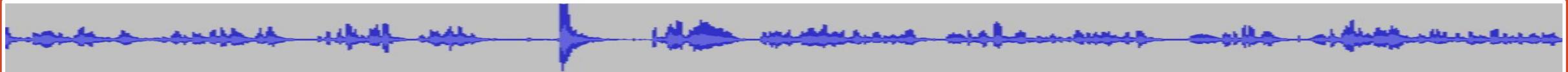


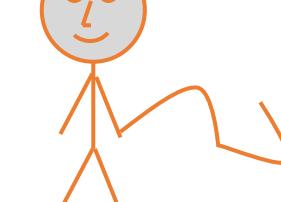
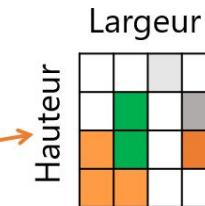
Concepts fondamentaux du Multimédia



JPEG, GIF,
PNG, WMF,
AI, S

**SOUND
FORGE**

Adobe
Photoshop
YouTube

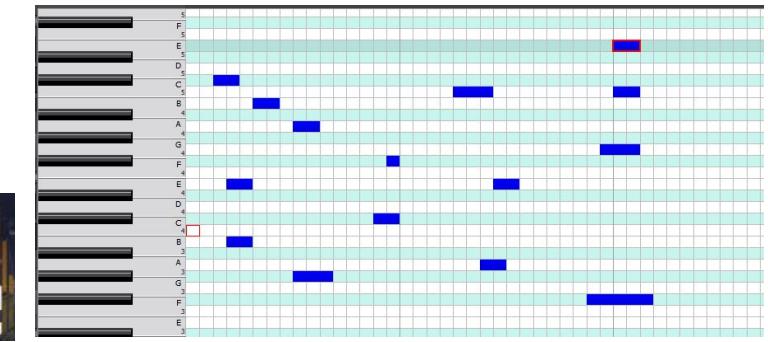
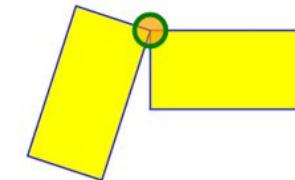


