Numériser le patrimoine

Mathilde Schwoerer

1 Océrisation avec Pytesseract

1.1 Préparation de la session de travail

On importe les boîtes à outils requises pour ce type de travaux. Nous utiliserons Pytesseract et Tesseract, les outils d'océrisation de Google pour nous entraîner à coder en Python.

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from PIL import Image
5 import pytesseract
```

On nécessite CV2 (Open-CV) pour manipuler les images et réaliser le preprocessing. Numpy permet de gérer les listes de nombres, Matplotlib.pyplot les images. PIL (Pillow) sert à afficher les images et à les convertir si besoin.

On indique également le chemin d'accès vers les documents à traiter. Il s'agit de l'édition du *Carmen de martyrio Maccabaerum* dans l'édition de Peiper [1]. On fournit à la machine des images au format .jpeg, qui présente comme caractéristique utile dans le cadre d'une océrisation d'être composée d'une matrice de pixels et donc de ne pas se prêter au zoom. On se souvient que chaque pixel de couleur est un triplé de nombres renseignés selon le rouge, le vert et le bleu. Chaque nombre se définit entre 0 et 255. Cette combinaison définit une unique couleur. Nous verrons ensuite, à l'aide de techniques de binarisation fondées sur un seuil à définir, comment faire passer l'image en gris, c'est-à-dire une matrice dans laquelle chaque pixel contient un unique nombre entre 0 et 255.

Les pages à traiter se trouvent dans un dossier intitulé /Photos/.

1.2 Définition des fonctions Python

Pour nous aider à préparer les images, nous avons implémenté deux fonctions Python qui réalisent ces tâches. La première prend une image en entrée pour la renvoyer en teintes de gris. Quant à la seconde, elle renvoie l'image binarisée, ici selon le seuil d'OTSU.

```
1 #Mettre l'image en gris :
2 def get_grayscale(image):
3     return cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
4
5
6 def thresholding(image):
7 retenues du pixel.
8     (T, threshInv) = cv2.threshold(image, 0, 255,cv2.
          THRESH_BINARY_INV | cv2.THRESH_OTSU)
```

1. R. Peiper, "Carmen de martyrio Maccabaeorum", CSEL, 23, 1854, p. 240-254.

```
9     return threshInv
```

1.3 Prétraitement d'une page témoin

Avant de traiter l'ensemble du corpus, on se livre d'abord à quelques prétraitements (ou *preprocessing*) pour voir quel est l'impact de chacun sur le résultat de l'océrisation sur la première page du poème qui nous sert de témoin. Ainsi, on océrise successivement l'image :

- brute ;
- en teinte de gris ;
- avec seuil. Ce principe repose sur la binarisation de l'image. Au-delà d'un certain seuil à déterminer, les pixels sont considérés comme noirs, tandis que ceux en-deçà dudit seuil apparaissent comme blancs ;
- avec le traitement de dilatation appliqué à l'image seuillée ;
- avec un traitement d'érosion appliqué à l'image seuillée.

Voici le code utilisé :

