

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE MOUHAMED BOUDIAF -M'SILA

Faculté des mathématiques et de  
l'informatique  
Département de l'informatique  
N° : .....



DOMAINE : Mathématique et  
Informatique  
FILIERE : Informatique  
OPTION : Système d'information et  
génie logiciel

Mémoire Présenté Pour L'obtention  
Du Diplôme De Master Académique

Par : BOUDJENAH Toufik

LEMAAREG Abdelbasit

Intitulé :

**Conception et réalisation d'une application  
mobile de correction automatique de QCM**

Soutenu devant le jury composé de :

AMRI Said

Université de M'Sila

Président

BANAZI Makhoulf

Université de M'Sila

Rapporteur

BOUCETTA Mouhamed

Université de M'Sila

Examineur

**Année universitaire : 2021/2022**

## **Remerciements**

Je tien à témoigner ma reconnaissance à DIEU tout puissant, qui m'a aidé et béni par sa volonté durant toute cette période. Ma profonde gratitude et sincères remerciements vont à notre promoteur D<sup>r</sup> BENAZI Makhloof pour sa présence continuelle, son encouragement tout au long de ce travail. J'adresse mes remerciements aux membres du jury, devant qui, nous avons l'honneur d'exposer notre travail, et qui ont pris la peine de lire ce mémoire pour juger son contenu. Je réserve ici une place particulière pour remercier vivement tous ceux qui, d'une manière ou d'une autre, M'ont aidés et encouragés à la réalisation de ce modeste travail. COUACHE Ammar et ABIZA L'Eulmi.

BOUDJENAH Toufik

## Remerciements

Nous adressons en premier lieu notre reconnaissance à notre DIEU tout puissant, de nous avoir permis d'arriver là, car sans lui rien n'est possible.

Nous tenons tout d'abord à remercier BENAZI Makhloof notre encadreur de mémoire, pour son encadrement, son suivi et ses conseils lors de la réalisation de notre mémoire

Sans oublier mon binôme BOUDJENAH Toufik pour son soutien moral, sa patience.

Nous tenons ensuite à remercier tous nos professeurs et enseignants de département d'informatique qui ont contribué à notre formation.

Nous souhaitons aussi remercier mesdames et messieurs les membres du jury pour leur précieux temps accordé à l'étude de notre mémoire.

LEMAAREG Abdelbasit

## Dédicace

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect : à mon cher père BOUZID.

A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureux, qui nous a quitté en silence : mon adorable mère.

A mes chère frères et sœurs chacun a son non

A ma petite famille : ma femme et mes enfants Salsabile, Mohamed et Assile qui mon encourager et mon soutenir tout au long de mes études. Que Dieu les protège et leurs offre la chance et le bonheur.

Merci pour leurs amours et leurs encouragements.

Sans oublier mon binôme LEMAAREG Abdelbassit pour son soutien moral, sa patience

À mon encadreur le docteur Makhoulf BENAZI de sa patience son soutien et encouragement et de m'avoir accepter et accepte de modeste travail

A mes chères amies de l'université de M'Sila là où je me suis trouvé comme si j'étais chez moi, chacun de son nom surtout :

ZOUREG Said, BASTI Chouaib, BENTAYEB Amine.

A mon collègue et ami d'études et de résidence, ou on a vécu de très bonne moments Mr BECHERE Messaoud.

A mes enseignants qui mon toujours encourager et soutenues afin de réussir mes études : Dr DEBBI Hichem, Dr BRAHIMI Belkacem, Dr BOUGHRARA Seddik ;

A mes chers amis de travail : ABIZA Leulmi, LANOIR Youcef, BEARAB Redha

A ces toutes personne de dédie ce travail

TOUFIK



## Dédicace

Je dédie ce mémoire :

A mes très chers parents pour leur soutien et  
encouragement durant toutes mes années  
d'études et sans lesquels je n'aurais jamais  
réussi

Et à ma famille

A tous mes professeurs et enseignants que j'ai  
eu durant tout mon cursus scolaire et qui m'ont  
permis de réussir dans mes études.

A tous mes amis de l'Université et d'ailleurs

A toute personne ayant contribué à ce travail  
de près ou de loin

***ABDELBASSIT***



# Table des matières

Remerciements	I
Dédicace	III
Table des matières	VI
Liste des figures	X
Liste des tableaux :	XII
Liste des abréviation et acronymes	XIII
INTRODUCTION GENERALE :	1
1. GENERALITES SUR LE TRAITEMENT D'IMAGES	3
1.1 Introduction	3
1.2 Définition de l'image :	3
1.3 Image numérique :	3
1.4 Résolution et/ou de Définition pour une photo numérique :	4
1.5 Acquisition d'une image :	4
1.6 Caractéristiques d'une image numérique :	5
1.6.1 Dimension :	5
1.6.2 Résolution :	5
1.6.3 Bruit :	6
1.6.4 L'histogramme :	6
1.6.5 Luminance :	6
1.6.6 Contraste :	7
1.7 Les types d'image :	7
1.7.1 Image binaire (noire ou blanc) :	7
1.7.2 Image à niveaux de gris :	7
1.7.3 Image en couleurs (RVB) :	8
1.8 Système de traitement d'images :	8

<b>1.8.1</b>	Pré-traitement d'image :	9
<b>1.8.2</b>	Modification d'histogramme :	9
<b>1.8.3</b>	Réduction du bruit par filtrage :	9
1.9	Extraction de contours :	10
1.10	Segmentation en régions :	10
<b>1.10.1</b>	Accroissement de région :	11
<b>1.10.2</b>	Méthodes du type "split and merge" :	12
1.11	CONCLUSION :	14
2.	CORRECTION AUTOMATIQUE DE COPIES AU FORMAT QCM	16
2.1	Introduction :	16
2.2	Acquisition d'images :	18
2.3	Filtrage :	18
2.4	Binarisation :	18
2.5	Correction de l'inclinaison :	21
2.6	Localisation des coins bas :	25
2.7	Extraction de la zone d'intérêt :	26
2.8	Détection des Cercles :	28
2.9	Lecture des choix :	33
2.10	Détection du matricule :	34
2.11	Conclusion :	38
3.	ANALYSE ET CONCEPTION	40
3.1	Introduction	40
3.2	Analyse	40
3.2.1	Identification des acteurs :	40
3.2.2	Les acteurs de notre système	40
3.3	Diagramme de contexte	41
3.4	Spécification des besoins	41

3.4.1	Les besoins fonctionnels :	41
3.4.2	Les besoins non fonctionnels :	42
3.5	Spécification des cas d'utilisation	42
3.5.1	Définition	42
3.5.2	Relations entre cas d'utilisation	42
3.5.3	Diagrammes de cas d'utilisation	42
3.6	Description textuelle des cas d'utilisation	44
3.7	Conception	50
3.7.1	Diagramme de séquence	50
3.7.2	Diagramme de classe :	54
3.7.3	Conception de la base de données	55
3.7.4	Passage du diagramme de classe au modèle relationnel de données	55
3.7.5	<b>Le modèle relationnel</b> de données	55
3.8	Conclusion	56
4.	IMPLEMENTATION ET RESULTAT	58
4.1	Introduction :	58
4.2	Matériel utilisé :	58
4.3	Environnement de développement :	59
4.3.1	Fonctionnalités :	60
4.4	Langage de programmation java :	61
4.5	Système de gestion de base de données : SQLite	61
4.6	Présentation des interfaces :	62
4.6.1	Interface d'authentification :	62
4.6.2	Interfaces accueil	63
4.6.3	Fenêtre étudiant :	63
4.6.4	Fenêtre Sujet :	64
4.6.5	Fenêtre questions :	65

4.6.6 Fenêtre camera :	65
4.6.7 Fenêtre résultat :	66
4.7 Conclusion :	68
Conclusion générale	59
BIBLIOGRAPHIE	41
Résumé	42

## Liste des figures

<b>Figure</b>	<b>Titre</b>	<b>Page</b>
<b>Figure 1.1</b>	Sous image de taille 5 x 5	04
<b>Figure 1.2</b>	Extrait d'une image de taille 5 X 5 Pixels	05
<b>Figure 1.3</b>	Histogramme et palette associés à une image	06
<b>Figure 1.4</b>	Image binaire avec un tableau de valeurs correspondant	07
<b>Figure 1.5</b>	Image numérique de niveau de gris	07
<b>Figure 1.6</b>	Image numérique en couleur	08
<b>Figure 1.7</b>	Schéma d'un système de traitement d'images	08
<b>Figure 1.8</b>	Différents types de contours	10
<b>Figure 1.9</b>	Voisinage d'un pixel	12
<b>Figure 1.10</b>	Découpage par quadtree d'une image 8x8. Traitement avec seuil=100%	13
<b>Figure 1.11</b>	Problème de découpage arbitraire de régions dans la phase de "split"	13
<b>Figure 2.1</b>	Le model de QCM utilisé avec différents blocs	16
<b>Figure 2.2</b>	Android 7.0 (Nougat)	17
<b>Figure 2.3</b>	Schéma synoptique de la méthode proposée.	17
<b>Figure 2.4</b>	Exemple d'une image scanné et ses propriétés	18
<b>Figure 2.5</b>	Découpage d'images (partie haute gauche et partie haute droite)	19
<b>Figure 2.6</b>	Résultat du l'étiquetage	19
<b>Figure 2.7</b>	Le Centre de gravité des coins de l'image découpée.	20
<b>Figure 2.8</b>	Les nouvelles coordonnées de centre de gravité de coin droit	21
<b>Figure 2.9</b>	Rotation d'une image autour de son origine.	21
<b>Figure 2.9</b>	Rotation d'une image autour de son origine.	21
<b>Figure 2.10</b>	Rotation de l'image avec des bordures noires	22
<b>Figure 2.11</b>	Organigramme de la méthode proposé (élimination des pixels noire)	23
<b>Figure 2.12</b>	Organigramme de la méthode de détection de la ligne	24
<b>Figure 2.13</b>	Image découpé (partie Bas gauche et partie bas droit)	25
<b>Figure 2.14</b>	Les nouvelles coordonnées du centre de gravité des coins bas.	26
<b>Figure 2.15</b>	Extraction de la zone d'intérêt	27
<b>Figure 2.16</b>	Histogramme horizontale et verticale du chiffre 5.	28
<b>Figure 2.17</b>	Remplissage et élimination de régions.	29
<b>Figure 2.18</b>	Histogramme verticale et horizontale de l'image de la zone d'intérêt.	30
<b>Figure 2.19</b>	Organigramme de localisation horizontale des cercles	31

<b>Figure 2.20</b>	Organigramme de localisation verticale des cercles	32
<b>Figure 2.21</b>	La zone de choix en rouge.	33
<b>Figure 2.22</b>	Extraction de la zone de choix. <b>(a)</b> zone choisie. <b>(b)</b> zone vide.	33
<b>Figure 2.23</b>	Modèle de matricule utilisé	34
<b>Figure 2.24</b>	Schéma synoptique de la méthode de détection du matricule	34
<b>Figure 2.25</b>	Extraction de la zone de matricule	35
<b>Figure 2.26</b>	Résultat d'opération d'élimination des régions.	36
<b>Figure 2.27</b>	Histogramme verticale et horizontale de la zone de matricule.	36
<b>Figure 2.28</b>	Localisation et extraction de matricule.	36
<b>Figure 2.29</b>	Résultat d'opération d'érosion suivie de dilatation.	37
<b>Figure 3.1</b>	Diagramme de contexte.	41
<b>Figure 3.2</b>	Diagramme détaillé du cas d'utilisation « correction automatique ».	43
<b>Figure 3.3</b>	Diagramme de séquence du cas " S'authentifier".	51
<b>Figure 3.4</b>	Diagramme de séquence du cas " Ajouter un étudiant ".	52
<b>Figure 3.5</b>	Diagramme de séquence du cas « Ajouter des questions ».	53
<b>Figure 3.6</b>	Diagramme de classe	54
<b>Figure 4.1</b>	Laptop HP Pavilion	58
<b>Figure 4.2</b>	smartphone Samsung Galaxy S7 Edge	59
<b>Figure 4.3</b>	Image Android studio 4.3 (Canary 8)	60
<b>Figure 4.4</b>	Logo de SQLite	62
<b>Figure 4.5</b>	Fenêtre « Authentification »	62
<b>Figure 4.6</b>	Fenêtre « créer un compte	62
<b>Figure 4.7</b>	Fenêtre « accueil »	63
<b>Figure 4.8</b>	Fenêtre étudiant	63
<b>Figure 4.9</b>	Fenêtre ajouter étudiant	63
<b>Figure 4.10</b>	Fenêtre liste des étudiants	63
<b>Figure 4.11</b>	Fenêtre sujet	64
<b>Figure 4.12</b>	Fenêtre d'ajout un d'sujet	64
<b>Figure 4.13</b>	Fenêtre questions	65
<b>Figure 4.14</b>	Fenêtres Résultat obtenue lors du scan d'une copie	65
<b>Figure 4.15</b>	Fenêtre résultats	66
<b>Figure 4.16</b>	Liste de résultat	66
<b>Figure 4.17</b>	Résultat d'un étudiant	67

## Liste des tableaux :

<b>Tableau</b>	<b>Titre</b>	<b>Page</b>
<b>Tableau 3.1</b>	Tableau de spécification des besoins	<b>41</b>
<b>Tableau 3.2</b>	Description du cas d'utilisation « S'authentifier »	<b>45</b>
<b>Tableau 3.3</b>	Description du cas d'utilisation « Ajouter un étudiant »	<b>46</b>
<b>Tableau 3.4</b>	Description du cas d'utilisation « Créer un sujet ».	<b>47</b>
<b>Tableau 3.5</b>	Description du cas d'utilisation « créer les questions du sujet »	<b>48</b>
<b>Tableau 3.6</b>	Description du cas d'utilisation « scanner les copie ».	<b>49</b>
<b>Tableau 3.7</b>	Description du cas d'utilisation « consulter la liste des notes des étudiants »	<b>50</b>

## Liste des abréviation et acronymes

**QCM** : Question à Choix Multiple.

**AMC** : Auto Multiple Choice.

**C.C.D** : Charge Coupled Device

**PPP** : Point Par Pouces

**DPI** : Dots Per Inch

**T.V** téléviseur

**Pixel** : Picture Element

**RVB** : Rouge Vert Bleu

**CPU** : Central Processing Unit

**UML** : Unified Modeling Language

**PDF** : Portable Document Format

**PNG** : Portable Network Graphics

**SQL** : Structured Query Language

# INTRODUCTION GENERALE :

Les technologies de l'information et de la communication ont été la révolution la plus importante qui a marqué ces dernières années, grâce aux applications mobiles qui sont devenues une partie indispensable des smartphones et le nombre d'utilisateurs de ces derniers est en constante augmentation.

Le nombre des étudiants aux universités et aux autres établissements d'enseignement a aussi connue une augmentation très importante, d'où leur évaluation est devenue une opération très couteuse en facteur temps, où les enseignants passent beaucoup de temps à corriger les copies des étudiants et élèves, ce qui peut être considéré comme un gaspillage de l'effort de l'enseignant, surtout, s'il existe d'autres méthodes de correction, l'exploitation de la technologie et devenue indispensable afin de résoudre ce problème.

La naissance des QCM (Question à choix multiples) daterait d'il y a à peu près 107 (1915) ans et est attribuée à un psychométricien américain qui proposait de remplacer dans certains cas les questions rédactionnelles (ouvertes) par des questions à réponse fermée, Les QCM sont un outil d'évaluation qui peut être utilisé pour une évaluation diagnostique, formative ou sommative.[9]

Les QCM peuvent tester beaucoup de connaissances en peu de temps, de même niveau cognitif que les questions rédactionnelles s'ils sont écrits selon certains critères qui en feront un outil d'évaluation des connaissances fiable et valide.

Plusieurs applications ont été présentée comme solution tel que : MOODLE [9] ou cette technologie est limité car l'étudiant doit posséder un smartphone ou un Pc et ce n'est pas le cas pour tous les étudiants, aussi et l'enseignant doit partager l'examen a tous les étudiants.

Une autre solution c'est présenté comme une application mobile nommé AMC [2] (Auto Multiple Choice), son *inconvenient* est qu'il faut toujours être connecter au Net, contrairement à la solution que nous allons proposer dans ce mémoire.

Comme une solution de ce problème nous proposons un système de correction automatique de copies d'examens au format QCM, qui est l'objet de ce mémoire, d'où Il est organisé en quatre chapitres. Le premier chapitre présente des généralités sur le traitement d'image. Le deuxième chapitre présente les étapes de la méthode proposée pour la correction automatique de copies d'examens au format QCM. Nous y détaillons, notamment, les formules mathématiques et les algorithmes qui sont utilisés. Le troisième chapitre sera consacré à la phase d'analyse et de conception qui précède le quatrième chapitre et la dernière partie de ce mémoire qui a été consacré à la partie pratique dénommé implémentation et réalisation.

# CHAPITRE 01

## GENERALITES SUR LE TRAITEMENT D'IMAGES

### 1. GENERALITES SUR LE TRAITEMENT D'IMAGES

#### 1.1 Introduction

#### 1.2 Définition de l'image :

#### 1.3 Image numérique :

#### 1.4 Résolution et/ou de Définition pour une photo numérique :

#### 1.5 Acquisition d'une image :

#### 1.6 Caractéristiques d'une image numérique :

##### 1.6.1 Dimension :

##### 1.6.2 Résolution :

##### 1.6.3 Bruit :

##### 1.6.4 L'histogramme :

##### 1.6.5 Luminance :

##### 1.6.6 Contraste :

#### 1.7 Les types d'image :

##### 1.7.1 Image binaire (noire ou blanc) :

##### 1.7.2 Image à niveaux de gris :

##### 1.7.3 Image en couleurs (RVB) :

#### 1.8 Système de traitement d'images :

##### 1.8.1 Pré-traitement d'image :

##### 1.8.2 Modification d'histogramme :

##### 1.8.3 Réduction du bruit par filtrage :

#### 1.9 Extraction de contours :

#### 1.10 Segmentation en régions :

##### 1.10.1 Accroissement de région :

##### 1.10.2 Méthodes du type "split and merge" :

#### 1.11 CONCLUSION :

# 1. GENERALITES SUR LE TRAITEMENT D'IMAGES

## 1.1 Introduction :

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années, on désigne par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations, de ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes, dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différentes techniques connues dans ce domaine.