Adobe
Illustrator

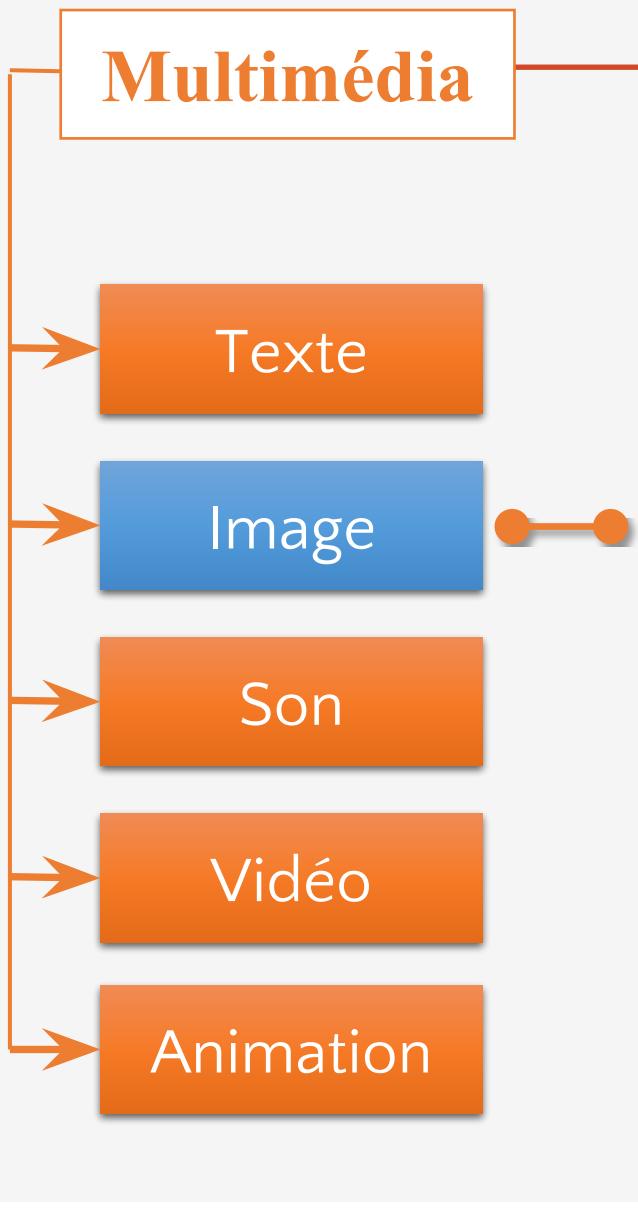
AVI, MOV,
MPEG4, WAV,
AIF,



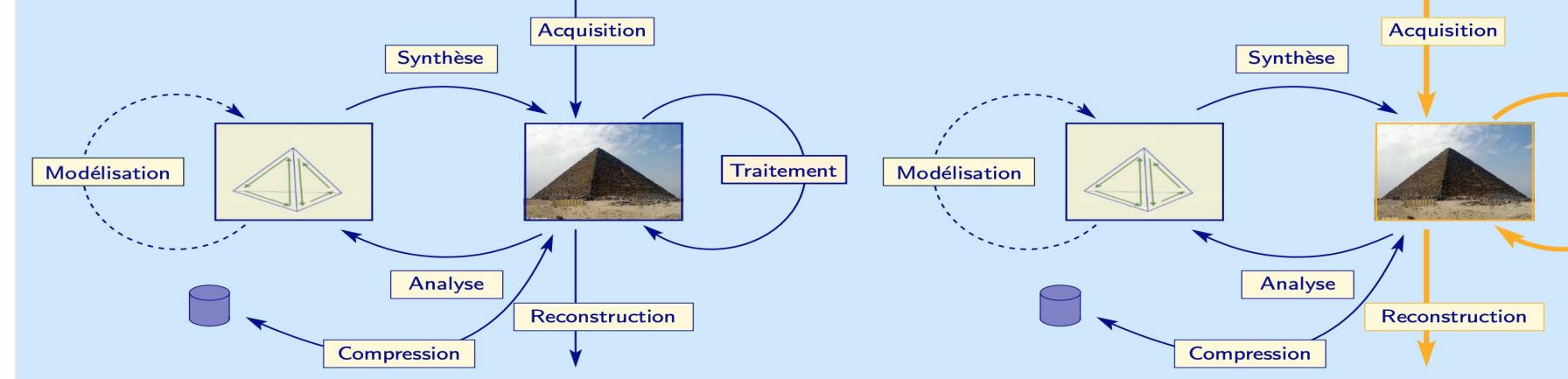
Adobe
Premiere Pro



Composantes du Multmédia



Traitement d'images



Le traitement d'images est une technique utilisée pour améliorer, transformer ou analyser des images numériques. Cela inclut diverses applications telles que la reconnaissance faciale, la médecine et la réalité augmentée.

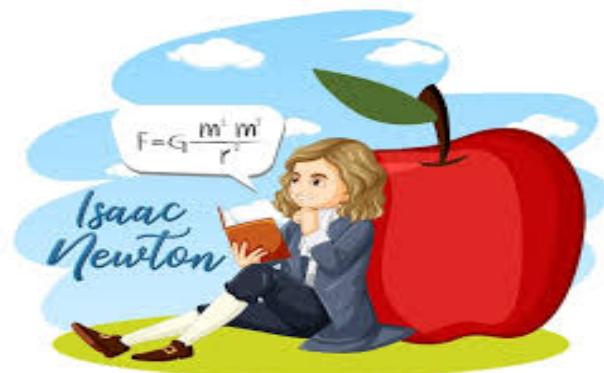


Composantes du Multmédia

Multmédia

- Texte
- Image
- Son
- Vidéo
- Animation

Quel est le sens que j'associe à l'image d'une pomme et pourquoi ?