```
1 img = cv2.imread('./CdMM/Photos/1.jpg')
2 gray = get_grayscale(img)
3 thresh = thresholding(gray, False, True)
4 opening = opening(thresh)
5 canny = canny(thresh)
```

La dernière méthode utilisée, l'érosion sur l'image binarisée, produit une image de très mauvaise qualité. Cette piètre résolution ruisselle ensuite sur l'océrisation.

Extrait d'océrisation avec cette méthode :

> DER DU DDR DL CRGA ul Miaeeeb, Xr lex ftit Antiochus Syme Gssunws olm. 667 m euis regmo meter megwe ferunt septem, ut fauna refero, de semen gemte crest. quos cura adsidwe gemeris mxneunimigse iubes et legi seruire deae : exe meunque wolemtes le populo meliore dex sue nura temelbanmt. Fex adus mutare foem≫, mouere tuooneu iwstorem wolui. populo ut migceret inquo. pma prom moth, prout e su, 10 et quadquid mosset toto comquimerms megmo, Si modo mautmret mores semsusque jrores.

Les autres méthodes offrent des résultats satisfaisants et peu différents entre eux. Voici par exemple l'image traitée en teintes de gris et le texte converti en chaîne de caractères :

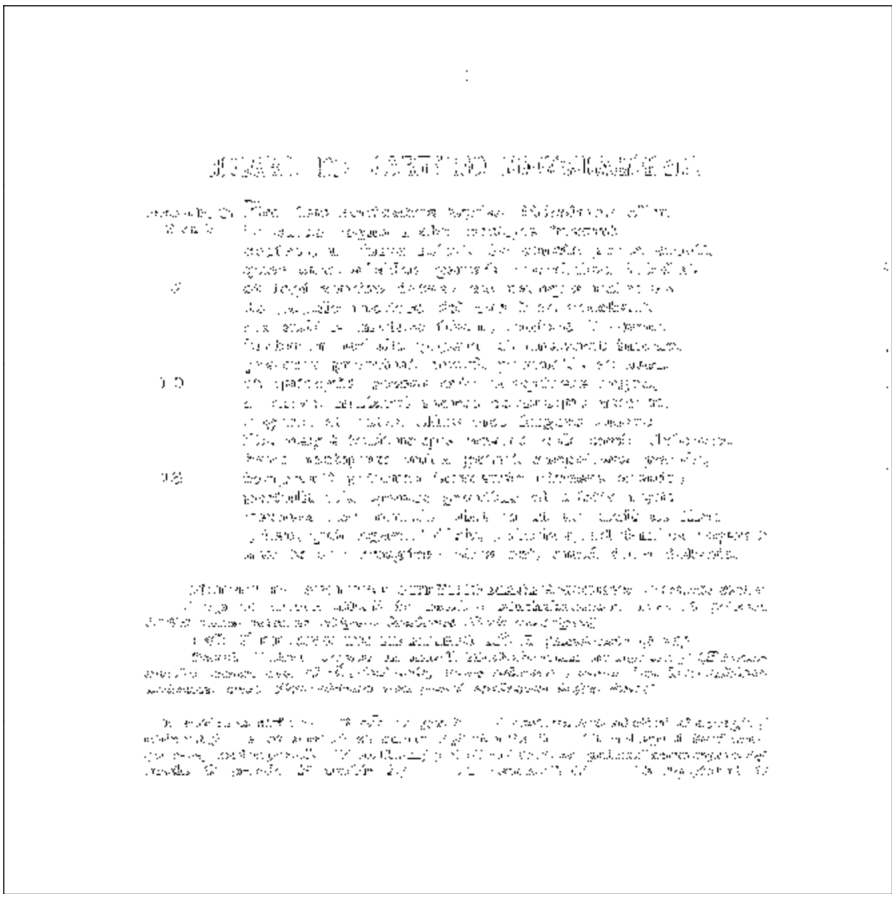
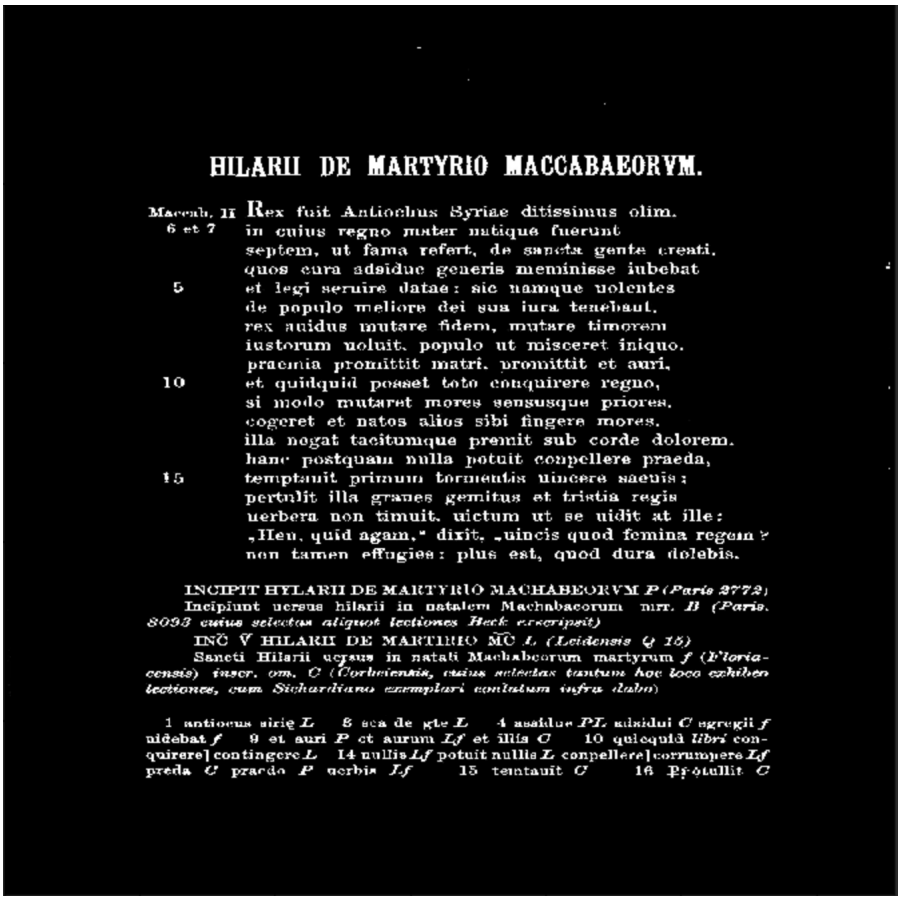


FIGURE 1 – Page témoin après binarisation et érosion

HILARII DE MARTYRIO MACCABAEORYM,
Maccab. II lex. fuit. Antiochus Syriae ditissimus olim.
6 et 7 jn cuius regno mater natique fuerunt septem, ut fama refert,
de sancta gente creati. quos cura adsidue generis meminisse iubebat
9 et legi seruire datae : sic namque uolentes de populo meliore dei sua
iura tenebant. rex auidus mutare fidem, mutare timorem iustorum
uoluit. populo ut misceret iniquo. praemia promittit matri, promittit
et auri,
10 et quidquid posset toto conquirere regno, si modo mutaret mores
sensusque priores, eogeret et natos alios sibi fingere mores.

Cette océrisation ne compte que trois erreurs. La première concerne le "V", le -u majuscule de "Maccabaeorum" dans le titre, interprété comme un Y. Ensuite, le R majuscule de *Rex* (l. 2) est lu comme un -l. La dernière transforme le -i de *in* en -j (l. 3). Sur cet échantillon, l'océrisation du texte brut, de la page en teinte de gris et de l'image binarisée donnent le même résultat, avec les mêmes trois erreurs. En revanche, si pytesseract interprète correctement *in* sur l'image dilatée, il commet de nombreuses erreurs ailleurs.