## 1.2 Définition de l'image :

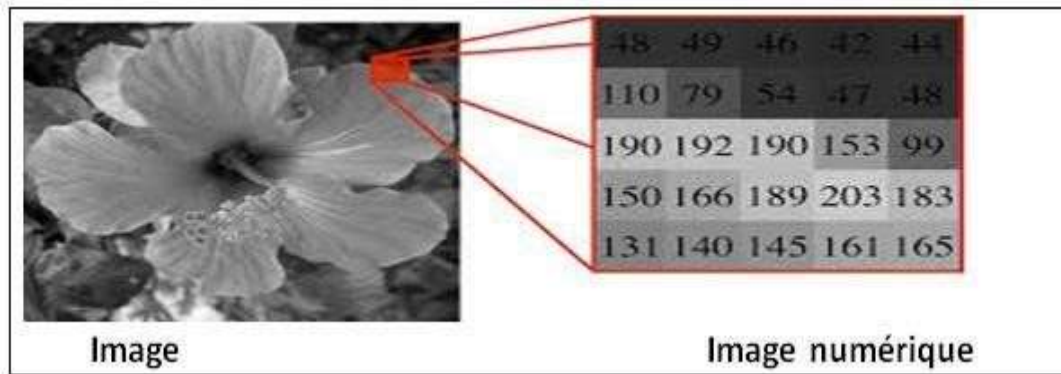
Une image est une représentation visuelle, voire mentale, de quelque chose (objet, être vivant ou concept). Elle peut être naturelle (ombre, reflet) ou artificielle (sculpture, peinture, photographie), visuelle ou non, tangible ou conceptuelle (métaphore), elle peut entretenir un rapport de ressemblance directe avec son modèle ou au contraire y être liée par un rapport plus symbolique.[12]

## 1.3 Image numérique :

Qu'est-ce qu'une image numérique ? Une image numérique est une image (dessin, icône, photographie...) créée, traitée, stockée sous forme binaire (suite de 0 et de 1)

Lorsqu'on agrandit une image numérique, on voit que celle-ci est composée d'un ensemble de "points", appelés pixels. (Figure 1.1)

Qu'est-ce qu'un pixel ? Le pixel (abréviation venant de l'anglais : **P**icture **E**lement) est l'élément de base d'une image ou d'un écran, c'est-à-dire un point. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions (largeur et hauteur) constituant l'image.[11]



**Figure 1.1 :** Sous image de taille 5 x 5

#### 1.4 Résolution et/ou de Définition pour une photo numérique :

Les termes Résolution et Définition sont souvent confondus dans le langage de la photo numérique, Ils sont liés à la notion de pixels.

La résolution d'une image est le nombre de pixels contenus dans l'image par unité de longueur, Elle s'exprime le plus souvent en ppp (point par pouces) ou en dpi (dots per inch), parfois en point par cm.

La résolution définit la netteté et la qualité d'une image. Plus la résolution est grande (c'est-à-dire plus il y a de pixels dans une longueur de 1 pouce), plus que l'image est précise dans les détails.

La définition est le nombre de points (ou pixels) que comporte une image numérique en largeur et en hauteur (le nombre de colonnes et nombre de lignes), On parle aussi de Taille en pixels.

**Exemple :** une image dont la définition est 1600x1200 correspond à une image de 1600 pixels en largeur et 1200 pixels en hauteur.

#### 1.5 Acquisition d'une image :

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système.

Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales.

- Les caméras numériques,
- Les scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la caméra, de la T.V. ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier.

L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (pixel), mais aussi l'intercorrélation de deux éléments voisins : plus cette intercorrélation est faible, meilleure est l'image.

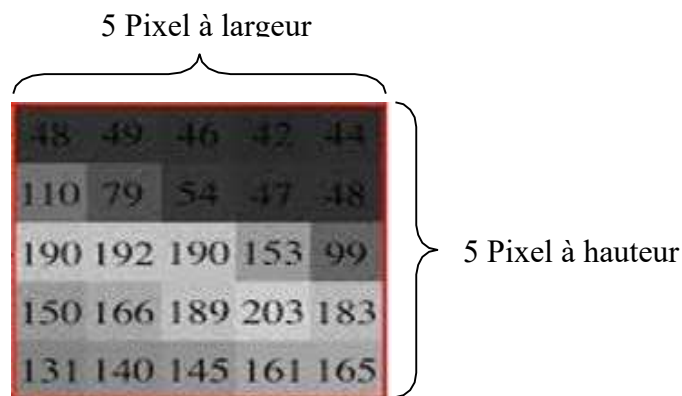
### 1.6 Caractéristiques d'une image numérique :

Une image est caractérisée par :

#### 1.6.1 Dimension :

C'est aussi l'unité utilisée pour spécifier l'image Par exemple : une image de 640x480 correspond à une image présentant 640 pixels en largeur et 480 en hauteur. (Voir Figure 1.2)

**Exemple :**



***Figure 1.2 :*** Extrait d'une image de taille 5 X 5 Pixels

#### 1.6.2 Résolution :

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les écrans, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre).

On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un écran ; plus grand est ce nombre, meilleure est la résolution.

### 1.6.3 Bruit :

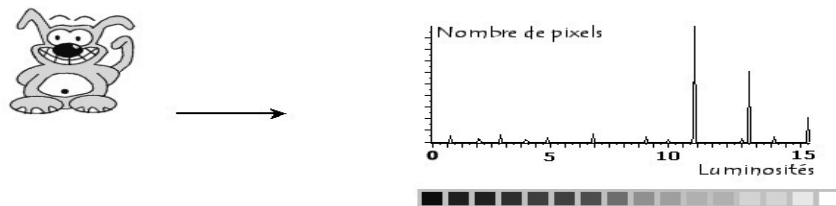
Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

### 1.6.4 L'histogramme :

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.



**Figure 1.3 :** Histogramme et palette associés à une image

### 1.6.5 Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- ✓ **Des images lumineuses (brillantes) ;**
- ✓ **Un bon contraste :** il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir : ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- ✓ **L'absence de parasites.**

**1.6.6**      **Contraste :**

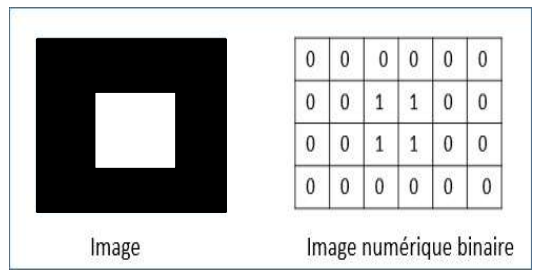
C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport [4] :

$$C = \frac{L1-L2}{L1+L} \dots\dots\dots(1.1)$$

**1.7 Les types d'image :**

**1.7.1** Image binaire (noire ou blanc) :

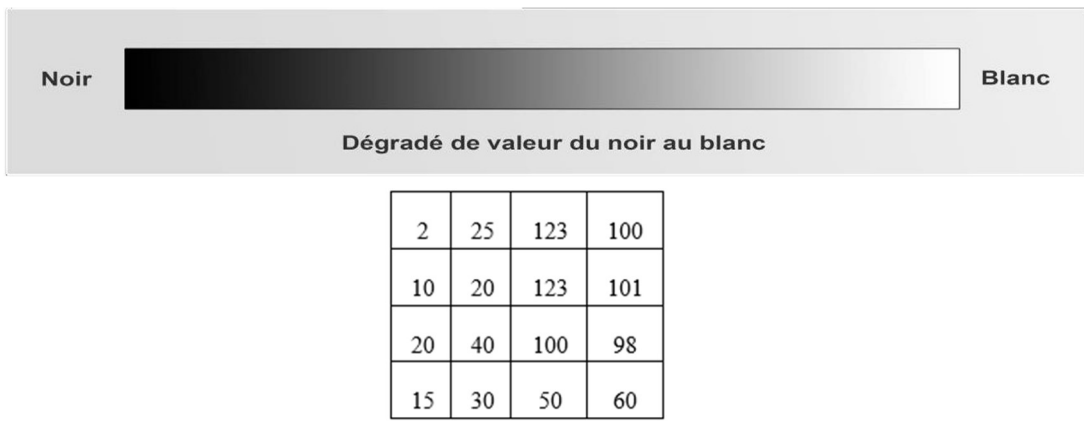
Image binaire contient avec seulement 2 valeurs possibles, chaque pixel a valeur que 0 pour le noir ou 1 pour le blanc.



**Figure 1.4 :** image binaire avec un tableau de valeurs correspondant

**1.7.2**      **Image à niveaux de gris :**

La couleur du pixel d'image à niveau de gris peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires donc les valeurs comprises entre 0 et 255. (0 est le noir)

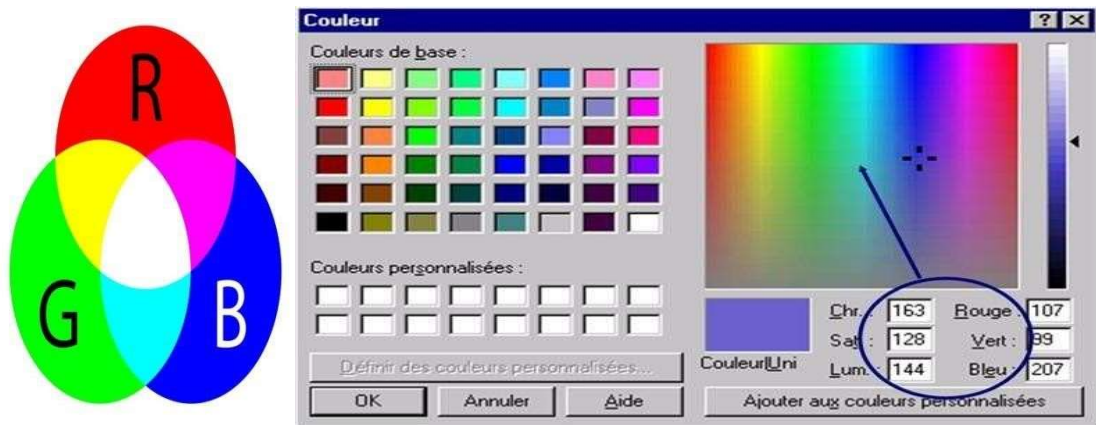


**Figure 1.5 :** Image numérique de niveau de gris

### 1.7.3 Image en couleurs (RVB) :

Dans image en couleurs chaque pixel possède une couleur décrite par la quantité de ces 3 composants : rouge(R), vert(V) et bleu (B), chacune de ces couleurs est codée sur l'intervalle [0, 255].) La combinaison de ces trois couleurs donne un point lumineux (un pixel) d'une certaine couleur.

Donc Le système RVB est une des façons de décrire une couleur en informatique. Par exemple : Le triplet {255, 255, 255} donnera du blanc, {255, 0, 0} un rouge pur, {100, 100, 100} un gris, etc. Le premier nombre donne la composante rouge, le deuxième la composante verte et le dernier la composante bleue.

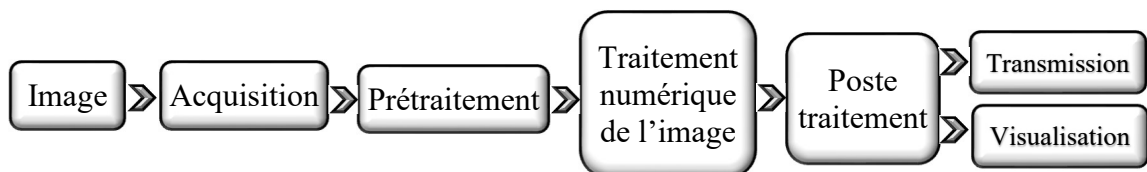


{0, 12,55}	{17, 25, 14}	{255, 255, 255}
{78, 85, 55}	{14,7, 47}	{250, 30, 5}
{47, 47, 47}	{8, 52, 16}	{0, 2, 5}

**Figure 1.6 :** Image numérique en couleur

### 1.8 Système de traitement d'images :

Un système de traitement numérique d'images est composé de :



**Figure 1.7 :** Schéma d'un système de traitement d'images

### 1.8.1 Pré-traitement d'image :

Cette phase a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de sa segmentation.

Le temps de traitement (temps CPU) est très important. C'est un facteur décisif, il doit être le plus petit possible. Ceci implique que les opérateurs doivent être locaux, c'est-à-dire qu'ils doivent intervenir sur un nombre limité de pixels et essentiellement sur des pixels dans le voisinage proche du pixel courant.

Les techniques de pré-traitements les plus courantes qu'on va présenter sont :

- La modification d'histogramme,
- La réduction du bruit par filtrage,

### 1.8.2 Modification d'histogramme :

On cherche à améliorer l'image en appliquant une transformation ponctuelle d'intensité ; à tout pixel  $a_s(x,y)$  on associe une intensité par la transformation  $T$  croissante de façon à conserver des contrastes entre régions (claire sur fond sombre, apparaît plus claire après transformation). Le pixel après transformation est le pixel  $a's(x,y)$  :  $a's(x,y)=T(a_s(x,y))$

On distingue deux types de modification d'histogramme :

- Expansion de dynamique
- Egalisation d'histogramme

### 1.8.3 Réduction du bruit par filtrage :

Le bruit peut provenir de différentes causes :

- Environnement lors de l'acquisition,
- Qualité du capteur,
- Qualité de l'échantillonnage.

Plusieurs filtres sont utilisés pour la réduction de bruit, ils sont divisés en deux catégories :

- Filtres linéaires stationnaires,
- Filtres non linéaires stationnaires.

La première catégorie comprend tous les opérateurs pouvant exprimer leur résultat comme une combinaison linéaire des niveaux de gris d'un voisinage de l'image. Ces filtres possèdent des caractéristiques spectrales, on parle ainsi de filtre passe-bas (l'image devient floue) ou de filtre passe-haut (les contours ressortent).

Voici quelques exemples de filtres réducteurs de bruit :

- Filtre moyenneur
- Filtre gaussien

➤ Filtre exponentiel de Shen

Le principal inconvénient des filtres linéaire est que la réduction de bruit s'accompagne d'un étalement des transitions entre régions. Ce problème peut être surmonté par l'utilisation des filtres non linéaires.

Le domaine de filtrage non linéaire est extrêmement vaste, Les opérateurs les plus courants en pré-traitement d'images sont :

- Les filtres d'ordre,
- Les filtres homomorphiques,
- Les filtres morphologiques.

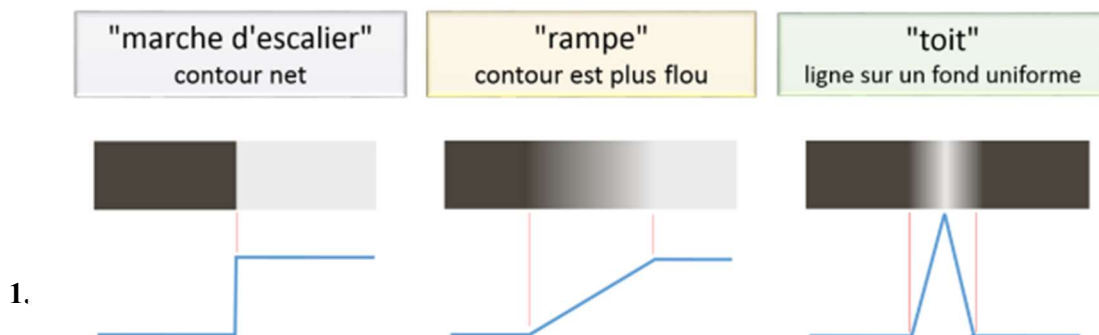
**1.9 Extraction de contours :**

Le but de la détection de contour d'une image est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse. La détection des contours d'une image réduit de manière significative la qualité des données et élimine les informations. Les primitives à extraire présentent les lignes de contrastes séparant des régions de niveaux de gris différents et relativement homogènes, ou bien des régions de texture différentes. En pratique, il s'agit de reconnaître les zones de transition et de localiser au mieux la frontière entre les régions.

Un contour est donc défini comme une zone de l'image où l'intensité des pixels change subitement, cette interruption dans l'image est le passage d'un niveau de gris à un autre, de manière plus ou moins rapide.

On distingue trois types de contours simples :

- **Marche d'escalier** : le contour est net (contour idéal).
- **Rampe** : le contour est plus flou.
- **Toit** : il s'agit d'une ligne sur un fond uniforme.



**Figure 1.8** : Différents type de contours

Fondamentalement, la segmentation est un processus qui consiste à découper une image en régions connexes présentant une homogénéité selon un certain critère, comme par exemple la couleur. L'union de ces régions doit redonner l'image initiale

On regroupe généralement les algorithmes de segmentation en trois grandes classes :

- Segmentation basée sur les pixels
- Segmentation basée sur les régions
- Segmentation basée sur les contours

La première catégorie travaille sur des histogrammes de l'image. Par seuillage, clustering ou clustering flou, l'algorithme construit des classes de couleurs qui sont ensuite projetées sur l'image. La segmentation est implicite puisqu'on suppose que chaque cluster de l'histogramme correspond à une région dans l'image. En pratique, ce n'est pas forcément le cas et il faut séparer les régions de l'image qui sont disjointes bien qu'appartenant au même cluster de couleur. Ces algorithmes sont assez proches des algorithmes de réduction de couleur.

La deuxième catégorie correspond aux algorithmes d'accroissement ou de découpage de région. L'accroissement de région est une méthode Bottom-up : on part d'un ensemble de petites régions uniformes dans l'image (de la taille d'un ou de quelques pixels) et on regroupe les régions adjacentes de même couleur jusqu'à ce qu'aucun regroupement ne soit plus possible.

Le découpage de région est le pendant top-down des méthodes d'accroissement : on part de l'image entière que l'on va subdiviser récursivement en plus petites régions tant que ces régions ne seront pas suffisamment homogènes.

La troisième catégorie s'intéresse aux contours des objets dans l'image. La plupart de ces algorithmes sont locaux, c'est à dire fonctionnent au niveau du pixel.

Les algorithmes que nous allons présenter sont :

- Accroissement de région fonctionnant au niveau du pixel.
- Split and merge classique.

#### **1.10.1** Accroissement de région :

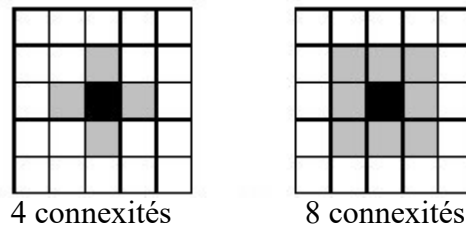
Les méthodes d'accroissement de régions sont les méthodes de segmentation les plus simples.

Le principe est basé sur une approche bottom-up : l'algorithme part de petits éléments de l'image qu'il va tenter de regrouper en éléments plus importants.

Nous présentons ici la version de base de l'algorithme d'accroissement de région qui fonctionne en agrégeant des pixels. Supposons une région de couleur homogène  $R$ .

Initialement,  $R = 1$  pixel. On va étendre la région  $R$  en incluant les pixels situés sur la frontière et dont la couleur est proche de celle de  $R$  (la variation de couleur est inférieure à un

seuil □ caractéristique de ce type d'algorithmes). En répétant cette procédure jusqu'à ce qu'il n'y ai plus de pixels de couleur assez proche sur la frontière, on obtient une région de couleur homogène maximale autour du pixel de départ. La région initiale "gonfle" en absorbant des pixels de la frontière, jusqu'à stabilité par rapport à la propriété d'homogénéité. Afin d'assurer un critère de connexité, on utilise les voisins en 4-connexités pour l'accroissement de région et la 8-connexités pour l'accroissement de contour (cas de détection de contours).



**Figure 1.9 :** Voisinage d'un pixel

Cette méthode présente deux limitations :

- Les régions obtenues dépendent fortement des pixels d'amorçage choisis et de l'ordre dans lequel les pixels de la frontière sont examinés.
- Le résultat final est très sensible à la valeur du seuil.