Composantes du Multmédia

Multmédia

- Texte
- Image
- Son
- Vidéo
- Animation

Images : Bitmaps et vectorielles

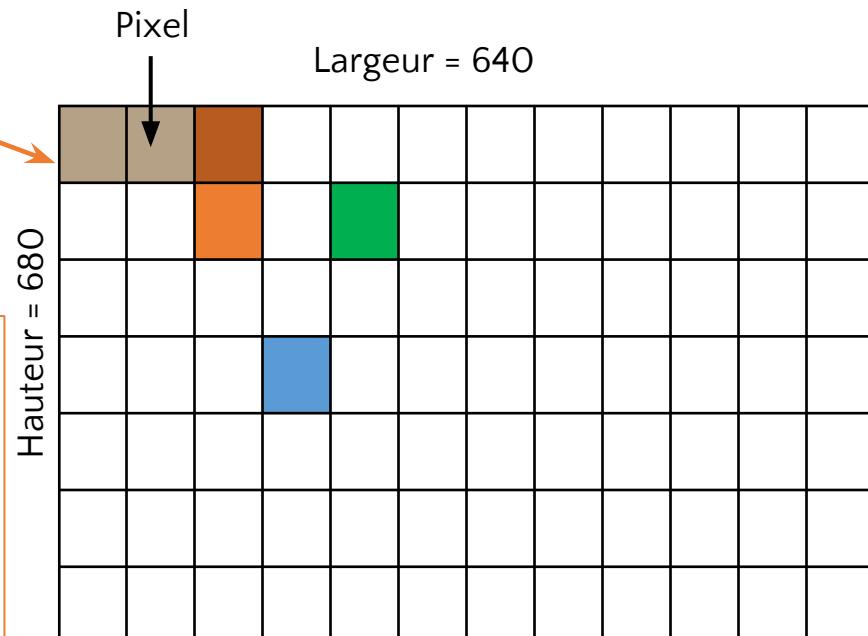
• Images Bitmaps

- Matrice de pixels. Chaque pixel a une couleur
- Caractéristiques :
 - Résolution : Largeur x Hauteur (1920 x 1080 pixels = 2 073 600 pixels => 2 Méga Pixels)
 - Codage de la couleur : 1 bit, 8 Bits, 16 bits, 24 bits, 32 bits
 - Densité : Nombre de pixels / unité de longueur (Dot Per Inch (1p=2.54 cm)) (ppp ou dpi)
 - 72 dpi, 300 dpi, 600 dpi, et plus
- Mode de représentation couleur : RGB(24 bits), CMJN, RGBA (32 bits), TSL, YUV, ...



Exemple : Image UHD

- Size= $3840 \times 2160 = 8 \text{ MP}$
- 24 bits = 3 octets
- $C = 3840 \times 2160 \times 3$
 $= 8\ 294\ 400 \text{ Oct} = 8100 \text{ KO} = 7,91 \text{ MO}$
- Impression à 300 dpi => $32,5 \text{ cm} \times 18,28 \text{ cm}$