HILARII DE MARTYRIO MACCABAEORVM.

Macab. II 6 et 7 Rex fuit Antiochus Syriae ditissimus olim.
in cuius regno mater utique fuerunt
septem, ut fama refert, de sancta gente creati,
quos cura adsidue generis meminisse iubebat
et legi seruire datae: sic namque uolentes
de populo meliore dei sua iura tenebant.
rex auidus mutare fidem, mutare timorem
iustorum uoluit. populo ut misceret iniquo.
praemia promittit matri. promittit et auri,
et quidquid posset toto conquirere regno,
si modo mutaret mores sensusque priores.
cogeret et natos alios sibi fingere mores.
illa negat tacitumque premit sub corde dolorem.
hanc postquam nulla potuit compellere praeda,
temptauit primum tormentis uincere saeuis:
pertulit illa graues gemitus et tristia regis
uerbera non timuit. uictum ut se uidit at ille:
„Heu, quid agam," dixit, „uincis quod femina regem?
non tamen effugies: plus est, quod dura dolebis.

INCIPIT HYLARII DE MARTYRIO MACHABEORVM *P (Paris 2772)*
Incipiunt uersus hilarii in natalem Machabaeorum mrr. *B (Paris. 8093 cuius selectas aliquot lectiones Heck exscripsit)*
INC̄ V̄ HILARII DE MARTIRIO M̄C̄ *L (Leidensis Q 15)*
Sancti Hilarii uersus in natali Machabeorum martyrum *f (Floriacensis) inscr. om. C (Corbeiensis, cuius selectas tantum hoc loco exhibemus lectiones, cum Sicharidiano exemplari conlatum infra dabo)*

1 antiocus sirię *L* 3 sca de gte *L* 4 assidue *PL* adsidui *C* egregii *f* 5 iidebat *f* 9 et auri *P* et aurum *Lf* et illis *C* 10 quicquid *libri* conquirere] contingere *L* 14 nullis *Lf* potuit nullis *L* compellere] corrumpere *Lf* preda *C* pręda *P* uerbis *Lf* 15 temtauit *C* 16 ꝑtulit *C*

FIGURE 2 – Page témoin en teintes de gris

Ainsi, quelque soit la méthode, entre conservation du texte brut, teintes de gris ou binarisation, l'océrisation semble probante.

1.4 Sectionnement de la page

On a voulu voir s'il était possible de localiser les zones de textes sur les images. Si on connaît leur position sur la page, on enrichit notre connaissance du document : dans le cas d'un manuscrit avec des notes marginales, on sait ainsi qu'il s'agit probablement d'un commentaire ou d'une correction. Pour cela, il existe une commande dans Pytesseract pour extraire des rectangles et leurs coordonnées autour du texte :