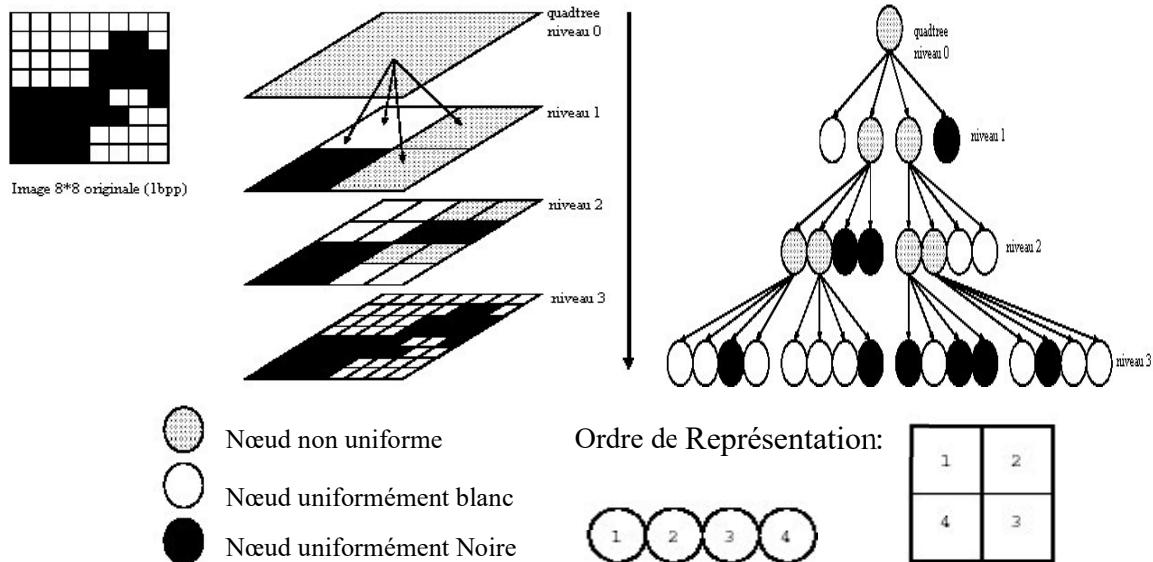
Cet algorithme fait partie de la classe d'algorithmes de segmentation dit "locaux". L'opération élémentaire consiste à manipuler des pixels adjacents et l'algorithme n'a aucune vision globale du résultat qu'il obtient.

#### 1.10.2 Méthodes du type "split and merge" :

L'algorithme "split and merge" a été présenté la première fois en 1974 par Pavlidis et Horowitz. Cet algorithme s'apparente dans son principe à l'algorithme d'accroissement de région que nous venons de présenter. La différence principale provient de la nature des régions élémentaires agrégées. Dans l'algorithme "split and merge", les régions agrégées proviennent d'une première phase (split) de traitement de l'image qui construit de manière récursive des régions carrées de taille variable mais homogènes.

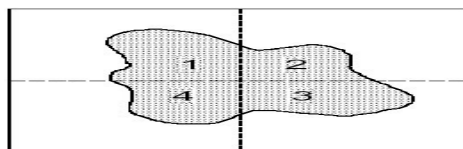
- **Split** : La méthode de découpage de l'image utilisée dans cet algorithme est basée sur la notion de "quadtree". Cette structure de données est un arbre quaternaire qui permet de stocker l'image à plusieurs niveaux de résolution. On part d'une région initiale qui est l'image tout entière. Si cette image vérifie un critère d'homogénéité de couleur, l'algorithme s'arrête. Sinon, on découpe cette région en quatre parties de même taille et on lance la procédure récursivement dans ces quatre parties. La région initiale va être stockée comme un nœud dans

Un graphe et les sous parties comme des fils de ce nœud. La figure (figure 1.10) montre une image en noir et blanc 8x8 et le découpage correspondant à chaque niveau, ainsi que la structure d'arbre associée à ce découpage.



**Figure 1.10 :** Découpage par quadtree d'une image 8x8. Traitement avec seuil=100%

➤ **Merge :** La procédure de découpage décrite précédemment aboutit à un nombre de régions trop élevé. La cause fondamentale de cette sur-segmentation est que l'algorithme découpe les régions de manière arbitraire. Il se peut qu'il coupe de cette façon une zone homogène en deux ou quatre parties (Figure 1.10).



**Figure 1.11 :** Problème de découpage arbitraire de régions dans la phase de "split"

Les parties 1,2,3 et 4 de la (Figure 11) par exemple, appartiennent à des branches différentes du quadtree. Elles sont donc considérées comme des régions différentes bien que leur couleur soit identique.

La solution, qui correspond à la phase "merge" de l'algorithme, est de procéder à une fusion de régions après le découpage. L'implémentation la plus simple de cette fusion cherche tous les couples de régions adjacentes dans l'arbre issu du découpage et cherche à les fusionner si leur couleur est assez proche.

### **1.11 CONCLUSION :**

Le domaine de traitement d'images a connu ces dernières années un intérêt de plus en plus croissant et suscite de par ses différentes applications la convoitise des chercheurs de différents horizons et domaines. On désigne par traitement d'images l'ensemble des techniques permettant de modifier une image dans le but de l'améliorer ou d'en extraire des informations, Nous avons introduit dans ce chapitre les notions de bases qui servent de fondement à la compréhension de différentes techniques de traitement d'images et quelques applications de domaine de traitement d'image. Plusieurs méthodes classiques de traitement ont été proposés dans la littérature, nous avons présenté quelques-unes qui nous semble les plus courantes et essentielles dans le processus du traitement et analyse d'image.

# CHAPITRE 2 :

## CORRECTION AUTOMATIQUE DE COPIES AU FORMAT QCM

- 2.1 Introduction :
- 2.2 Acquisition d'images :
- 2.3 Filtrage :
- 2.4 Binarisation :
- 2.5 Correction de l'inclinaison :
- 2.6 Localisation des coins bas :
- 2.7 Extraction de la zone d'intérêt :
- 2.8 Détection des Cercles :
- 2.9 Lecture des choix :
- 2.10 Détection du matricule :
- 2.11 Conclusion : **Error! Bookmark not defined.**

## 2. CORRECTION AUTOMATIQUE DE COPIES AU FORMAT QCM

### 2.1 Introduction :

Dans ce chapitre nous allons entamer le mécanisme d'acquisition puis le traitement d'image de la copie d'examen, un examen qui doit être formulé sous forme d'un QCM.

Un QCM (Un questionnaire à choix multiples) se compose d'un ensemble de questions structurées. Cet ensemble vise un objectif global diagnostique, formatif, ou sommatif par rapport au contenu d'une formation.

Ce chapitre donc présente une méthode de correction automatique de copies d'examen au format QCM en utilisant le traitement d'images. Le modèle de copie Qcm doit être bien choisi afin d'accomplir cette tâche (voir la figure 2.1).

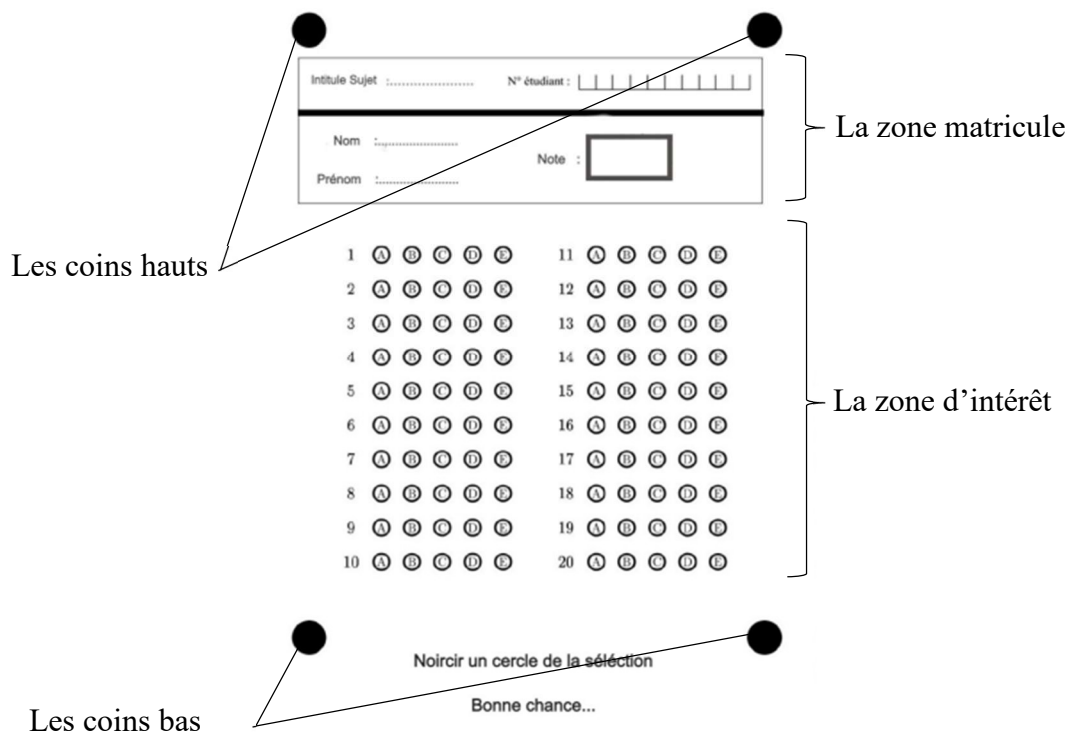


Figure 2.1: Le model de QCM utilisé avec différents blocs

Tout d'abord la copie d'examen est capturée par un smartphone doté d'un système d'exploitation Android, d'une version 7 au maximum dite Nougat (voir figure 2.2) pour obtenir une image numérique, puis un certain nombre de traitements est appliqué à cette image selon l'organigramme (voir figure 2.3).



**Figure 2.2:** Android 7.0 (Nougat)



**Figure 2.3:** Schéma synoptique de la méthode proposée.

## 2.2 Acquisition d'images :

L'image acquise est en niveaux de gris codé sous 8 bits et au format A4 de dimensions 8.27 x 11.67 pouces (21 x 27.9 cm), comme l'indique la figure 2.4.

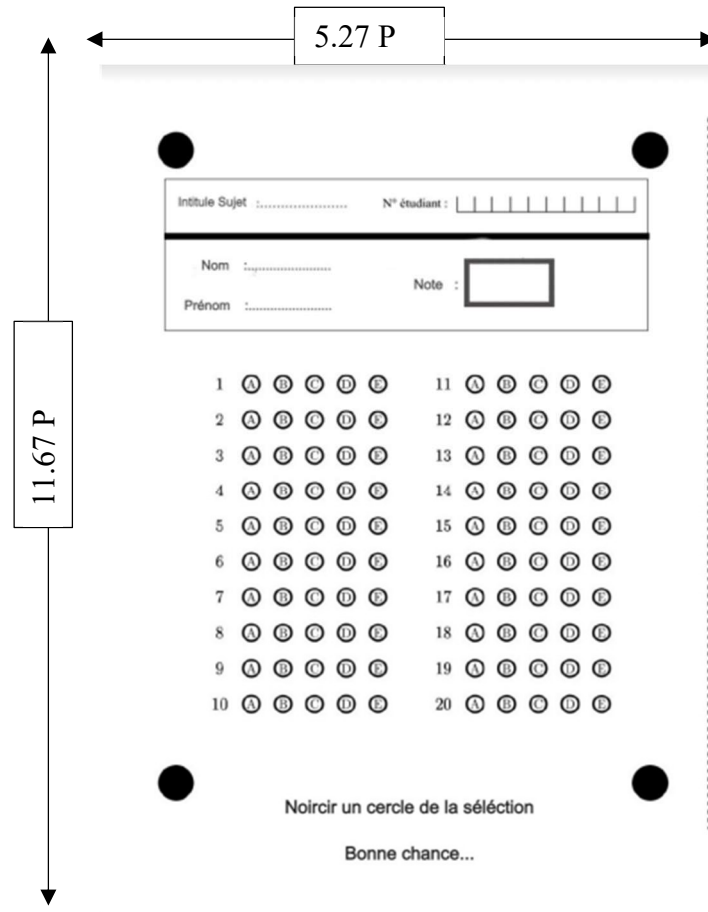


Figure 2.4: Exemple d'une image scanné et ses propriétés

## 2.3 Filtrage :

Tout d'abord un filtre gaussien de taille 7 x 7 est appliqué à l'image à niveau de gris pour réduire le bruit et garder les traits.

## 2.4 Binarisation :

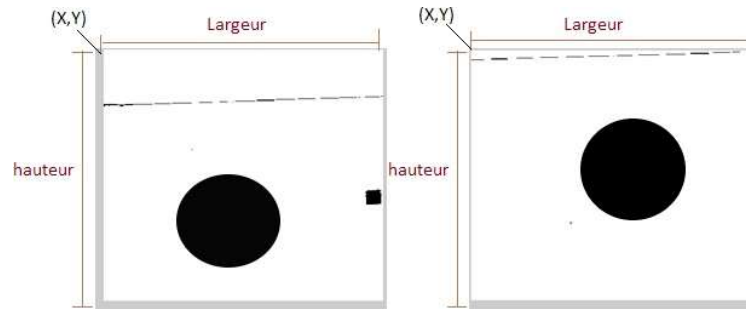
Dans cette étape, la méthode d'OTSU est utilisée pour déterminer un seuil de binarisation de l'image filtrée, Les pixels qui ont une valeur supérieure auront la valeur max égale à 255 (blanc) et ceux qui ont une valeur inférieure auront la valeur min égale à 0 (noir).

Localisation des coins hauts :

Pour localiser des coins hauts de la copie d'examen, nous proposons les étapes suivantes :

✓ **Découpage des deux parties hautes de l'image (partie gauche et droite)** (voir figure 2.5) La hauteur et la largeur des parties découpées sont fixées à :

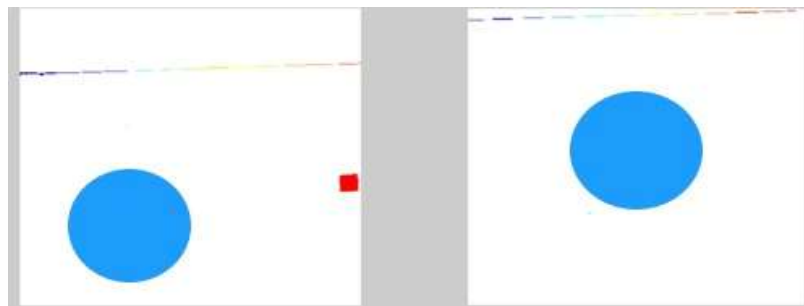
- Partie gauche (X=0, Y=0, Hauteur=360 pixels, Largeur= 380 pixels)
- Partie droite (X=2100, Y=0, Hauteur=360 pixels, Largeur= 380 pixels)



**Figure 2.5:** Découpage d'images (partie haute gauche et partie haute droite)

✓ L'étiquetage :

Cette opération affecte un numéro d'identification à chaque composante connexe d'une image (séparer les différents objets). (Voir figure 2.6)



**Figure 2.6:** Résultat de l'étiquetage

✓ Calcul du nombre de pixels de chaque étiquette :

Le but de cette opération est de trouver l'objet ayant un nombre de pixels maximal.

Le nombre des pixels blanc est calculé pour chaque objet, pour prendre l'objet ayant un plus grand nombre de pixels.

Donc l'objet qui a le plus grand nombre de pixels blanc représente un coin.

✓ Calcul des coordonnées de centre de gravité des coins :

Les équations suivantes sont utilisées pour calculer les coordonnées du centre de gravité  $x_g$  et  $y_g$  :

$$m_{00} = \sum_{X=0}^{X-1} \sum_{Y=0}^{N-1} f(X, Y) \dots\dots\dots(2.1)$$

$$X_g = \frac{\sum_{X=0}^{X-1} \sum_{Y=0}^{N-1} f(X, Y) \cdot X}{m_{00}} \dots\dots\dots (2.2)$$

$$Y_g = \frac{\sum_{X=0}^{X-1} \sum_{Y=0}^{N-1} f(X, Y) \cdot Y}{m_{00}} \dots\dots\dots(2.3)$$

$f(X, Y)$ : l'image d'entrée.

$m_{00}$  : Le nombre de pixel de l'objet.

$X_g$ : abscisse de centre de gravité.

$Y_g$  : ordonnée de centre de gravité.



**Figure 2.7:** le Centre de gravité des coins de l'image découpée.

Les coordonnées du centre de gravité de coin gauche dans l'image initiale est exacte par contre les coordonnées droites ce n'est pas réelles pour cela on va passer à l'étape suivant.

✓ Calcul les nouvelles coordonnées de centre de gravité de coin droit :

Nous avons utilisé les équations suivantes :

$$\left[ \begin{array}{l} x_{g1} = X_{g1} \text{ ancien} \dots\dots\dots (2.4) \\ y_{g1} = Y_{g1} \text{ ancien} \dots\dots\dots (2.5) \end{array} \right.$$

$$\left[ \begin{array}{l} x_{g2} = m - (L - X_{g2} \text{ ancien}) \dots\dots\dots (2.6) \\ y_{g2} = Y_{g2} \text{ ancien} \dots\dots\dots (2.7) \end{array} \right.$$

$x_{g1}$  ,  $y_{g1}$  ,  $x_{g2}$  ,  $y_{g2}$  : les nouvelles coordonnées.

$X_{g1} \text{ ancien}$ ,  $Y_{g1} \text{ ancien}$ ,  $X_{g2} \text{ ancien}$ ,  $Y_{g2} \text{ ancien}$  : les anciens coordonnées.

m : largeur du l'image initial (2480 pixel).

L : largeur de partie coupé (380 pixels).

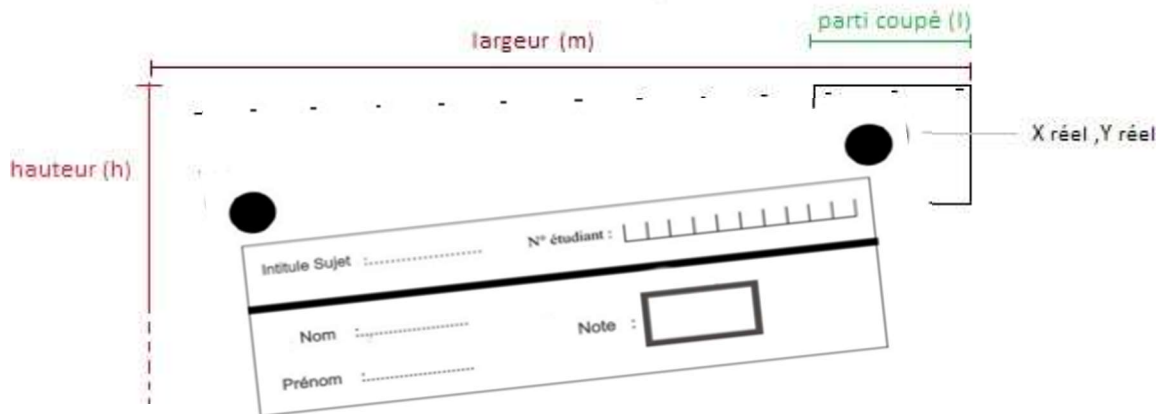


Figure 2.8: Les nouvelles coordonnées de centre de gravité de coin droit.

### 2.5 Correction de l'inclinaison :

Tout d'abord il faut calculer l'angle de rotation en utilisant les deux coordonnées des centres de gravité trouvés précédemment :

- { Coin gauche (xg1, yg1).
- { Coin droit (xg2, yg2).

L'équation utilisée est la suivante :

$$angle = \text{atan} \left( \frac{yg2 - yg1}{xg} \right) \dots \dots \dots (2.8)$$

La rotation est faite autour de l'origine (0,0).



Figure 2.9: Rotation d'une image autour de son origine.