Composantes du Multmédia

Multmédia

→ Texte

→ Image

→ Son

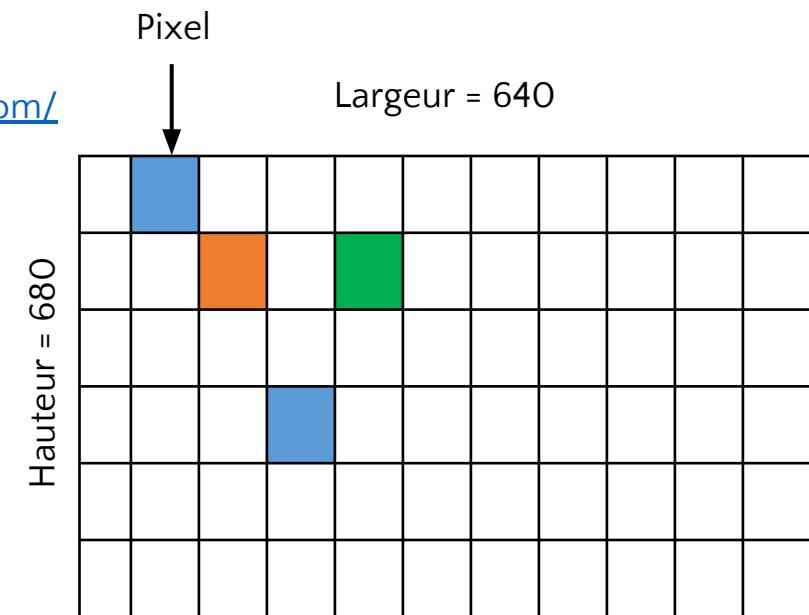
→ Vidéo

→ Animation

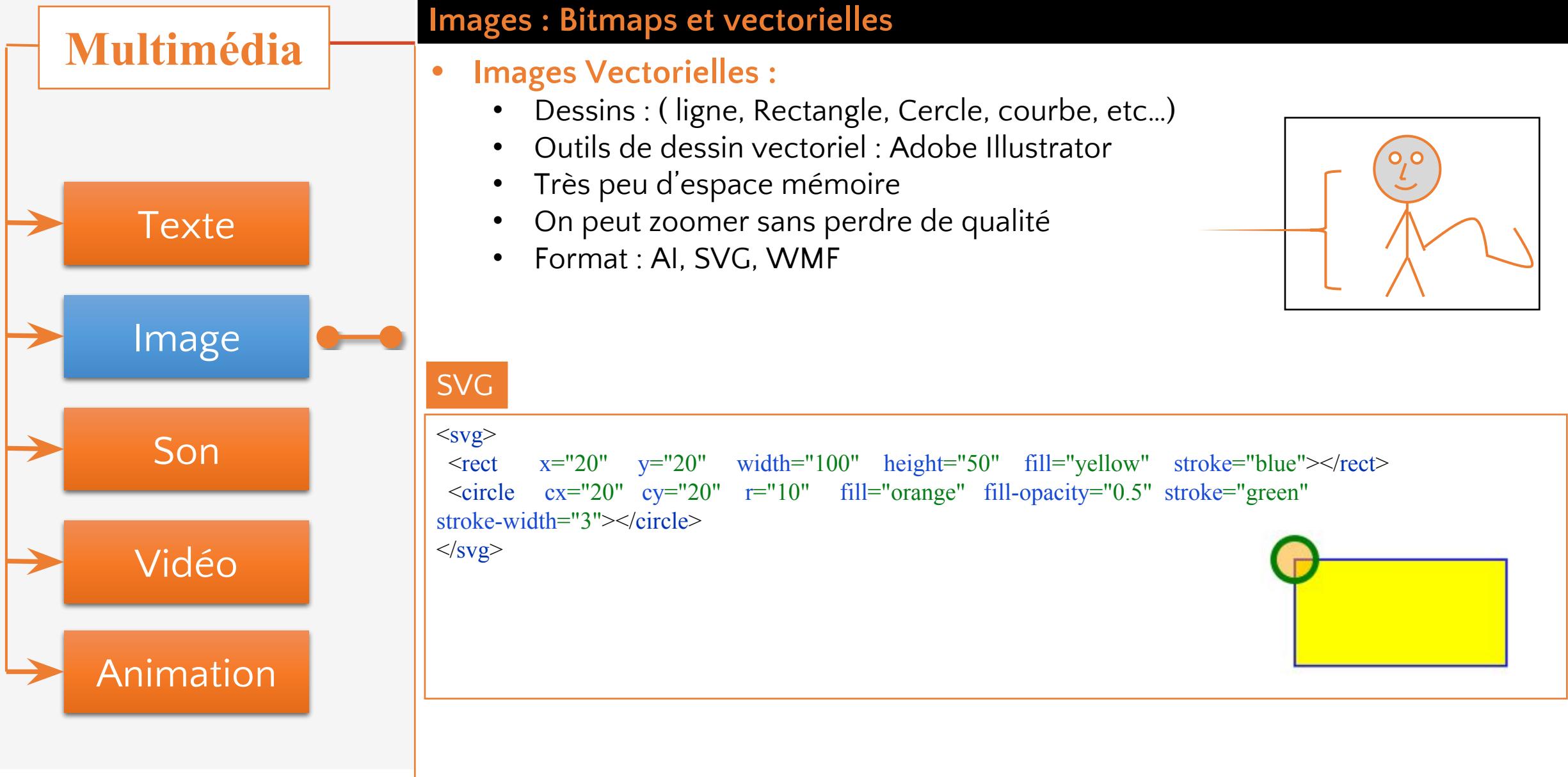
Images : Bitmaps et vectorielles

• Images Bitmaps

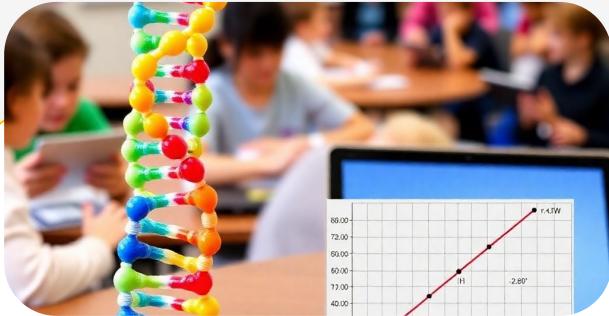
- Beaucoup d'espace mémoire, Zoomer + Fait perdre la qualité;
- Formats BMP, JPEG, PNG, GIF, TIF, PSD
- Compression : Produire le même contenu avec moins d'informations
 - Avec perte ou sans perte
- Quand on change les dimensions d'une image, il faut garder la proportion en L et H pour éviter de déformer l'image
- Outils de traitement d'images :
 - Adobe Photoshop,
 - Windows Paint
 - GIMP : <https://www.gimp.org/>
 - Photopea : <https://www.photopea.com/>