```
1 d = pytesseract.image_to_data(img, output_type=Output.DICT)
```

Bien que tous les mots n'aient pas été localisés et que le titre n'ait pas été repéré, l'ensemble est plutôt probant :

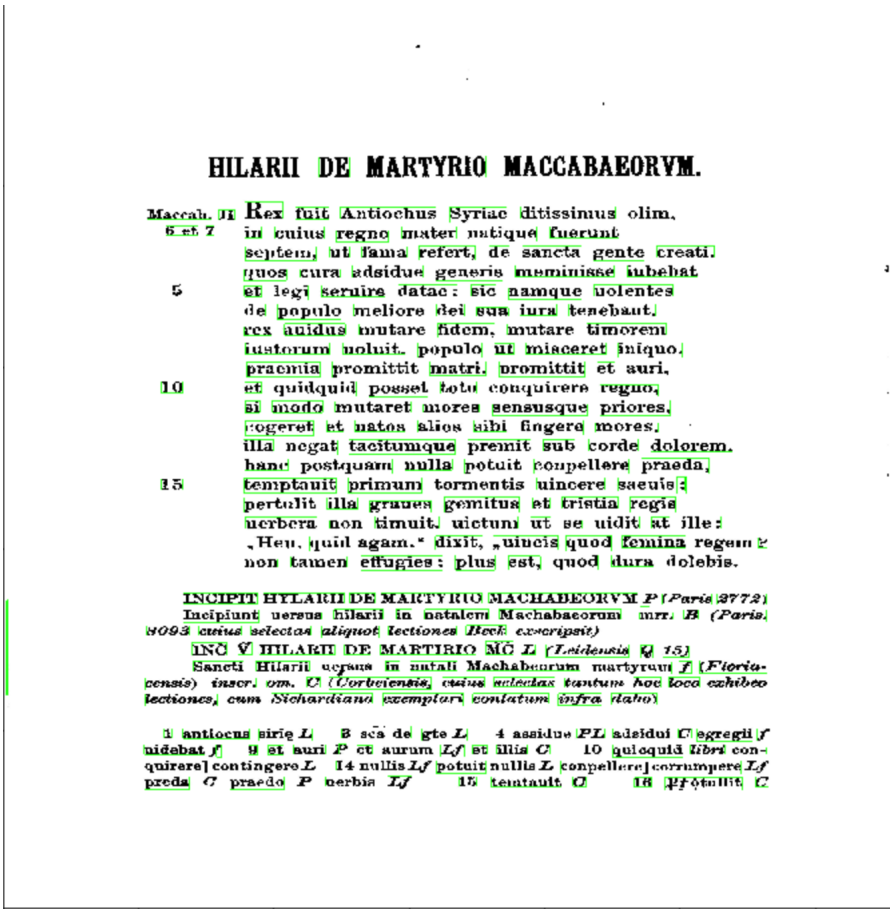


FIGURE 3 – Boîtes de texte

1.5 Création d'une boucle

La méthode précédente fonctionne manuellement page par page, néanmoins elle ne se prêterait pas à la numérisation d'un ouvrage dans son ensemble, car elle serait trop laborieuse. C'est pourquoi nous mettons en place une boucle qui traite les images une par une automatiquement. Nous enregistrons au fur et à mesure le texte, copié dans un fichier .txt qui contiendra l'ensemble du texte océrisé. À l'avenir, ce pourra être une bonne matière première pour un fichier .xml.

Une chaîne de caractères nommée "texte réuni" accueillera le résultat de l'océrisation de chaque page. On l'initialise vide. Pour chaque image allant de la page 1 à la page 14, on cherche le chemin qui pointe vers le fichier .jpeg. On l'ouvre ensuite à l'aide d'Open-CV. On en extrait le texte avec Pytesseract. On l'affiche et on le met bout à bout avec la variable "texte réuni". À l'issue de cette boucle, on enregistre dans un fichier .txt la variable "texte réuni" obtenue.