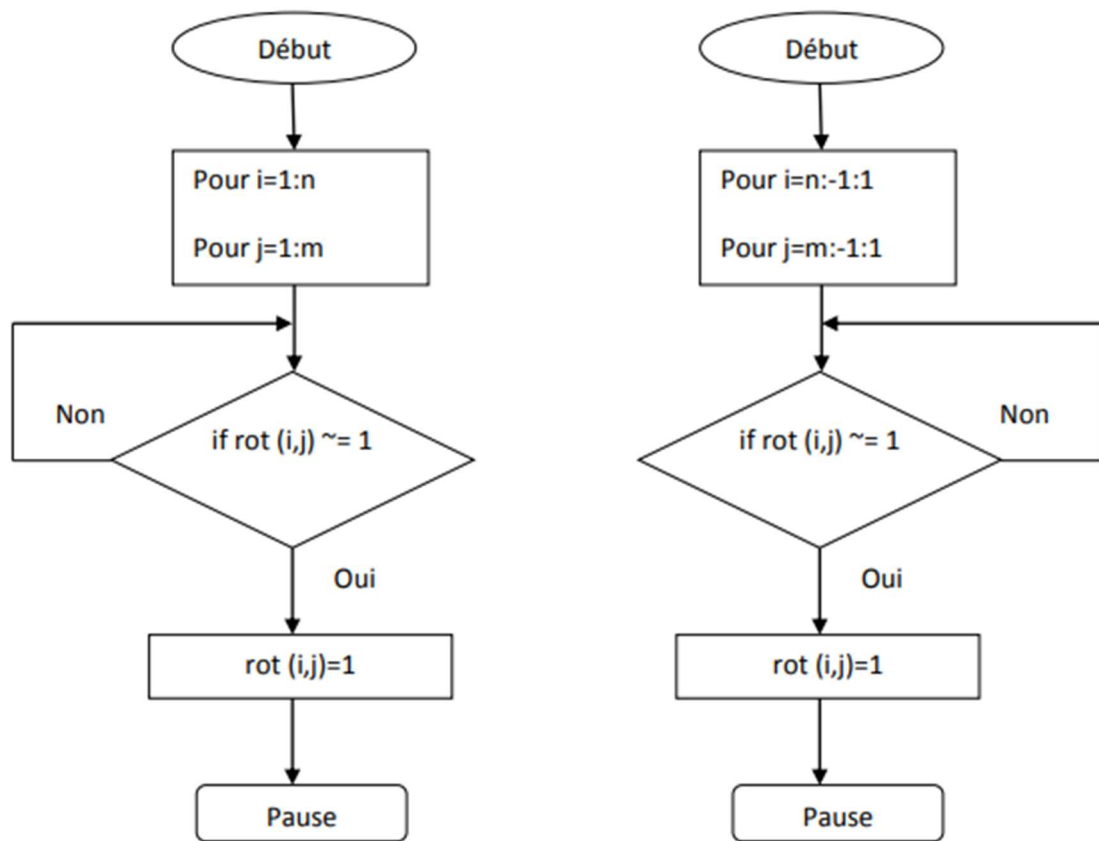
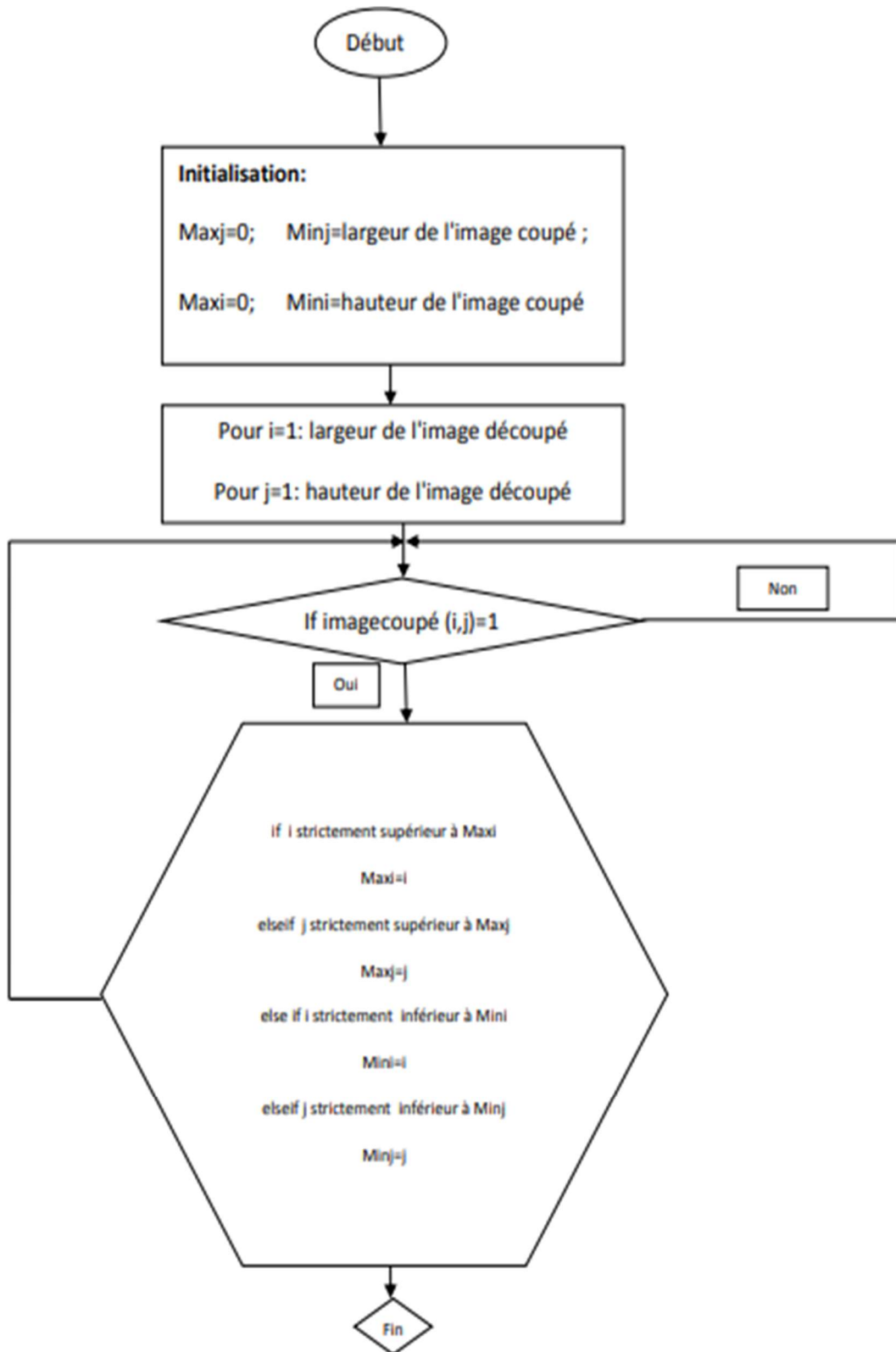


Figure 2.11: Organigramme de la méthode proposé (élimination des pixels noire)

Trouver les positions de la ligne détectée :

Le but est de trouver (Mini, Maxi) dans le sens des lignes et (Minj, Maxj) dans le sens des colonnes, la méthode utilisée est implémentée suivant l'organigramme :



**Figure 2.12** : organigramme de la méthode de détection de la ligne

Trouver les positions (Maxi, Maxj, Minj, Mini) dans l'image initial :

Les équations utilisées sont les suivantes :

$$\text{Mini} = \text{Mini} + 850 \dots \dots \dots (2.9)$$

$$\text{Maxi} = \text{Maxi} + 850 \dots \dots \dots (2.10)$$

$$\text{Minj} = \text{Minj} \dots \dots \dots (2.11)$$

$$\text{Minj} = \text{Minj} \dots \dots \dots (2.12)$$

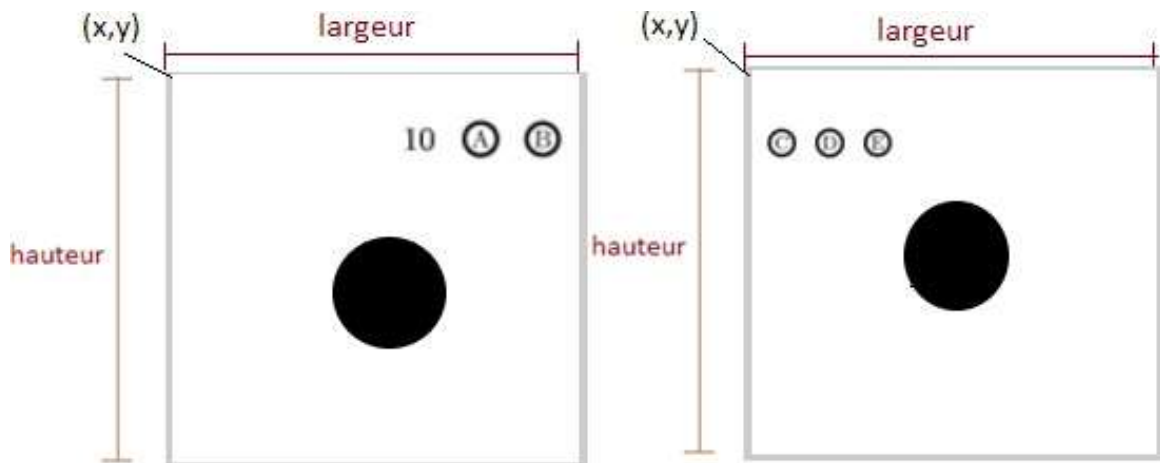
### 2.6 Localisation des coins bas :

On suit les étapes suivantes :

✓ **Découper les deux parties bas de l'image (partie gauche et droit)** (voir figure 2.13)

La hauteur et la largeur de la partie découpé sont fixées.

- Partie gauche (X=0, Y=3137, Hauteur=360 pixels, Largeur= 380 pixels)
- Partie droite (X=2100, Y=3137, Hauteur=360 pixels, Largeur= 380 pixels)



**Figure 2.13:** Image découpé (partie Bas gauche et partie bas droit)

- ✓ L'étiquetage.
- ✓ Calculer le nombre des pixels pour chaque étiquète et trouver les coins.
- ✓ Calculer les coordonnées de centre de gravité des coins

On utilise les équations (2.1), (2.2), (2.3) pour calculer le centre de gravité : Coin bas gauche (xg3, yg3) et coin bas droite (xg4, yg4).

✓ **Calculer les nouvelles coordonnées du centre de gravité des coins bas :**

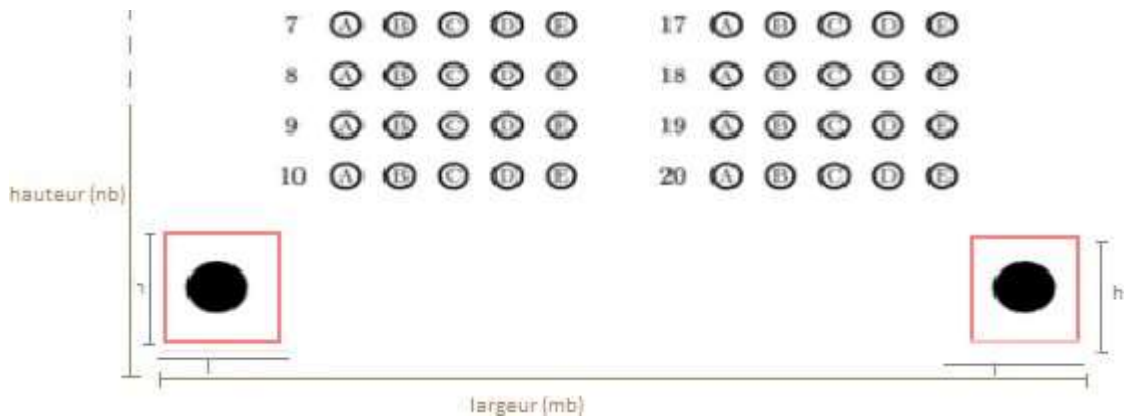
On utilise les équations suivantes :

$$\left\{ \begin{array}{l} x_{g3} = X_{g3 \text{ ancien}} \dots\dots\dots (2.13) \\ y_{g3} = nb - (h - Y_{g3 \text{ ancien}}) \dots\dots\dots (2.14) \\ x_{g4} = mb - (l - X_{g4 \text{ ancien}}) \dots\dots\dots (2.15) \\ y_{g4} = nb - (h - Y_{g4 \text{ ancien}}) \dots\dots\dots (2.16) \end{array} \right.$$

$x_{g3}$  ,  $y_{g3}$   $x_{g4}$  ,  $y_{g4}$ : sont les nouvelles coordonnées.

$X_{g3 \text{ ancien}}$ ,  $Y_{g3 \text{ ancien}}$   $X_{g4 \text{ ancien}}$ ,  $Y_{g4 \text{ ancien}}$  : sont les anciennes coordonnées.

- $mb$  : largeur du l'image initial (2480 pixel).
- $nb$  : hauteur du l'image initial (3507 pixel).
- $l$ : largeur de la partie découpé (380 pixel).
- $h$ : hauteur de la partie découpé (360 pixel).



**Figure 2.14:** les nouvelles coordonnées du centre de gravité des coins bas.

**2.7 Extraction de la zone d'intérêt :**

A partir des coordonnées trouvés précédemment, la zone intérêt est découpées.

En utilisant :  $Maxi$ ,  $x_{g3}$ ,  $y_{g3}$ ,  $x_{g4}$ , la position de la partie découpée (voir figure2.15) est comme suite :

$$\left\{ \begin{array}{l} x=x_{g3} \\ y=Maxi \\ Largeur =x_{g4}-x_{g3} \\ Hauteur= y_{g3}-Maxi \end{array} \right.$$



### 2.8 Détection des Cercles :

La méthode utilisée pour la détection des cercles est basée sur l'histogramme de projection.

#### **Histogrammes de projection :**

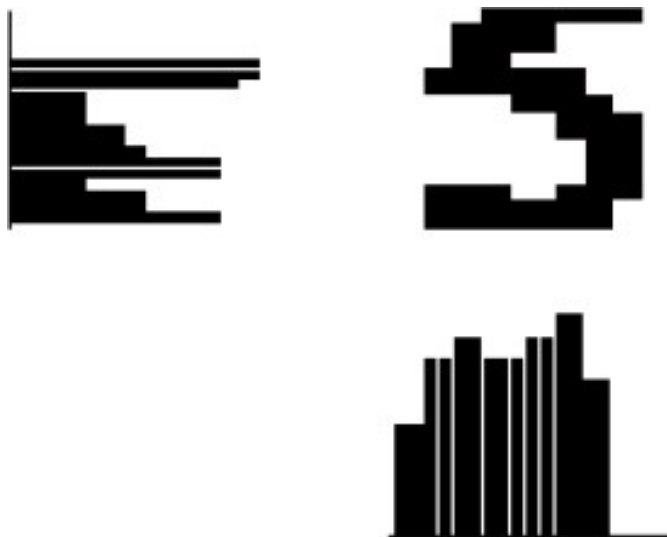
L'histogramme de projection est depuis très longtemps utilisé dans le domaine de la reconnaissance des formes. Son principe est de sommer le nombre de pixels noirs ou bien blanc de chacune des lignes (respectivement des colonnes) de l'image binaire de la forme (figure 2.16).

La projection d'histogrammes a été introduite en 1956 par Glauberman, cette technique est principalement utilisée pour la segmentation des caractères, des mots et de lignes de texte.

La projection horizontale donne le nombre de pixel de chaque ligne ainsi que la projection verticale donne le nombre de pixels de chaque colonne.

Toutefois, la projection d'histogramme est très sensible à la rotation et dans une certaine mesure, la variation dans le style d'écriture.

En outre, des informations importantes sur le caractère forme semble être perdu.



**Figure 2.16:** Histogramme horizontale et verticale du chiffre 5.

## Chapitre 2 - correction automatique de copies au format QCM

**Remarque :** L'image de la zone intérêt utilisée doit être binarisée (1 pour les pixels blancs et 0 pour les pixels noirs).

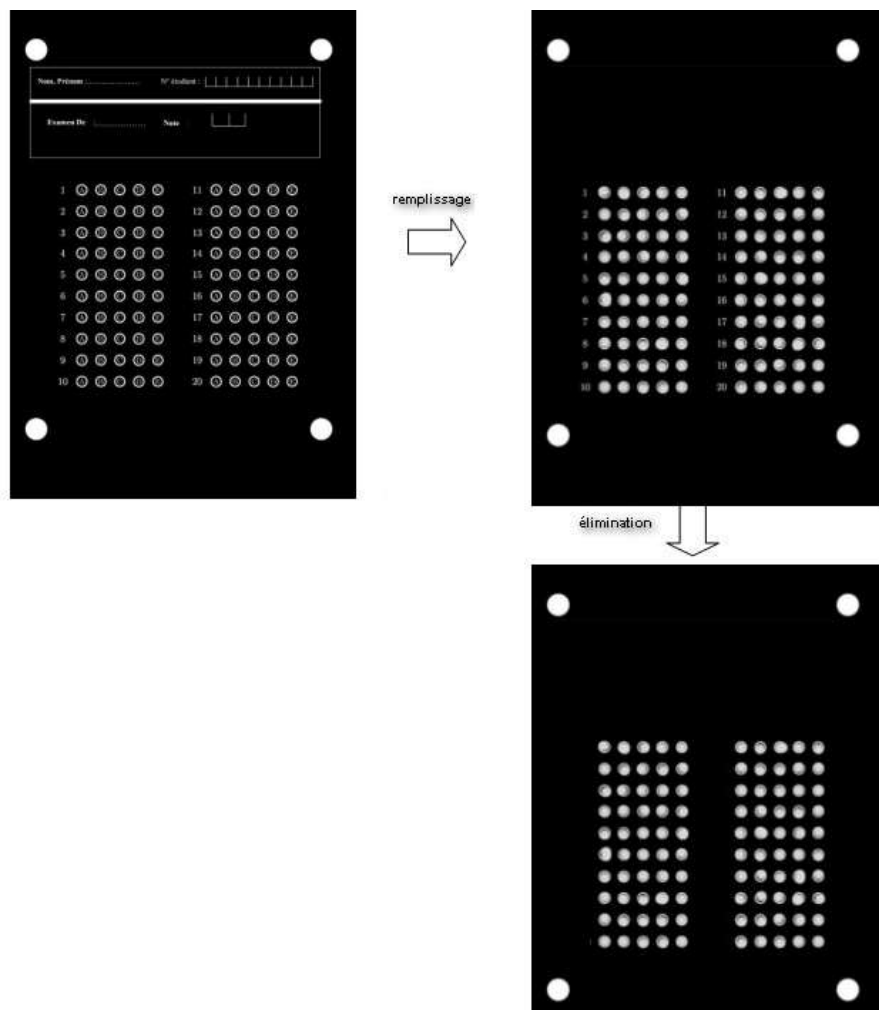
Avant de calculer l'histogramme de projection verticale et horizontale il faut garder uniquement les cercles, Pour cela on doit procéder comme suite :

- ✓ Remplissage des régions :

Le but de cette opération est de remplir les cercles par des pixels (voir la figure 2.17)

- ✓ Élimination des régions.

