	13/10/09	3185 87
15	CourrierDev041.png	UNIROSS	19003422	20/10/09	178.68
16	nom_image				
17					
18	CourrierDev009.png	MOREY	19566	13/10/09	3272 14
19	CourrierDev010.png	MOREY	19569	13/10/09	2787 42
20	CourrierDev014.png	UNIROSS	19003417	20/10/09	226.28

Le choix numéro 3 : pour quitter l'exécution.

## 8 Code source

```

1 %nb = input(' Nombre de types de documents différents ');
2
3 D =[dir(fullfile('.','*.gif'));dir(fullfile('.','*.png'));dir(
4     fullfile('.','*.jpeg'));dir(fullfile('.','*.jpg'));dir(fullfile(
5     '.','*.pdf'))];
6
7 chemin = strcat(D(1).folder,'\\');
8 nbImg = size(D,1);
9 arg = [' le dossier contient ',num2str(nbImg), ' images '];
10 condition=true;
11 while condition
12     prompt=' 1-CATEGORISATION DES DOCUMENTS \n 2-RECONNAISSANCE ET
13         BASE DES DONNEES \n 3-QUITTER \n \n Valider votre choix ';
14     c=input(prompt);
15     if c==1
16         disp(arg);
17         h = multiwaitbar(3,[0 0 0 0],{ 'waitbar1 ... ', 'waitbar2 ... ', '
18             waitbar3 ... '} );
19
20     for n=1:1:nbImg
21         info = [ '--- Traitement de l image ',D(n).name,' ---'];
22         info2 = [ '--- Traitement de l image ',num2str(n),' / ',num2str(
23             nbImg), ' ---'];
24         disp(info2);
25         bin=[--- binarisation',sprintf(' %d files ',n)];
26         oc=[--- calcul ocr ',num2str(n),' / ',num2str(nbImg), ' ---'];
27         multiwaitbar(3,[n/nbImg,n/nbImg,n/nbImg],{ info ,bin ,oc },h );
28         fic = strcat(chemin,lower(D(n).name));
29
30         img = imread(fic);
31
32         %disp('--- calcul du seuil et binarisation ---');
33         seuil = graythresh(img);
34         imgNB = imbinarize(img, seuil);
35         % disp('--- calcul ocr ---');
36         ocrResults = ocr(imgNB);
37         texteReconnu = upper(ocrResults.Text);
38         %disp('--- classement image ---');
39
40         if ~contains(texteReconnu, 'DEMANDE') == 0 && ~contains(
41             texteReconnu, 'PRIME') == 0 && ~contains(texteReconnu,
42                 'DEMENAGEMENT') == 0 || ~contains(texteReconnu, 'DÉMÉNAGEMENT')
43                 == 0
44             mkdir PrimesDemenagement;
45             imwrite(img, ['.\PrimesDemenagement\' D(n).name], 'png');
46         elseif ~contains(texteReconnu, 'DEMANDE') == 0 && ~contains(
47             texteReconnu, 'AIDE') == 0 && ~contains(texteReconnu, 'LOGEMENT
48 ') == 0
49             mkdir AidesLogement;
50             imwrite(img, ['.\AidesLogement\' D(n).name], 'png');
51         elseif ~contains(texteReconnu, 'FACTURE') == 0
52             mkdir Factures;
53             imwrite(img, ['.\Factures\' D(n).name], 'png');
54         else
55             mkdir Autres;
56             imwrite(img, ['.\Autres\' D(n).name], 'png');
57         end
58         mes=msgbox('Classement image', 'Success');

```

```

50 end
51 disp("____");
52 elseif c==2
53 H = multiwaitbar(3,[0 0 0 0],{ 'waitbar1...','waitbar2...','
54 waitbar3...'});
55 D2 =[dir(fullfile('.\Factures','*.gif'));dir(fullfile('.\Factures',
56 , '*.png'));dir(fullfile('.\Factures','*.jpeg'));dir(fullfile(
57 ,'\Factures','*.jpg'))];
58 chemin2 = D2(1).name;
59
60
61 nbImg2 = size(D2,1);
62 mkdir DATABASE;
63 fid = fopen('.\DATABASE\factures.csv', 'a');
64 fprintf(fid, 'Nom_image;Désignation;Numéro_facture;Date;Montant\n')
65 ;
66
67 zone_recherche = [5 5 1250 1220];
68
69 zones{1,1} = [1575 55 325 55];
70 zones{2,1} = [1795 295 275 45];
71 zones{3,1} = [2100 3040 230 60];
72
73 zones{1,2} = [150 1125 145 55];
74 zones{2,2} = [495 1125 170 55];
75 zones{3,2} = [2145 2870 170 60];
76
77 zones{1,3} = [1250 330 195 60];
78 zones{2,3} = [2040 330 195 60];
79 zones{3,3} = [2100 3205 140 55];
80
81 item = cell(3);
82
83 for n=1:1:nbImg2
84 info1 = [ ' Traitement de l image ',D(n).name, ' ____'];
85 info2 = [ '____ Traitement de l image ',num2str(n), '/', num2str(
86 nbImg2), ' ____'];
87 disp(info2);
88 bin1=[ 'lecture de l image',sprintf(' %d files ',n)];
89 oc1=[ 'recherches des infos ',num2str(n), '/', num2str(nbImg2), ,
90 ' ____'];
91 multiwaitbar(3,[n/nbImg2,n/nbImg2,n/nbImg2],{ info1 ,bin1 ,oc1 },H)
92
93 info = [ '____ lecture de l image ',num2str(n), '/', num2str(nbImg2)
94 , ' ____'];
95 % disp(info);
96 fic = strcat('./Factures/ ',D2(n).name);
97
98 img = imread(fic);
99 % disp('-- calcul du seuil et binarisation --');
100 seuil = graythresh(img);
101 imgNB = imbinarize(img, seuil);
102 % disp('-- calcul ocr --');
103 ocrResults = ocr(imgNB, zone_recherche);
104 texteReconnu = upper(ocrResults.Text);
105 % disp('-- recherches des infos --');

```

```

104
105     if ~contains(texteReconnu, 'MOREY') == 0
106         fact = 2;
107         nom = 'MOREY';
108     elseif ~contains(texteReconnu, 'UNIROSS') == 0
109         fact=3;
110         nom = 'UNIROSS';
111     else
112         fact = 1;
113         nom = 'ALLIBERT';
114     end
115
116     for i=1:1:3
117         ocrResults = ocr(imgNB, zones{i, fact});
118         item{i} = strtrim(ocrResults.Text);
119     end
120     fprintf(fid,[num2str(D2(n).name) ';' nom ';' regexp替換(item{1},
121         ',' ',') ';' regexp替換(item{2}, ',', ',') ';' regexp替換(item{3},',
122         ',' ',') '\n']);
123
124     fclose(fid);
125     delete(H.figure)
126     close(H)
127     message=msgbox('Le fichier .csv ','Success');
128     disp("_____");
129     elseif c==3
130         break;
131     else
132         disp('Entrée invalide !');
133         disp("_____");
134     end
135 end

```

## 9 Conclusion

Dans la fin, il fallait penser aux objectifs et aux méthodes qu'on a suivi pour atteindre le but de ce travail. sans oublier que nous avons rencontré quelques difficultés au niveau de l'implémentations du programme de chaque fonction et les erreurs associées.

En revanche la création de l'outil de processus de dématérialisation en Matlab nous a permis de découvrir plus profondément plusieurs aspects concernant Les OCR et la dématérialisation en général.