```
1 Texte_reuni=" "
2 for i in range(1,14):
```

```
3       path="./CdMM/Photos/" + str(i) + ".jpg"
4       img = cv2.imread(path)
5       Texte_etape_i=pytesseract.image_to_string(img, lang='lat')
6       print(Texte_etape_i)
7       Texte_reuni= Texte_reuni + " " + Texte_etape_i
8 with open('Texte_ocerise_carmen.txt', 'w') as file:
9       file.write(Texte_reuni)
```

En conclusion, nous avons établi une procédure automatisée qui transforme un fichier .jpeg en une chaîne de caractères contenue dans un fichier .txt grâce à la reconnaissance optique des lettres. Nous avons donc répondu à notre problématique. Cependant, nous avons pris au départ un texte imprimé, facile à océriser. Or si l'on s'attaque à une page de manuscrit richement ornée [2], de nombreux problèmes apparaissent, comme l'illustrent, en guise d'ouverture, l'image suivante binarisée et un extrait du texte qui en découle :

FIGURE 4 – Boîtes de texte

2. Lien vers l'image.

Quare. eulliy Ciecrome phibppienrii mods yos A. m hibes pin ? fear mapit. er] log rep. patree, Conci ea que q : wert boc tempeve azbitiez. erpená tebis bzewitez. confélium et poofveno. — monec, Cgo aim fperaiem aliquábe tm Confilu aucrorintéq : rem p. efie reno curam anandoum nidy fratiebam quati misiles quam confalavi-ac fénateua .Ylec weto ug ? um cembam. aut anv.p. "Oriaelson. ocdlog. exco ote qe men telluris cciecur (dms. faquo temple ? tum. m me fiit ta. fiirarméta pasas Atbemenfiag renesaut uetue exempli. gvent ettam bnm ufi aus, quot fum mféimpiois Dfcoiss. ntüspauevat- ruens ila.

2 Galerie des auteurs latins

Ce projet part du constat que les auteurs latins de l'Antiquité tardive restent méconnus à nos étudiants jusqu'à leur troisième année, où ils les découvrent tous en même temps, ce qui a tendance à les décourager. Bien moins austère qu'elle ne peut le paraître, l'Antiquité tardive regorge de perles littéraires et de personnalités singulières. Esquisser une galerie de portraits ludiques avec l'espoir d'amener nos étudiants à la lecture des textes, voilà l'idée fondamentale derrière la réalisation de ces quelques pages HTML [3]. Créature friande de détails croustillants, l'humain, comme l'avaient bien compris les grands maîtres de la rhétorique antique, s'intéresse davantage aux discours agréables, instructifs et touchants [4].

2.1 Processus

2.1.1 Création du CSS et d'une page HTML de référence

La première étape a consisté à créer une page HTML pour tester le code et contrôler son affichage. C'est le fichier tertullien.html. Après avoir réfléchi aux catégories, on ébauche la page dans les grandes lignes, en plaçant des titres et des paragraphes, par exemple. Ensuite, l'élaboration d'une feuille de style CSS rattachée au fichier HTML permet de paramétrer le dessin de la page plus finement. Voici le code employé pour relier la feuille CSS au document HTML :