Les régions qui ont le nombre de pixels inférieur à 1600 sont éliminées (voir la figure 2.17)



**Figure 2.17:** remplissage et élimination de régions.

- ✓ Calculer d'histogramme de projection verticale et horizontale :

Les équations utilisées sont :

$$Hist\_v(y) = \sum_{x=1}^n f(x, y) \dots \dots \dots (2.17)$$

$$Hist\_h(y) = \sum_{y=1}^m f(x, y) \dots \dots \dots (2.18)$$

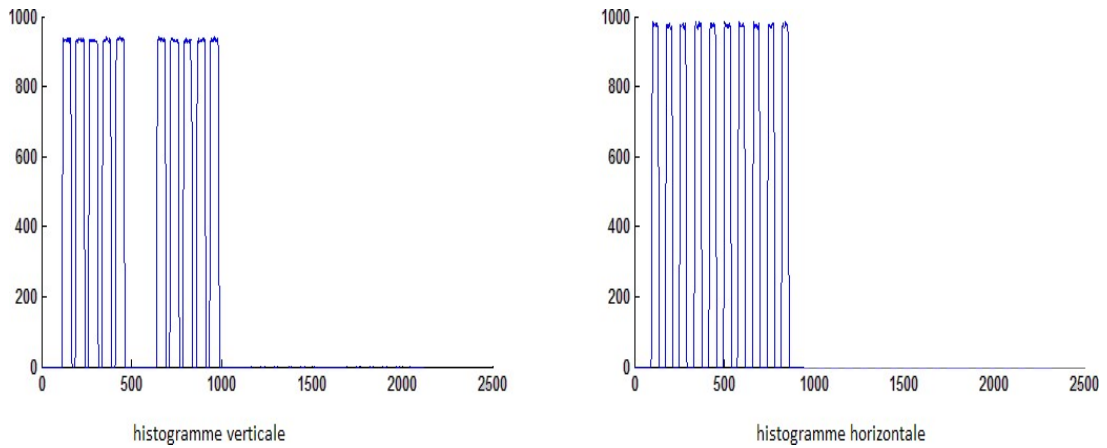
$f(x, y)$  : image initiale

$Hist\_v(y)$  : Histogramme vertical.

$Hist\_h(y)$  : Histogramme horizontal.

$n$  : nombre de ligne.

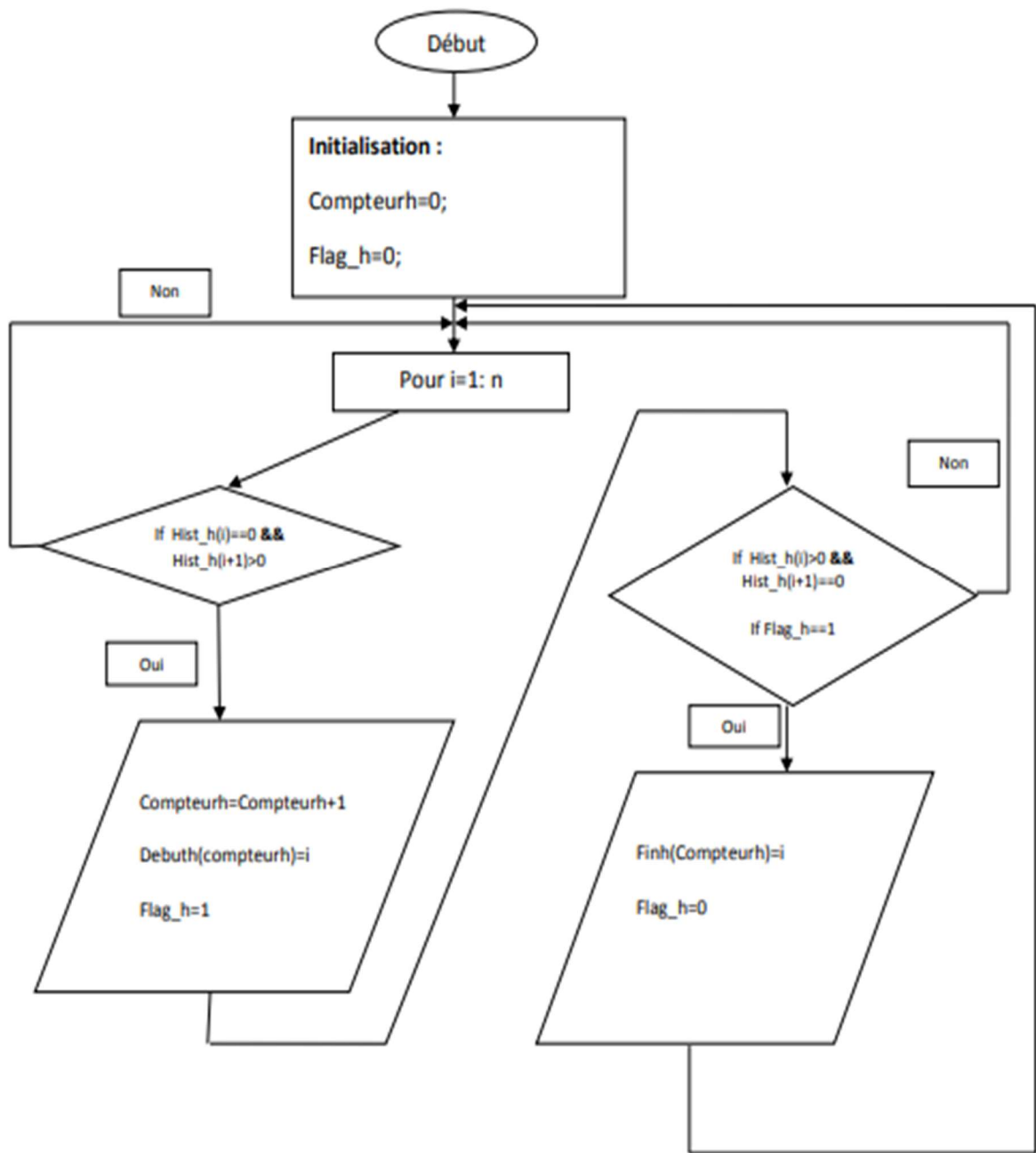
$m$  : nombre de colonne.



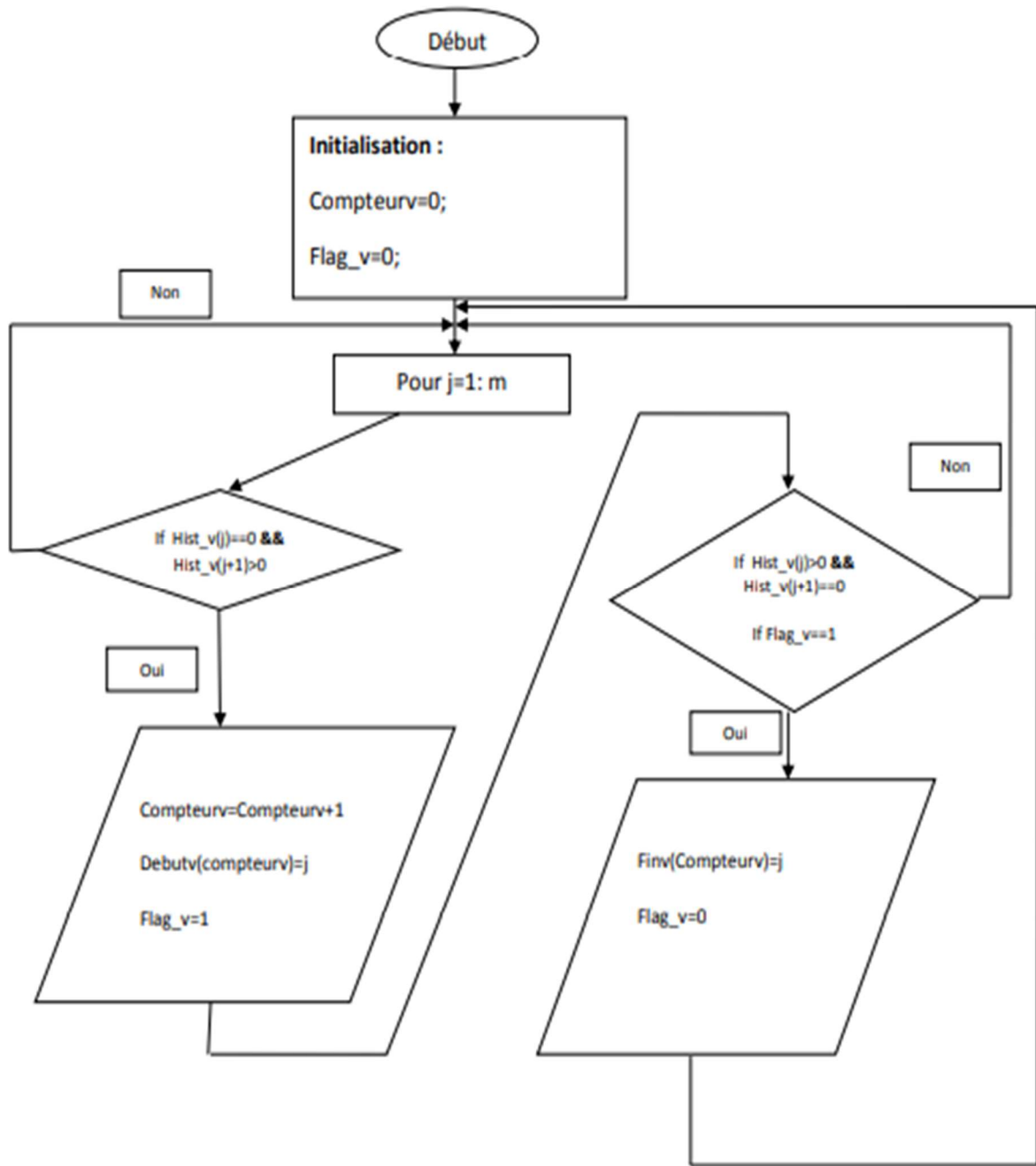
**Figure 2.18:** Histogramme verticale et horizontale de l'image de la zone d'intérêt.

✓ Localisation des cercles :

A partir des histogrammes trouvés, on calcule le début de chaque front montant et la fin de chaque front descendant, la méthode est implémentée suivant les deux organigrammes des figures 2.19 et 2.20 :



**Figure 2.19** : organigramme de localisation horizontale des cercles

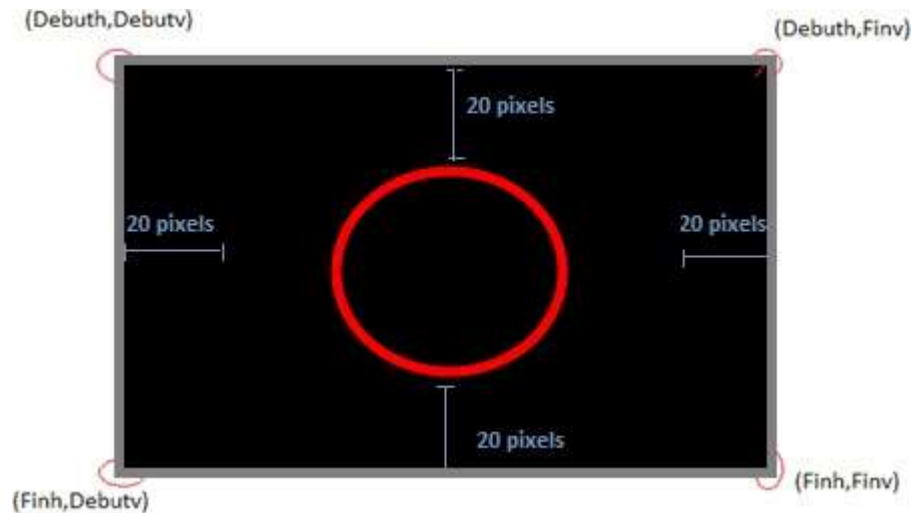


**Figure 2.20 :** organigramme de localisation verticale des cercles

### 2.9 Lecture des choix :

Les étapes suivantes sont appliquées à la zone d'intérêt :

- Supprimer les pixels isolés.
- Trouver la zone de choix :
- On Change les positions (Débuth, Finh, Débutv, Finv) pour positionner au centre de chaque cercle ; on ajoute 20 pixels pour les coordonnées (Débuth, Debutv) et on retranche 20 pixels pour (Finh, Finv) selon la figure 2.21).



**Figure 2.21:** la zone de choix en rouge.

- ✓ Extraction de chaque zone de choix.



**Figure 2.22:** Extraction de la zone de choix. (a) zone choisie. (b) zone vide.

- Appliquer un étiquetage pour chaque zone extraite.
- Lecture des choix.

On calcule le nombre d'étiquètes dans chaque zone, si le nombre d'étiquète est supérieur à zéro la zone prendre la valeur 1 sinon prendre la valeur 0.

Donc Chaque cercle prendra une valeur soit 1 soit 0.

Les résultats sont implémentés dans une matrice  $C$  de  $n$  lignes et de  $m$  colonnes.

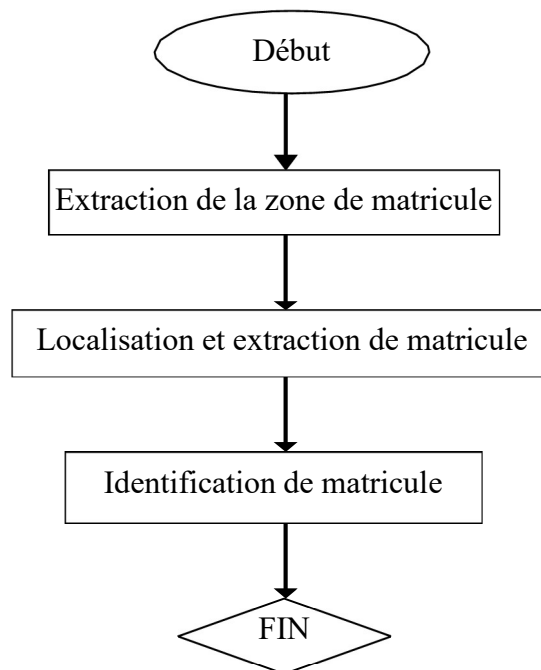
### 2.10 Détection du matricule :

Le matricule utilisé pour la détection est composé des chiffres et lettres, une partie fixe et une partie qui change, L'étudiant doit écrire son matricule en noircissant les huit segments. (Voir figure 2.23).



**Figure 2.23:** Modèle de matricule utilisé

La méthode proposée pour la détection est la suivante



**Figure 2.24:** Schéma synoptique de la méthode de détection du matricule

#### ✓ Extraction de la zone du matricule :

A partir des coordonnées trouvées précédemment la zone de matricule est découpée en utilisant les résultats des équations (2.10), (2.13), (2.15)

La position de la partie découpée est comme suite :

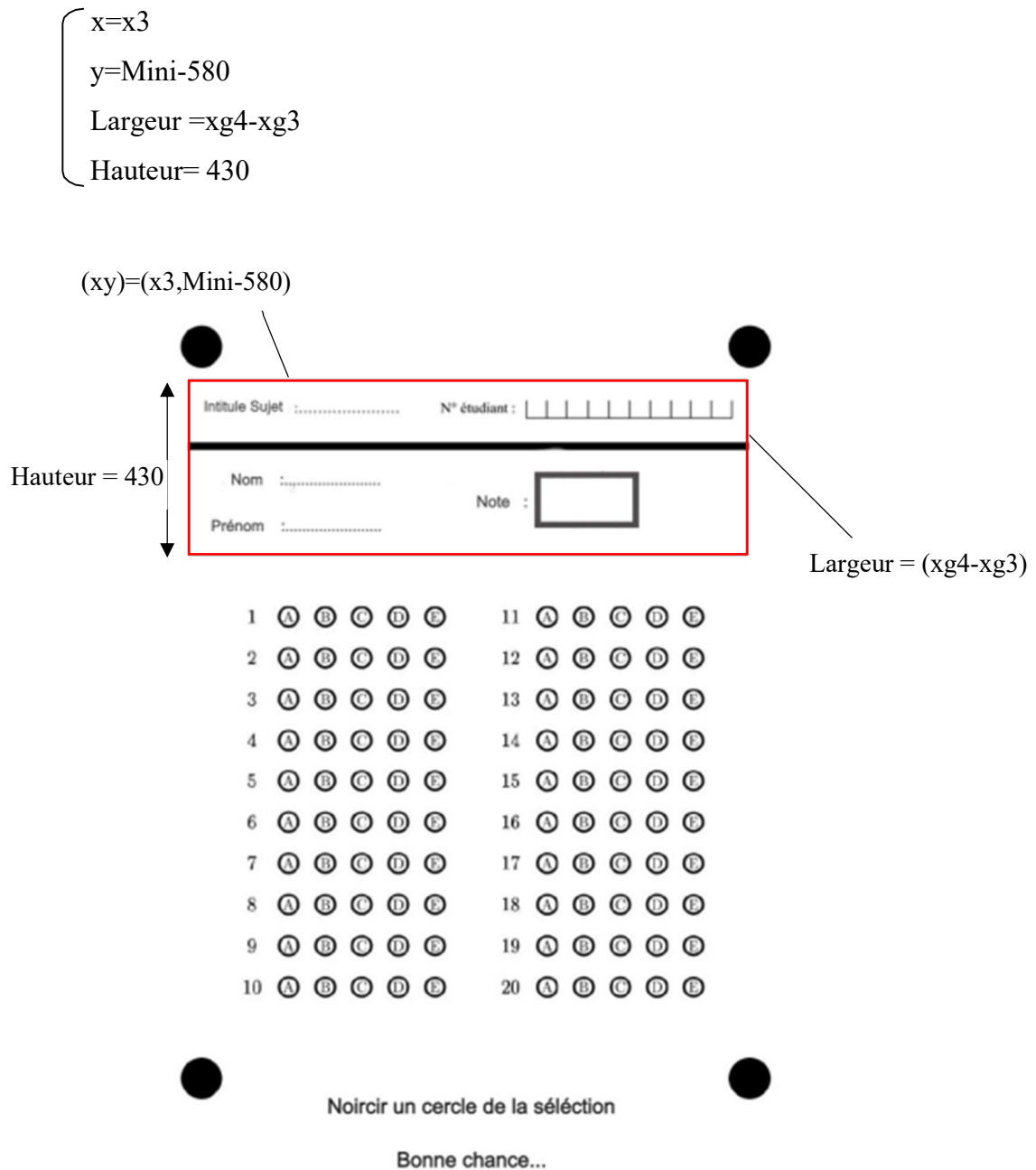


Figure 2.25: Extraction de la zone de matricule

- ✓ Localisation et extraction de matricule : on utilise les opérations suivantes :
  - **Élimination des régions :**  
Les régions qui ont le nombre de pixels inférieur à 2000 sont éliminer (voir la figure 2.26)  
Le but est de garder que les grands objets.



Figure 2.26: Résultat d'opération d'élimination des régions.

○ Calculer l'histogramme de projection verticale et horizontale. Le but de cette opération est de trouver la position de matricule.

En utilisant les équations (2.17) et (2.18) pour calculer l'histogramme vertical et horizontal, Ensuite nous utilisons la méthode précédemment implémentée (figure 2.19 et 2.20) pour trouver les coordonnées (Debut<sub>h</sub>, Fin<sub>h</sub>, Debut<sub>v</sub>, Fin<sub>v</sub>).

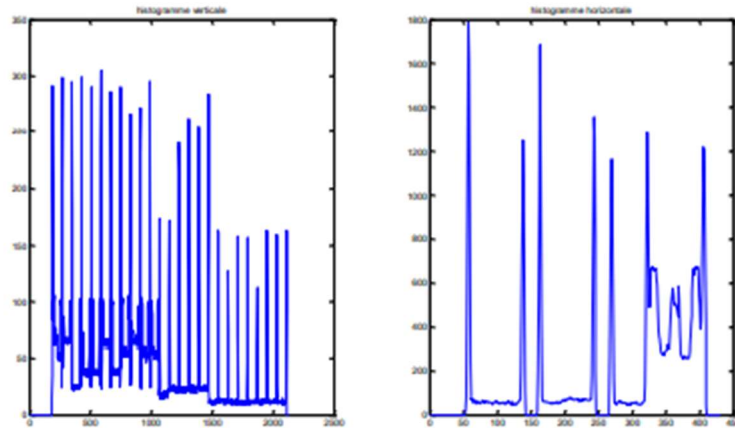


Figure 2.27: histogramme verticale et horizontale de la zone de matricule.

\* La position de matricule est indiquée par la figure 2.28 comme suite :



Figure 2.28: localisation et extraction de matricule.

➤ Identification de matricule :

Les étapes adoptées pour réaliser une telle opération, sont comme suit :

- Erosion.
- Elimination de régions : les régions ayant un nombre de pixels inférieur à 100 sont éliminées.
- Dilatation.



**Figure 2.29:** Résultat d'opération d'érosion suivie de dilatation.

- Normalisation de matricule : la taille utilisée est 50 x 400 pixels.
- La corrélation entre l'image de la base de données et l'image de matricule extraite.

**2.11 Conclusion :**

Sans ce chapitre nous avons présenté les différentes étapes de traitement d'image qui représente dans notre cas une image d'un examen QCM scannée par un smartphone, ainsi que la correction automatique de ce dernier, commençant par :

Acquisition de l'image, le filtrage, la binarisation, localisation des coins supérieurs et inférieurs, la correction de l'inclinaison localisation et extraction de la zone d'intérêt afin de lire les choix, en terminant par la détection du matricule de l'étudiant.

# CHAPITRE 03

## ANALYSE ET CONCEPTION

### 3.1 Introduction

### 3.2 Analyse

#### 3.2.1 Identification des acteurs :

#### 3.2.2 Les acteurs de notre système

### 3.3 Diagramme de contexte

### 3.4 Spécification des besoins

#### 3.4.1 Les besoins fonctionnels :

#### 3.4.2 Les besoins non fonctionnels :

### 3.5 Spécification des cas d'utilisation

#### 3.5.1 Définition

#### 3.5.2 Relations entre cas d'utilisation

#### 3.5.3 Diagrammes de cas d'utilisation

### 3.6 Description textuelle des cas d'utilisation

### 3.7 Conception

#### 3.7.1 Diagramme de séquence

#### 3.7.2 Diagramme de classe :

#### 3.7.3 Conception de la base de données

#### 3.7.4 Passage du diagramme de classe au modèle relationnel de données

#### 3.7.5 **Le modèle relationnel de données**

### 3.8 Conclusion

## 3. ANALYSE ET CONCEPTION

### 3.1 Introduction

Dans ce chapitre, nous commençons par présenter notre projet, puis on entamera la phase « Analyse » qui mettra en évidence l'acteur principal dans l'application et ses besoins. Suivi d'une phase « conception », s'appuyant sur la phase précédente, qui donnera la modélisation des objectifs à atteindre.

Notre travail sera consacré au développement d'une application mobile, qui permet la correction automatique des copies d'examen sous forme QCM. Parmi les tâches à réaliser :

- Création d'un compte administrateur
- Inscription ou authentification
- Introduire les noms des étudiants
- Création du sujet.
- Ajouter des questions
- Etablir un corrigé type
- Proposer un barème de notation à chaque question
- Scanner les copies des étudiants avec la caméra du Smartphone
- Afficher les résultats sous forme d'une image format PNG.
- Afficher les résultats de chaque étudiant sous format PDF.

De ce fait, notre démarche se basera sur le langage UML, qui permettra de bien représenter les aspects fonctionnels, statiques et dynamiques de notre projet par la série des diagrammes qu'il offre.

### 3.2 Analyse

#### 3.2.1 Identification des acteurs :

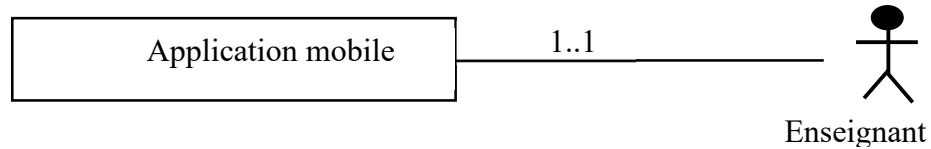
Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système, il peut consulter et/ou modifier directement l'état du système, en émettant et/ou recevant des messages susceptibles d'être porteurs de données.

#### 3.2.2 Les acteurs de notre système

Un seul acteur qui intervient dans notre système est : **l'enseignant** qui est une personne inscrite et validée par lui-même, chargée de préparer l'examen et de proclamer les résultats obtenus.

### 3.3 Diagramme de contexte

Le diagramme de contexte est un modèle conceptuel de flux, Il permet de bien délimiter le champ d'étude et de spécifier le nombre d'instances d'acteurs connectés au système.



**Figure 3.1** : Diagramme de contexte.

### 3.4 Spécification des besoins

On trouve deux types de besoin : des besoins fonctionnels et d'autres non fonctionnels.

#### 3.4.1 Les besoins fonctionnels :

Un besoin est l'ensemble logique d'opérations qu'un programme ou une partie d'un programme doit exécuter pour répondre aux attentes d'un acteur.

Le tableau (3.1) résume les besoins fonctionnels de notre application :

<i>Acteur</i>	<i>Besoins</i>
<i>Enseignant</i>	<p><b>B1</b> : S'authentifier.</p> <p><b>B2</b> : Ajouter un module.</p> <p><b>B3</b> : Créer un sujet.</p> <p><b>B4</b> : Créer les questions du sujet.</p> <p><b>B5</b> : Créer les choix de réponses.</p> <p><b>B6</b> : Créer un corrigé type.</p> <p><b>B7</b> : Fixer un barème de notation pour chaque question.</p> <p><b>B8</b> : Créer la liste des étudiants.</p> <p><b>B8</b> : Modifier la liste des étudiant.</p> <p><b>B10</b> : scanner les copies d'examen (sujet).</p> <p><b>B11</b> : Etablir une liste des notes des étudiants sous format image</p> <p><b>B12</b> : Consulter la liste des notes des étudiants.</p> <p><b>B13</b> : Sauvegarder la liste sous format PDF</p>

**Tableau 3.1:** Tableau de spécification des besoins.

### 3.4.2 Les besoins non fonctionnels :

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception.

Les besoins non fonctionnels de notre application se résument à :

- SGBD de type :SQL light
- Systèmes d'exploitation : Android 7 (Nougat) ou version antérieure
- Langages de programmation : java
- La disponibilité : le système constitue le cœur de l'activité, il est nécessaire qu'il soit en service lors de sa demande sans aucune connexion à internet ou à un réseau mobile sauf s'il a besoin d'impression.
- Sécurité : c'est primordial pour l'application, étant donné qu'il gère des données confidentielles, l'authentification est indispensable.
- Performance : fluidité de l'application et temps de réponse immédiat, d'où l'enseignant récupère le résultat en temps réel.

## 3.5 Spécification des cas d'utilisation

### 3.5.1 Définition

Un cas d'utilisation est une manière spécifique d'utiliser un système où les acteurs sont à l'extérieur ; Ils modélisent tout ce qui interagit avec lui.

Un cas d'utilisation réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie.

### 3.5.2 Relations entre cas d'utilisation

Pour clarifier un diagramme, UML permet d'établir des relations entre les cas d'utilisation.

Il existe principalement :

- **La relation d'inclusion** : Un cas A est inclus dans un cas B si le comportement décrit par le cas A est inclus dans le comportement du cas B : on dit alors que le cas B dépend de A.
- **La relation d'extension** : Si le comportement de B peut être étendu par le comportement de A, on dit alors que A étend B. Une extension est souvent soumise à condition.

**La relation de généralisation** : Un cas A est une généralisation d'un cas B si B est un cas particulier de A.

### 3.5.3 Diagrammes de cas d'utilisation

Nous allons décrire les diagrammes de cas d'utilisation selon les acteurs :

Diagramme de cas d'utilisation « acteur : enseignant » (Voir figure III.2).

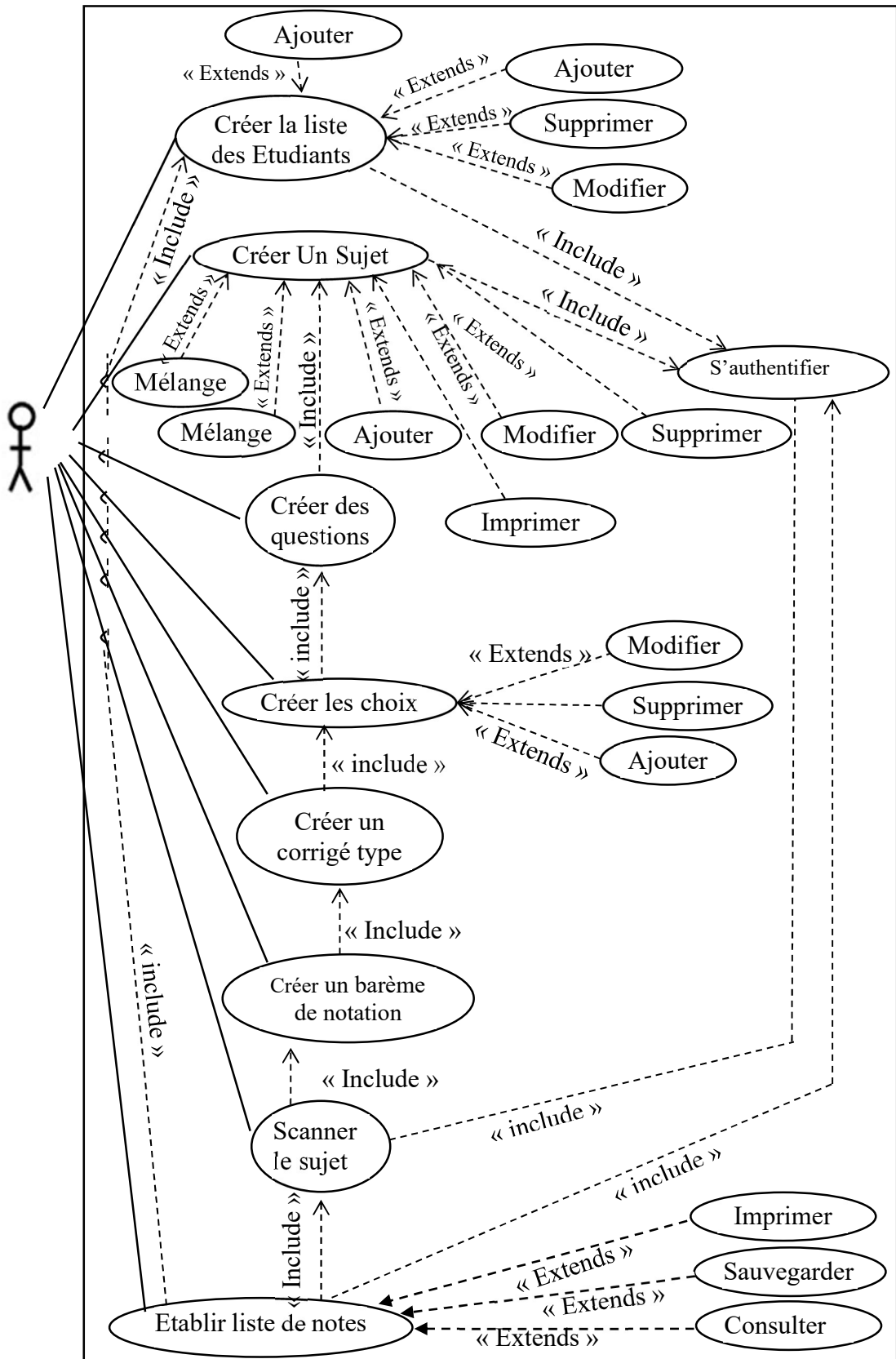


Figure 3.2: Diagramme détaillé du cas d'utilisation « correction automatique ».

### 3.6 Description textuelle des cas d'utilisation

Bien que de nombreux diagrammes d'UML permettent de décrire un cas, il est recommandé de rédiger une description textuelle car c'est une forme souple qui convient dans bien des situations. Une description textuelle couramment utilisée se compose de deux parties :

- ✓ La première partie permet d'identifier le cas. Elle doit contenir :
  - Le nom du cas.
  - Un résumé de son objectif.
  - Les acteurs impliqués (principaux et secondaires).
- ✓ La deuxième partie contient la description du fonctionnement du cas sous la forme d'une séquence de messages échangés entre les acteurs et le système. Elle contient une **séquence nominale** qui correspond au fonctionnement nominal du cas.

Cette séquence se développe en trois points :

- **Les préconditions** : Elles indiquent dans quel état est le système avant que se déroule la séquence.
- L'enchaînement des messages.
- **Les postconditions** : Elles indiquent dans quel état se trouve le système après le déroulement de la séquence nominale.

Comme on peut y avoir :

- **Une séquence alternative** : diverge de la séquence nominale (c'est un embranchement dans une séquence nominale).
- **Une séquence d'exception** : intervient quand une erreur se produit (l'enchaînement nominal s'interrompt, sans retour à la séquence nominale). [3]

**Nous décrivons les cas suivants :**

- ✓ S'authentifier.
- ✓ Ajouter un étudiant
- ✓ Créer un sujet.
- ✓ Créer les réponses
- ✓ Scanner les copies en support papier
- ✓ Etablir la liste de notes des étudiants

<i>Description du cas d'utilisation : « S'authentifier »</i>
<p><b>Identification</b></p> <p>Nom du cas : « S'authentifier ».</p> <p>Résumé : Permettre à l'utilisateur d'accéder à son espace.</p> <p>Acteur principal : L'enseignant.</p>
<p><b>Pré-conditions</b></p> <p>L'utilisateur possède un compte créé préalablement sinon il doit créer un.</p>
<p><b>Enchaînement nominal</b></p> <p>Le système demande à l'utilisateur de saisir le « nom d'utilisateur » et d'introduire « le mot de passe ».</p> <p>L'utilisateur remplit les champs « nom d'utilisateur » et « Mot de passe » puis Valide.</p> <p>Le système affiche à l'utilisateur son espace.</p>
<p><b>Enchaînement alternatif</b></p> <p><b>A1 : nom d'utilisateur ou mot de passe saisi incorrect (informations incorrectes).</b></p> <p>L'enchaînement peut démarrer après le point 2 de l'enchaînement nominal :</p> <p>4. Le système affiche une erreur d'identification.</p> <p>L'enchaînement nominal reprend au point 1.</p>
<p><b>Post-conditions :</b></p> <p>L'utilisateur accède à son espace.</p>

**Tableau 3.2:** Description du cas d'utilisation « S'authentifier ».

<b><i>Description du cas d'utilisation : « Ajouter un étudiant »</i></b>
<p><b>Identification</b></p> <p><b>Nom du cas :</b> « Ajouter un étudiant ».</p> <p><b>Résumé :</b> Permettre à l'utilisateur d'ajouter des étudiants</p> <p><b>Acteur principal :</b> L'enseignant.</p>
<p><b>Pré-conditions :</b></p> <ul style="list-style-type: none"><li>• L'utilisateur doit s'authentifier.</li><li>• L'utilisateur accède au volet « Etudiant ».</li></ul> <p><b>Enchaînement nominal :</b></p> <ol style="list-style-type: none"><li>1. Le système affiche un formulaire pour saisir les informations de l'étudiant tel que son matricule, son nom et son prénom.</li><li>2. L'utilisateur saisit les informations</li><li>3. L'utilisateur valide.</li><li>4. Le système affiche un message de confirmation de l'opération.</li></ol> <p><b>Enchaînement Alternatif</b></p> <p><b>A1 Un champ obligatoire non rempli.</b></p> <p>L'enchaînement peut démarrer après le point 3 de l'enchaînement nominal :</p> <ol style="list-style-type: none"><li>4. Le système affiche un message d'erreur.</li></ol> <p>L'enchaînement nominal reprend au point 1.</p> <p><b>Post- conditions :</b></p> <p>Le système enregistre le nouveau Etudiant dans la base de données.</p>

**Tableau 3.3:** Description du cas d'utilisation « Ajouter un étudiant ».

<i>Description du cas d'utilisation : « Créer un sujet »</i>
<p><b>Identification</b></p> <ul style="list-style-type: none"><li>• Nom du cas : « Créer un sujet ».</li><li>• But : Permettre à l'utilisateur de créer un sujet.</li><li>• Acteur principal : l'enseignant.</li></ul>
<p><b>Pré-conditions</b></p> <p>L'utilisateur doit s'authentifier. L'utilisateur accède au volet « sujet ».</p> <p><b>Enchaînement nominal</b></p> <ol style="list-style-type: none"><li>1. L'utilisateur clique sur le bouton d'ajout de sujet.</li><li>2. Le système affiche le formulaire d'ajout de sujet</li><li>3. L'utilisateur saisit les informations du sujet tel que : intitulé de sujet, nombre de questions et sa note globale L'utilisateur click sur le bouton « <b>ajouter les questions</b> ».</li><li>4. Le système affiche le formulaire d'ajout de questions.</li></ol> <p><b>Post-conditions :</b></p> <p>Le système enregistre de sujet et prépare l'ajout des questions</p>

**Tableau 3.4:** Description du cas d'utilisation « Créer un sujet ».

<b>Description du cas d'utilisation : « Ajouter les questions »</b>
<p><b>Identification</b></p> <p>Nom du cas : « ajouter questions ».</p> <p>Résumé : Permettre à l'utilisateur d'ajouter les questions du sujet.</p> <p>Acteur principal : l'enseignant.</p>
<p><b>Pré-conditions</b></p> <p>L'utilisateur doit s'authentifier.</p> <p>L'utilisateur doit créer un sujet</p> <p>L'utilisateur accède au volet "ajouter questions".</p> <p><b>Enchaînement nominal</b></p> <ol style="list-style-type: none"><li>1. L'utilisateur clique sur le bouton d'ajout de questions.</li><li>2. Le système affiche le formulaire d'ajout de questions.</li><li>3. L'utilisateur remplit le champ de question ainsi les réponses proposées à chaque question (choix).</li><li>4. L'utilisateur confirme la bonne réponse puis valide.</li><li>5. Le système enregistre le formulaire.</li></ol> <p><b>Enchaînement alternatif :</b></p> <p>L'enchaînement peut démarrer après le point 4 de l'enchaînement nominal :</p> <p>Le système affiche un message d'erreur et retourne au 3</p> <p><b>Post-condition :</b></p> <p>Le système enregistre les questions du sujet.</p>

**Tableau 3.5:** Description du cas d'utilisation « créer les questions du sujet ».

<b><i>Description du cas d'utilisation : « Scanner les copies »</i></b>
<p><b>Identification</b></p> <p>Nom du cas : « Scanner les copie ».</p> <p>Résumé : Permettre à l'utilisateur de scanner les copies des étudiants afin d'avoir les résultats obtenus.</p> <p>Acteur principal : l'enseignant.</p>
<p><b>Pré-conditions</b></p> <p>L'utilisateur doit s'authentifier.</p> <p>L'utilisateur doit créer le sujet ainsi ses questions en confirmant la bonne réponse.</p> <p><b>Enchaînement nominal</b></p> <ol style="list-style-type: none"><li><b>1.</b> L'utilisateur clique sur le bouton camera.</li><li><b>2.</b> Le système affiche l'image de la copie</li><li><b>3.</b> Le système scanne la copie de l'étudiant et affiche le résultat (note obtenue).</li></ol> <p><b>Post-conditions :</b></p> <p>Le système affiche le résultat de chaque étudiant.</p>

**Tableau 3.6:** Description du cas d'utilisation « scanner les copie ».

<b>Description du cas d'utilisation : « consulter la liste de notes des étudiants »</b>
<p><b>Identification</b></p> <p>Nom du cas : « <b>Consulter la liste de notes des étudiants</b> ».</p> <p>But : Permettre à l'utilisateur de consulter la liste de notes des étudiants.</p> <p>Acteur principal : l'enseignant.</p>
<p><b>Pré-conditions</b></p> <p>L'utilisateur doit s'authentifier.</p> <p>L'utilisateur accède au volet "Etudiant".</p> <p>L'utilisateur clique sur modifier</p> <p><b>Enchaînement nominal</b></p> <ul style="list-style-type: none"> <li>• L'utilisateur accède au volet "Etudiant".</li> <li>• L'utilisateur clique sur le bouton <b>liste</b></li> <li>• Le système affiche la liste des étudiants et permet à l'utilisateur de télécharger ou imprimer les résultats des étudiants.</li> </ul>

**Tableau 3.7:** Description du cas d'utilisation « consulter la liste des notes des étudiants ».

### 3.7 Conception



Dans cette phase une nouvelle vue du modèle fait son apparition. Cette vue exprime les modules et les exécutable physiques sans aller à la réalisation concrète du système. Elle est basée sur :


- Le diagramme de séquence.
- Le diagramme de classe.