```
1      <!DOCTYPE html>
2 <html>
3      <head>
4          <link rel="stylesheet" href="Theme_Prosopo1.css">
5          ...
6      </head>
```

Dès lors, on fait des allers-retours incessants entre le fichier CSS et la page HTML. On décide par exemple de l'apparence d'une barre de menu dans le CSS :

```
1 #Menu {
2    list-style-type: none;
3    margin: 0;
4    padding: 0;
5    overflow: hidden;
6    background-color: rgba(185, 70, 37, .8);
```

3. L'idée a éclos au cours d'une discussion avec Alice Leflaëc.
4. Les fameux *delectare, docere, mouere.*

```
7   box-shadow: 0 0 15px 0 rgba(0,0,0,.10);
8   text-align:center;
9 }
```

Le symbole "#" devant "Menu" est le sélecteur id. Il permet de donner une identité unique à l'objet qu'il caractérise. Le sélecteur accompagné du nom de l'objet doit se retrouver en attribut de l'argument "id =", ainsi :

```
1   <ul id="Menu">
2           <li class="dropdown">
3               <a href="javascript:void(0)" class="dropbtn">
                    Po tes</a>
4               <div class="dropdown-content">
5                   <a href="#">Ausone</a>
6                   <a href="#">Paulin de Nole</a>
7                   <a href="#">Prudence</a>
8                   <a href="#">Pseudo Cyprianus Gallus</a>
9               </div>
10          </li>
11          ...
12    </ul>
```

On peut complexifier le code pour obtenir des effets "interactifs" (angl. "responsive"), comme la légère coloration de l'arrière-plan lorsque l'on passe la souris pour sélectionner une page dans le menu déroulant :

```
1 .dropdown-content a:hover{
2 background-color: rgba(185, 70, 37, 0.3);
3 }
```

L'expression ".dropdown-content" correspond au menu déroulant contenu dans la barre de menu. Le point est un sélecteur de classe. Le sélecteur " :hover" spécifie dans quelle circonstance le style s'applique : quand on passe la souris sur les liens présents dans le menu déroulant. L'effet concerne un changement de couleur de l'arrière-plan du lien survolé.

Quand on a fini d'élaborer la maquette, on crée les autres pages HTML de la galerie, avant d'ajouter les liens correspondant à chacune dans la barre de menu :

```
1  <ul id="Menu">
2          <li class="dropdown">
3              <a href="javascript:void(0)" class="dropbtn">Po tes</a
                   >
4              <div class="dropdown-content">
5                  <a href="/Users/mathildeschwoerer/Documents/Humanit
                       %C3%A9s%20Num%C3%A9riques/Galerie%20des%20
                       auteurs%20latins/Ausone.html">Ausone</a>
6                  <a href="/Users/mathildeschwoerer/Documents/Humanit
                       %C3%A9s%20Num%C3%A9riques/Galerie%20des%20
                       auteurs%20latins/Paulin.html">Paulin de Nole</a
                       >
7                  ...
8              </div>
9          </li>
10     ...
11 </ul>
```

2.2 Pistes d'amélioration

Ce projet demeure encore au stade "beta", il y a encore du travail à fournir :
— remplir les champs de chaque page. La rédaction s'effectuera avec l'aide de collègues intéressés par cette galerie d'auteurs, toutefois l'architecture du site est complète.
— proposer des extraits de chaque auteur en .xml pour éventuellement créer un mini-jeu de recherche dans le texte pour les étudiants en rajoutant de l'interaction-utilisateur sur le site ;
— enfin, héberger la galerie sur un Git Hub pages dédié une fois qu'elle sera complète.

3 Sources

3.1 Bibliographie : ressources explicatives

— Tutoriaux HTML utilisés pour concevoir mon travail.
— Documentation HTML.
— Tutoriaux CSS.
— Documentation CSS.

3.2 Ressources des outils

— Les images libres de droit proviennent du site : `https://www.gettyimages.fr`
— Un outil de *web design* pour le choix des couleurs : `http://coolors.co.`