#### 3.7.1 Diagramme de séquence

L'objectif de ce type de diagramme offert par UML est de représenter les interactions entre les objets en mettant l'accent sur le classement chronologique des messages échangés.

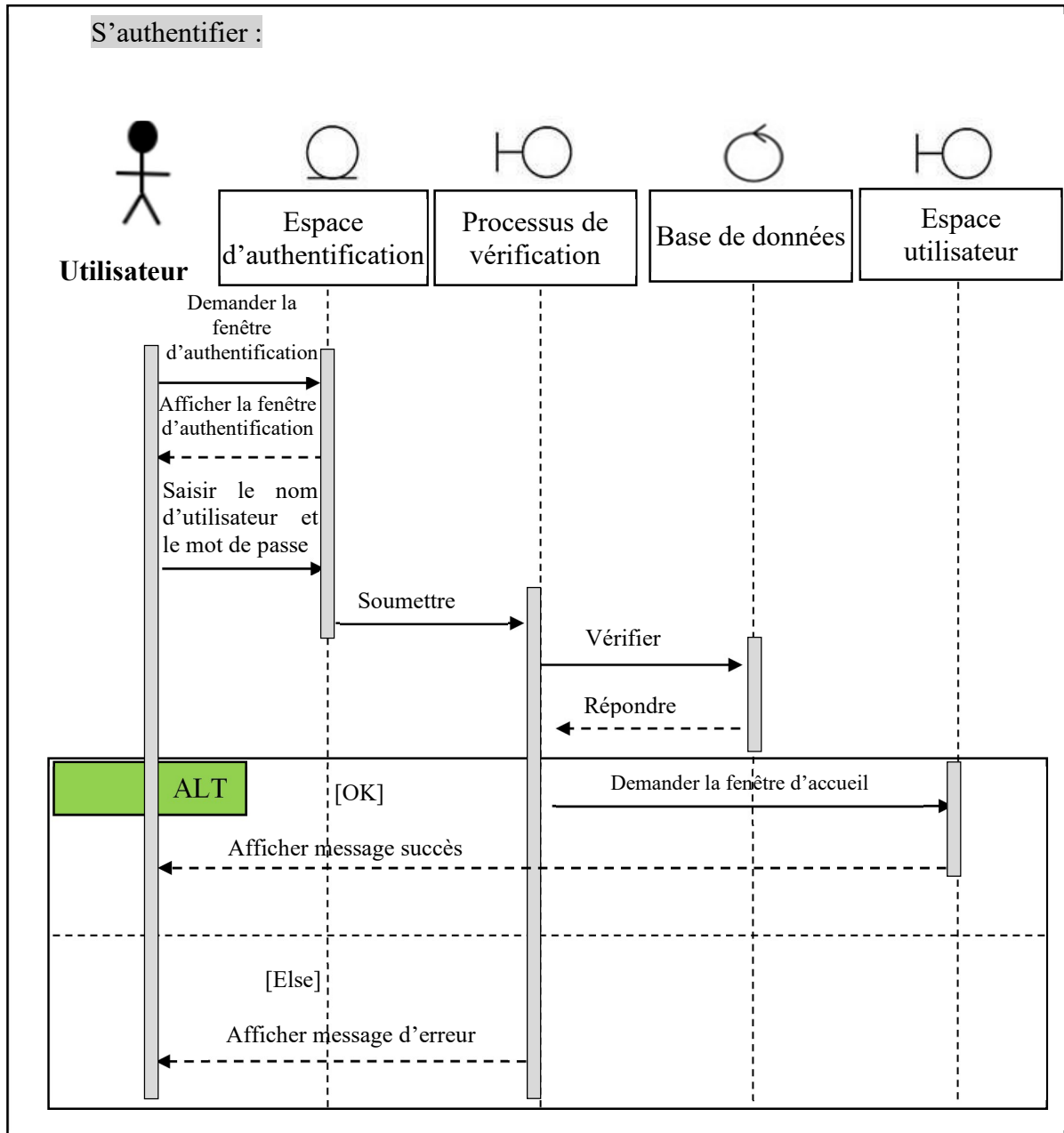
Les scénarios sont des instances des cas d'utilisation et sont traduits en diagrammes de séquences. Les classes d'analyses peuvent être divisé en trois catégories d'objet :

- **Les objets de type interface** : ils représentent l'interface entre l'acteur et le système présenter par l'icone 
- **Les objets de type entité** : ils représentent principalement les bases de données, présenter par l'icone 

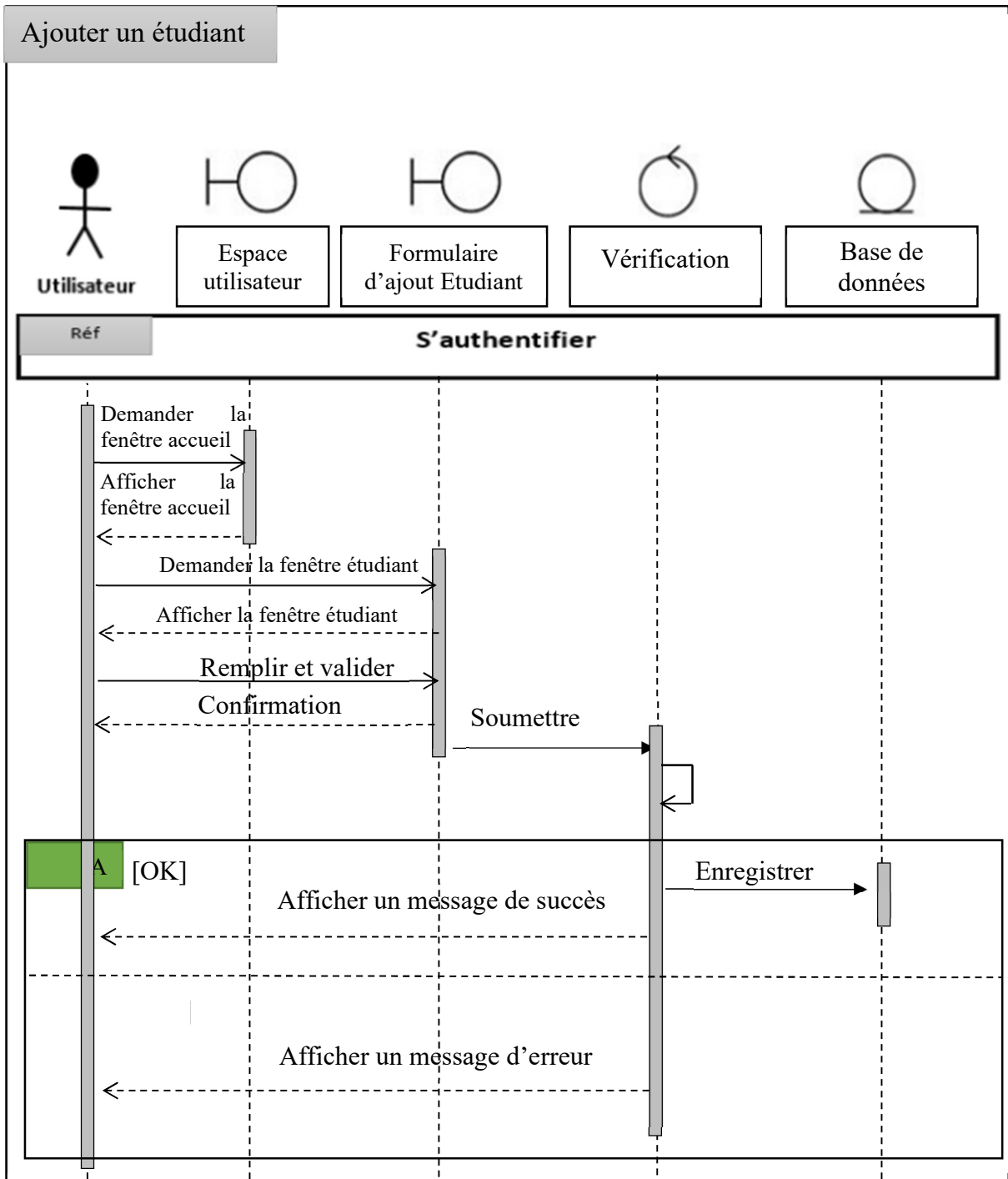
- **Les objets de contrôle** : ils représentent les processus du système, présenter par l'icone 

Vu le nombre élevé de cas d'utilisation recensés, et afin d'éviter qu'ils s'étalent sur plusieurs pages, nous allons étudier que quelques diagrammes :

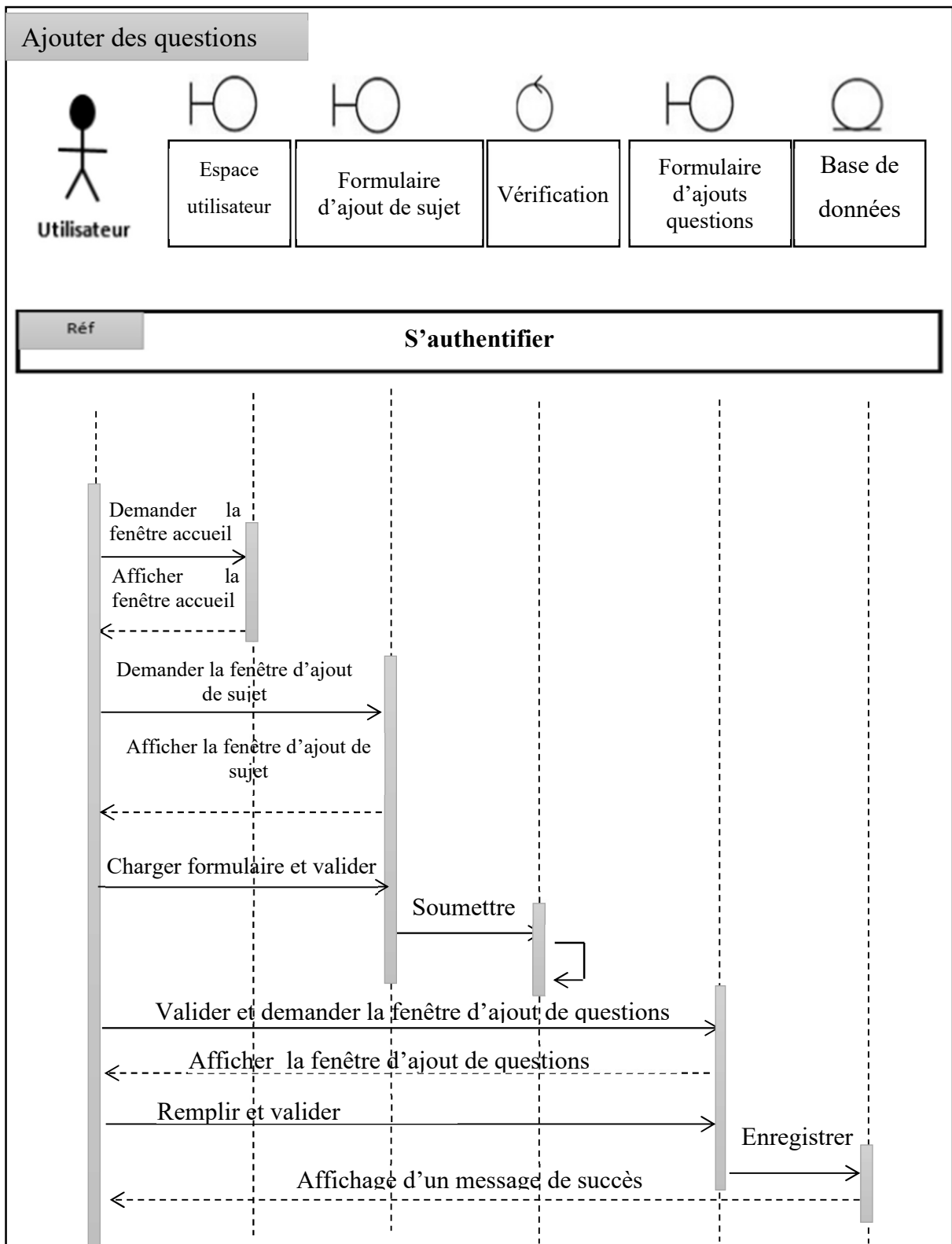
- Diagramme de séquence du cas " S'authentifier".
- Diagramme de séquence du cas "Ajouter un étudiant ".
- Diagramme de séquence du cas "Ajouter des questions".



**Figure 3.3:** Diagramme de séquence du cas " S'authentifier".



**Figure 3.4:** Diagramme de séquence du cas " Ajouter un étudiant ".



**Figure 3.5:** Diagramme de séquence du cas « Ajouter des questions ».

3.7.2 Diagramme de classe :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il montre la structure interne du système.

Il contient principalement des classes, ces dernières contiennent des attributs et des opérations, aussi il démontre les liens entre ces classes ainsi que leurs cardinalités.

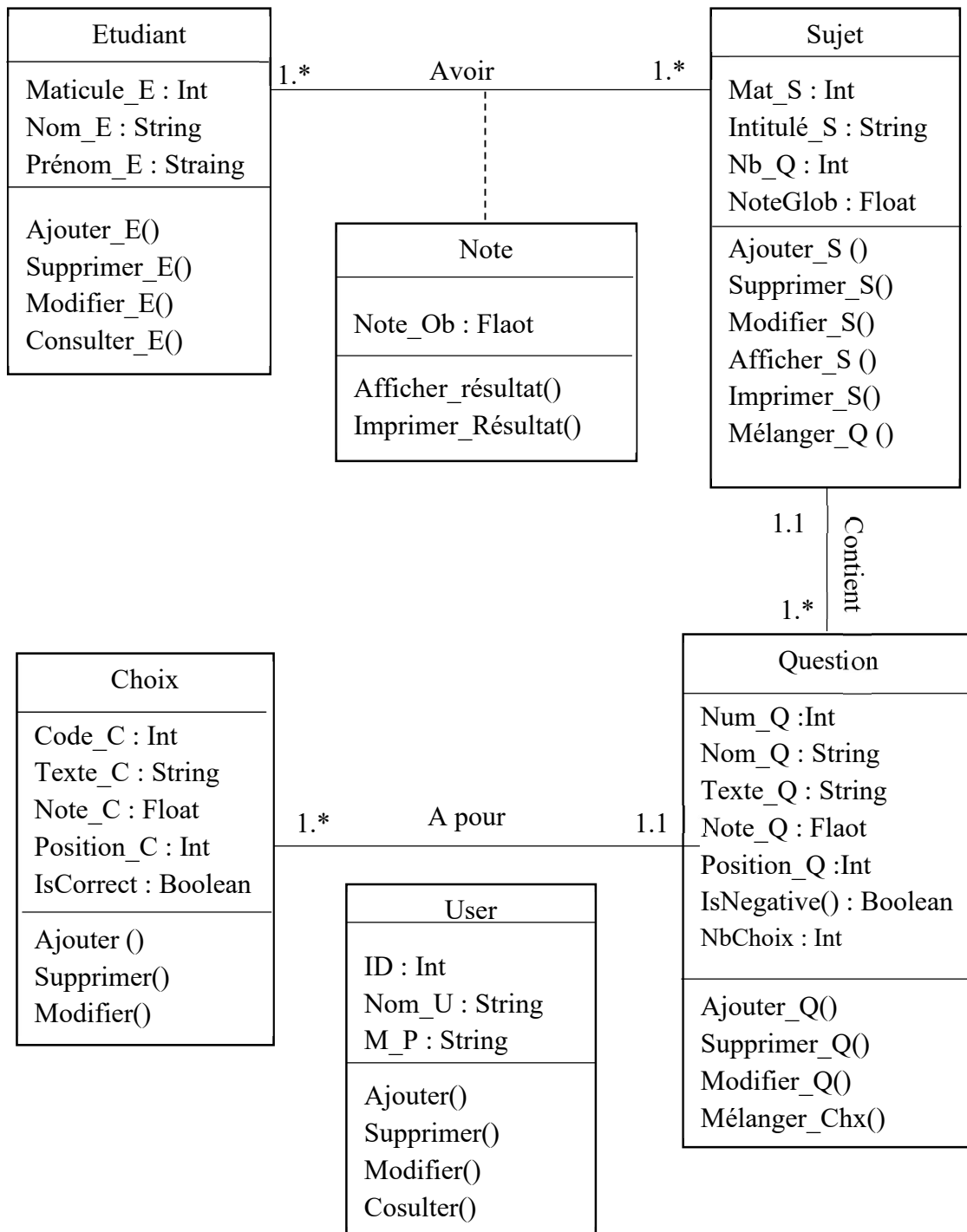


Figure 3-6 Diagramme de classe

### 3.7.3 Conception de la base de données

Une base de données est une collection de données cohérente et structurée.

Pour concevoir la base de données du système, nous avons commencé par recenser les différentes entités qui peuvent intervenir dans l'application. [10]

En se basant sur ces entités, et en respectant les différentes règles du modèle relationnel nous avons déduit les tables de la base de données.

### 3.7.4 Passage du diagramme de classe au modèle relationnel de données

- Chaque classe devient une table dont la clé primaire correspond généralement à l'identifiant de la classe.
- L'association de plusieurs à plusieurs devienne une table dont la clé primaire est composée des clés primaires des tables correspondant aux classes associées.
- L'association d'un à plusieurs (père-fils) provoque la migration de la clé primaire du père vers le fils et qui devient clé étrangère.
- Les classes spécifiques deviennent des tables dont la clé primaire est celle de la table correspondant à la classe générale.

### 3.7.5 Le modèle relationnel de données

- **Etudiant** (Matricule\_E, Nom\_E, Prenom\_E)
- **Sujet** (Mat\_S, Intitulé\_S, Nb\_Q, NoteGlob)
- **Question** (Num\_Q, Nom\_Q, Texte\_Q, Note\_Q, Position\_Q, IsNegative(), NbChoix, #Mat\_S)
- **Choix** (Code\_C, Texte\_C, Note\_C, Position\_C, IsCorrect, #Num\_Q)
- **Note** (Matricule\_E, Mat\_S)
- **User** (ID, Nom\_U, M\_P, Note\_Ob)

### **3.8 Conclusion**

Dans ce chapitre, nous nous sommes concentrés sur les aspects analytique et conceptuel de notre application ainsi que les Bases de données qui interagissent avec elle.

Nous avons défini les acteurs de notre application et les tâches qu'ils assurent, puis élaboré le diagramme de contexte, le diagramme de cas d'utilisation, les diagrammes de séquence et le diagramme de classes. Enfin, nous avons conçu le modèle relationnel, qui nous a permis la conception et création de la base de données.

Dans le chapitre suivant nous entamerons l'étape d'implémentation de notre application.

# CHAPITRE 4

## IMPLEMENTATION ET RESULTAT CHAPITRE 4

### IMPLEMENTATION ET RESULTAT

#### 4. IMPLEMENTATION ET RESULTAT

4.1 Introduction :

4.2 Matériel utilisé :

4.3 Environnement de développement :

4.3.1 Fonctionnalités :

4.4 Langage de programmation java :

4.5 Système de gestion de base de données : SQLite

4.6 Présentation des interfaces :

4.6.1 Interface 'authentification :

4.6.2 interfaces accueil

4.6.3 Fenêtre étudiant :

4.6.4 Fenêtre Sujet :

4.6.5 Fenêtre questions

4.6.6 Fenêtre camera

4.6.7 Fenêtre résultat :

4.7 Conclusion :

## 4. IMPLEMENTATION ET RESULTAT

### 4.1 Introduction :

Dans ce chapitre nous allons commencer tout d'abord par la description de notre environnement de travail, le langage ainsi que les outils utilisés pour l'implémentation. Par la suite nous allons présenter le fonctionnement général de notre application ainsi que ses différentes interfaces.

### 4.2 Matériel utilisé :

Durant toute la période de développement nous avons utilisé comme matériel :

- **Pour développement** : un laptop HP avec les caractéristiques suivante : (voir figure : 4.1)

- **Fabriquant** : HP
- **Modèle** : HP Pavilion - 15-bc314nf
- **RAM**: 12 GO
- **Processeur** : I5 7eme Gen
- **Disque dur** : 1 TO
- **Graphique** : NVIDIA® GeForce® GTX 950M (2 Go de mémoire GDDR5 dédiée)
- **Système d'exploitation** : Microsoft Windows 10 version 21H2 mise à jour novembre

2021



**Figure 4.1** : Laptop HP Pavilion

- **Pour les tests** : un smartphone avec les caractéristiques suivante : (voir figure 4.2)

- **Fabriquant** : Samsung
- **Modèle** : Galaxy S7 Edge

- **RAM**: 4 GO
- **Disque dur** : 32 GO
- **Système d'exploitation** : Android 8.0
- **Batterie**: 3500 mAH
- **Camera**: 12 m



**Figure 4.2:** smartphone Samsung Galaxy S7 Edge

- Android studio version: Bumblebee (2021.1.1) Stable
- Emulateur : S7 Edge
- Langage de programmation : Java

#### **4.3 Environnement de développement :**

Pour l'implémentation et le développement de l'application nous avons utilisé la version Bumblebee (2021.1.1) Stable d'Android Studio doté avec un émulateur S7

Android studio est un environnement de développement pour développer des applications mobiles Android, Edge, Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.

Avant Android Studio, de 2009 à 2014, Google propose comme environnement de développement officiel une distribution spécifique de l'environnement Eclipse, contenant notamment le SDK d'Android.[6]

Android Studio est annoncé le 15 mai 2013 lors du Google I/O et une version Early Access Preview est disponible le jour même.

Le 8 décembre 2014, Android Studio passe de version bêta à version stable 1.0. L'environnement devient alors conseillé par Google, et Eclipse est délaissé. (Voir figure 4.3)



**FIGURE 4.3:** image Android studio 4.3 (Canary 8)

### 4.3.1 Fonctionnalités :

Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur. [1].

Android studio été notre premier choix pour le développement de notre application à cause de ses fonctionnalités qui sont :

- Exécution instantanée de l'application. ...

- Éditeur de mise en page visuelle. ...
- Émulateur rapide. ...
- Éditeur de code de renseignement. ...
- Ajout d'une nouvelle activité en tant que modèle de code. ...
- Aide à la création d'une application pour tous les appareils. ...
- Aide à se connecter avec Firebase. ...
- Soutenez KOTLIN [3]

### 4.4 Langage de programmation java :

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation. [6]

### 4.5 Système de gestion de base de données : SQLite

**SQLite** est une bibliothèque portable, open source, écrite en langage C, qui embarque un moteur de base de données relationnelle accessible par le langage SQL et implémente en grande partie les propriétés ACID, à savoir atomicité, cohérence, isolation et durabilité.

Contrairement à la plupart des SGBD, **SQLite** ne repose pas sur une architecture traditionnelle client/serveur, mais lit et écrit directement depuis un fichier classique. En d'autres termes, **SQLite** intègre une base de données SQL complète (tables, index, déclarations et données) dans un fichier indépendant de la plateforme.

Les fichiers de bases de données SQLite sont multi-plateformes, et peuvent être copiés d'un système 32 bits à un système 64 bits, ou d'une architecture big-endian à une architecture little-endian. Par conséquent, **SQLite** se présente comme un choix raisonné dans la création de fichiers d'application.

Enfin, **SQLite** s'impose comme une bibliothèque compacte et très légère, des caractéristiques idéales qui lui permettent de parfaitement fonctionner sur des systèmes embarqués comme les smartphones ou les lecteurs MP3. [5]



**Figure 4.4** : logo de SQLite

#### 4.6 Présentation des interfaces :

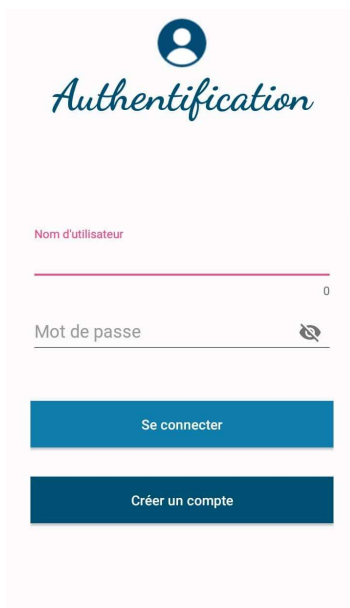
L'application prend en charge les fonctionnalités : gestion des étudiants, et comme fonctionnalités principales : la création et la correction automatique des QCM.

Nous allons présenter dans ce qui suit les principales interfaces illustrant le fonctionnement de l'application :

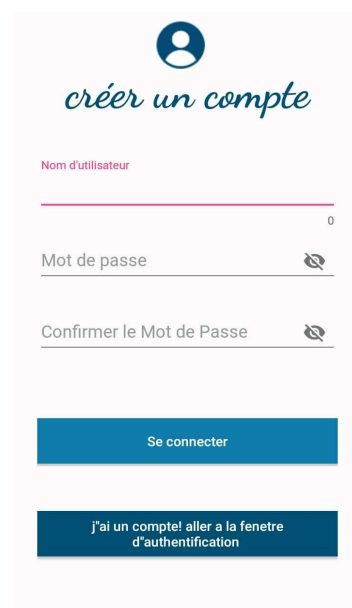
- Interface d'authentification
- Interface accueil (espace utilisateur)
- Interface étudiants
- Interface Sujet
- Interface d'ajout des questions
- Caméra

##### 4.6.1 Interface d'authentification :

Au démarrage de l'application, l'interface d'authentification s'affiche. L'utilisateur doit saisir son nom d'utilisateur et son mot de passe pour s'authentifier puis accède à son espace : (voir figure 4.5), de même il peut créer un nouveau compte (voir figure 4.6)



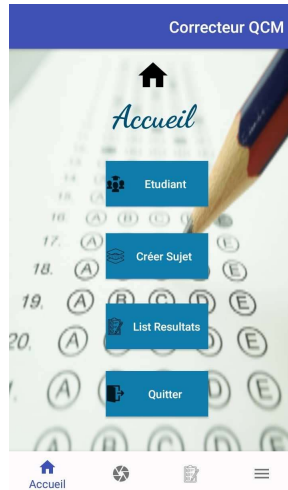
**Figure 4.5** : Fenêtre « Authentification »



**Figure 4.6** : Fenêtre « créer un compte »

4.6.2 Interfaces accueil

Après avoir authentifié, l'utilisateur accède à son espace du travail (accueil), à partir duquel il peut accéder aux différentes fonctionnalités en cliquant sur le lien correspondant (voir Figure 4.7), d'où il peut ajouter des étudiants ou créer des sujets, et de consulter les résultats obtenus.



**Figure 4.7** : Fenêtre « accueil »

4.6.3 Fenêtre étudiant :

Depuis la fenêtre accueil l'utilisateur peut accéder à la **fenêtre étudiant** (voir Figure 4.8), en cliquant sur le bouton **Créer Etudiant**, d'où il peut :

- Ajouter un étudiant (voir Figure 4.9)
- Voir la liste des étudiants (voir Figure 4.10)
- Quitter la fenêtre



**Figure 4.8** : fenêtre étudiant



**Figure 4.9** : fenêtre ajouter étudiant



**Figure 4.10** : fenêtre liste des étudiant

### 4.6.4 Fenêtre Sujet :

Depuis la fenêtre accueil l'utilisateur peut accéder à la **fenêtre Sujet** (voir Figure 4.11), en cliquant sur le bouton Créer **sujet**, d'où il peut :

- Ajouter un nouveau sujet
- Modifier un sujet existant
- Supprimer un sujet existant
- Afficher les questions d'un sujet existant.

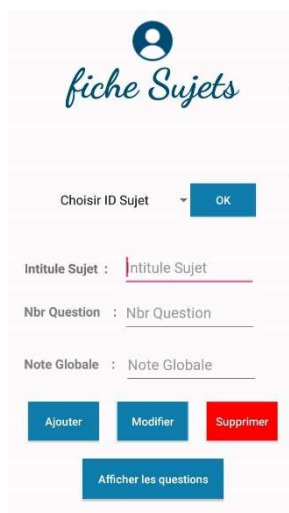


Figure : 4.11 : Fenêtre sujet

En cliquant sur le bouton ajouter, une fenêtre apparaitre et nous permettre d'ajouter un nouveau sujet et nous faire appel à saisir les informations tel que son intitulé, le nombre total de questions qui contient ainsi la note globale ( voir Figure 4.12)

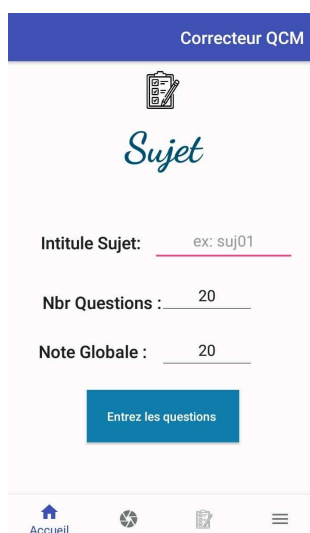


Figure 4.12 : Fenêtre d'ajout d'un sujet

A partir de cette fenêtre l'utilisateur doit créer les questions que doit contenir le sujet.

## Chapitre 4 : implémentation et réalisation

### 4.6.5 Fenêtre questions :

Là où l'utilisateur est appelé d'ajouter les questions au sujet (voir figure 4.13) :



Figure 4.13 : Fenêtre questions

A partir de cette fenêtre l'utilisateur peut :

- Ajouter une question jusqu'à où il atteint le nombre de questions déclaré à la fiche sujet.
- Mélanger l'ordre de questions.
- Enregistrer et imprimer le sujet sous format PDF.
- Comme il doit choisir la bonne réponse en cliquant sur le bouton retour.

### 4.6.6 Fenêtre camera :

L'utilisateur faire appel à la caméra depuis la fenêtre sujet (voir figure 4.12), en procédant au scanne de la copie d'examen (voir figure 4.13), le résultat sera obtenu et enregistré immédiatement et instantanément.

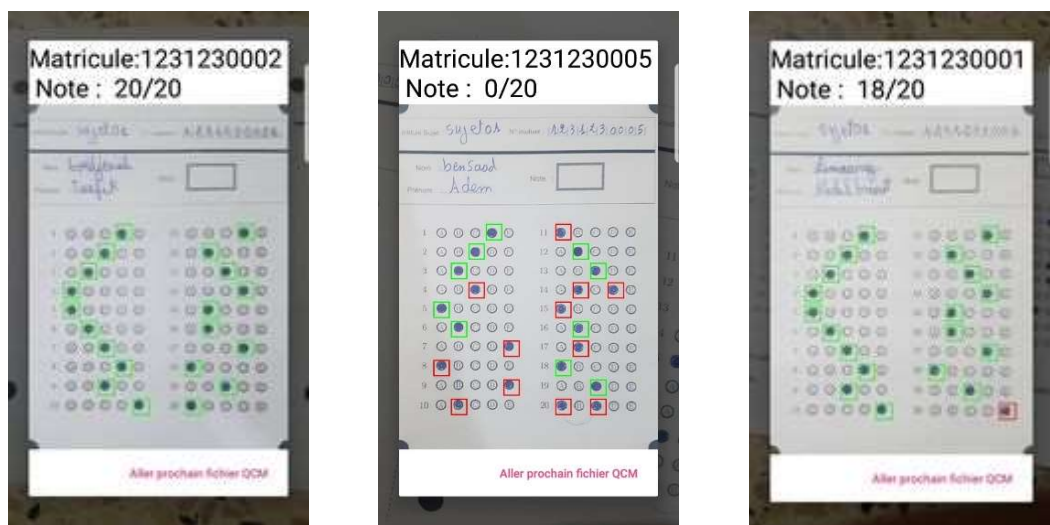


Figure 4.14 : fenêtres résultat obtenue lors du scan d'une copie

## Chapitre 4 : implémentation et réalisation

### 4.6.7 Fenêtre résultat :

Depuis la fenêtre d'accueil l'utilisateur peut consulter les résultats obtenus par les étudiants (voir figure 4.15) :



Figure 4.15 : fenêtre résultats

A partir de cette dernière, l'utilisateur peut consulter les résultats obtenus en cliquant sur le bouton « **voir liste** ».

Comme il peut imprimer la liste de résultat obtenue en cliquant sur le bouton « **Imprimer** » (Voir Figure 4.16)



matricule	Nom	Prenom	Note
1231230001	Lemaareg	abdel basset	18
1231230002	BOUDJENA H	toufik	20
1231230003	BECHERE	masoud	15
1231230004	BENNAAMA	IMAN	10
1231230005	BENSAAD	ADEM	0
1231230006	CHETRAOUI	AGMED	10
1231230007	HADDAD	MINOU	12

Figure 4.16 : liste de résultat

## Chapitre 4 : implémentation et réalisation

Aussi il peut consulter la note de chaque étudiant individuellement en boucle cliquant sur la ligne de chaque étudiant (voir figure 4.17)



Etudiant

Matricule : 1231230003

Nom : BECHERE

Prenom : masoud

Enregistrer Supprimer

Figure 4.17 : résultat d'un étudiant

### **4.7 Conclusion :**

Dans Ce dernier chapitre ou nous l'avons consacré à la présentation de l'étape de réalisation de notre application.

Le passage de l'algorithme à l'application est la phase la plus importante, d'après plusieurs tests nous avons réussi a trouvé une méthode de correction automatique de copies d'examen au format QCM ont utilisant un smartphone, Les résultats présentés montrent bien la fiabilité de notre application.

Nous avons débuté ce chapitre par la présentation des outils de développement qui nous ont permis la réalisation de notre travail à savoir l'environnement de développement et les langages de programmation.

Puis, nous sommes passés à la présentation de notre application en décrivant ses fonctionnalités et présentant plusieurs interfaces.

# Conclusion générale

Au cours de notre travail, nous avons réalisé une application dédiée à la correction automatique de copies d'examens au format QCM, en utilisant un Smartphone comme un système d'acquisition.

En premier lieu, nous avons présenté quelque notion de bases au traitement d'image.

Ensuite, nous avons introduit les différentes méthodes utilisant pour la correction automatique de copies d'examen au format QCM.

Puis nous avons entamé la phase d'analyse et de conception en utilisant la UML comme un langage de modélisation, où on a présenté le diagramme de séquence et le diagramme de classe en terminant par le modèle logique de donnée qui a représenté une introduction à la dernière phase.

Et enfin nous avons décrit l'espace de travail de notre application, ainsi que les différents résultats obtenus.

A travers ce travail, nous avons pu acquérir beaucoup de connaissances dans le domaine du traitement d'image. Premièrement, nous avons utilisé les outils appris au cours de notre formation, et enrichi nos connaissances dans le domaine du traitement d'image. On a aussi appris à utiliser le langage de programmation java et l'environnement de développement Android studio.

Nous proposons comme continuité à ce travail l'intégration a des systèmes en ligne comme MOODLE, l'importation et l'exportation des listes et des notes de et vers d'autres formats comme Excel ...

# BIBLIOGRAPHIE

- [1] AICHE W, AMROUCHE ABE, correction automatique de copies d'examen au format QCM, mémoire pour l'obtention d'un diplôme de master, Université Saad DAHLEB de BLIDA, 2015/2016.
- [2] AMC : <https://www.auto-multiple-choice.net> visité le 28/05/2022
- [3] BENYOUCEF L, BOUAMARA N, conception et réalisation d'une application mobile multiplateforme pour la gestion des ventes, Mémoire pour l'obtention de master, Université de Mouloud MAAMERIE de Tizi Ouzou, 2019/2020
- [4] BOUKHLOUF D Enseignante d'informatique Université Mohammed KHIDER BISKRA, Support de cours, Généralité sur le traitement d'image, Université de BISKRA
- [5] **Edu1d.ac-toulouse** : <https://edu1d.ac-toulouse.fr/> visité le 15/02/2022
- [6] **Wikipédia** : [www.Wikipédia.com](http://www.Wikipédia.com) visité le 21/05/2022
- [7] **Logiciels** : [www.logiciels.com](http://www.logiciels.com) visite le 21/05/2022
- [8] **01net** : [www.01net.com](http://www.01net.com) visité le 21/05/2022
- [9] **Espe** : <http://espe-rtd-reflexpro.u-ga.fr>: visité le 28/05/2022
- [10] **Archives ouverts** : <https://tel.archives-ouvertes.fr> visite le : 13/05/2022
- [11] **Techno** : <https://www.techno-science.net> visité le : 12/02/2022
- [12] **Etudier** : Le monde des images - 1773 Mots | Etudier visité le 12/02/2022

## Résumé

**Résumé :** Ce projet consiste à préparer un outil d'assistance pour les enseignants qui permet la préparation, la création et la correction automatique de sujet aux formats QCM (Question à Choix Multiples) via téléphone portable.

Le traitement est appliqué à des images en niveaux de gris codé sous 8 bits et au format A4.

Le système d'acquisition utilisé est un smartphone doté d'un système d'exploitation Android. Le projet est réalisé en utilisant le langage de programmation java, et Android studio comme IDE.

**Mots clés :** QCM, Correction Automatique, Traitement d'Image, Application Mobile.

---

**ملخص:** هذا المشروع عبارة عن اعداد وسيلة مساعدة للأساتذة، تسمح بإعداد، إنشاء مواضيع الامتحانات على شكل QCM (أسئلة ذات خيارات متعددة)، وكذا التصحيح الآلي لإجابات الطلبة عبر المسح الضوئي بواسطة الهاتف المحمول.

تم معالجة الصور بالترج الرمادي مشفرة بتنسيق 8 بت وعلى أوراق من مقاس A4. يتم استعمال هاتف ذكي مزود بنظام تشغيل Android للقيام بعملية النسخ (التصوير)، تم استخدام لغة البرمجة JAVA، وأندرويد ستوديو كبيئة تطوير متكاملة.

**الكلمات المفتاحية:** التصحيح الآلي للامتحانات. التصحيح الآلي، معالجة الصور، تطبيقات الهاتف المحمول.

---

**Abstract:** This project consists of preparing a support tool for teachers that allows the preparation, creation and automatic correction of subjects in MCQ (Multiple Choice Question) formats, via mobile phone.

The process is applied to grayscale images encoded in 8-bit and A4 format.

The acquisition system used is a smartphone with an Android operating system. The project is carried out using the java programming language, and Android studio as IDE.

**Keywords:** MCQ, Automatic Correction, Image Processing, Mobile Application.