Composantes du Multmédia



Applications du traitement d'images



Visualisation du concept

Modélisation 3D : les images et les animations 3D aident à visualiser des concepts complexes, en particulier en sciences et en mathématiques.

Diagrammes : Des graphiques améliorés facilitent la compréhension de concepts abstraits.



Apprentissage interactif

Applications éducatives : les applications qui utilisent le traitement d'image permettent aux élèves d'interagir avec le contenu, par exemple, en reconnaissant des formes ou des objets. Réalité augmentée : En superposant des informations numériques au monde réel, la réalité augmentée enrichit l'apprentissage.

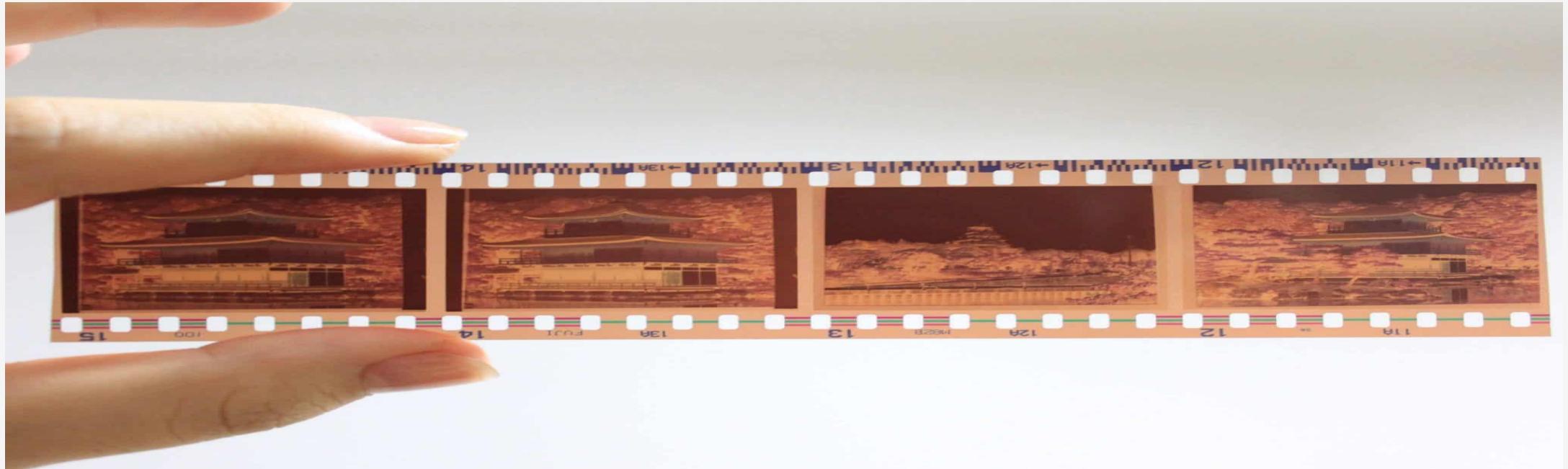


Soutien à la créativité

Édition d'images : Les outils de traitement d'images permettent aux élèves de créer et de modifier des projets visuels, favorisant ainsi l'expression artistique. Projets multimédias : Les élèves peuvent développer des présentations ou des vidéos incorporant des images traitées, ce qui rend l'apprentissage plus engageant.

Traitements d'images en Python

Les tâches courantes du traitement d'images incluent l'affichage des images, les manipulations basiques comme le recadrage, le retournement, la rotation, ou encore la segmentation, la classification et les extractions de caractéristiques, la restauration et la reconnaissance d'images. Python devient un choix judicieux pour de telles tâches de traitement d'images. Cela est dû à sa popularité croissante en tant que langage de programmation scientifique et à la disponibilité gratuite de nombreux outils de traitement d'images de pointe dans son écosystème.



Traitement d'images en Python

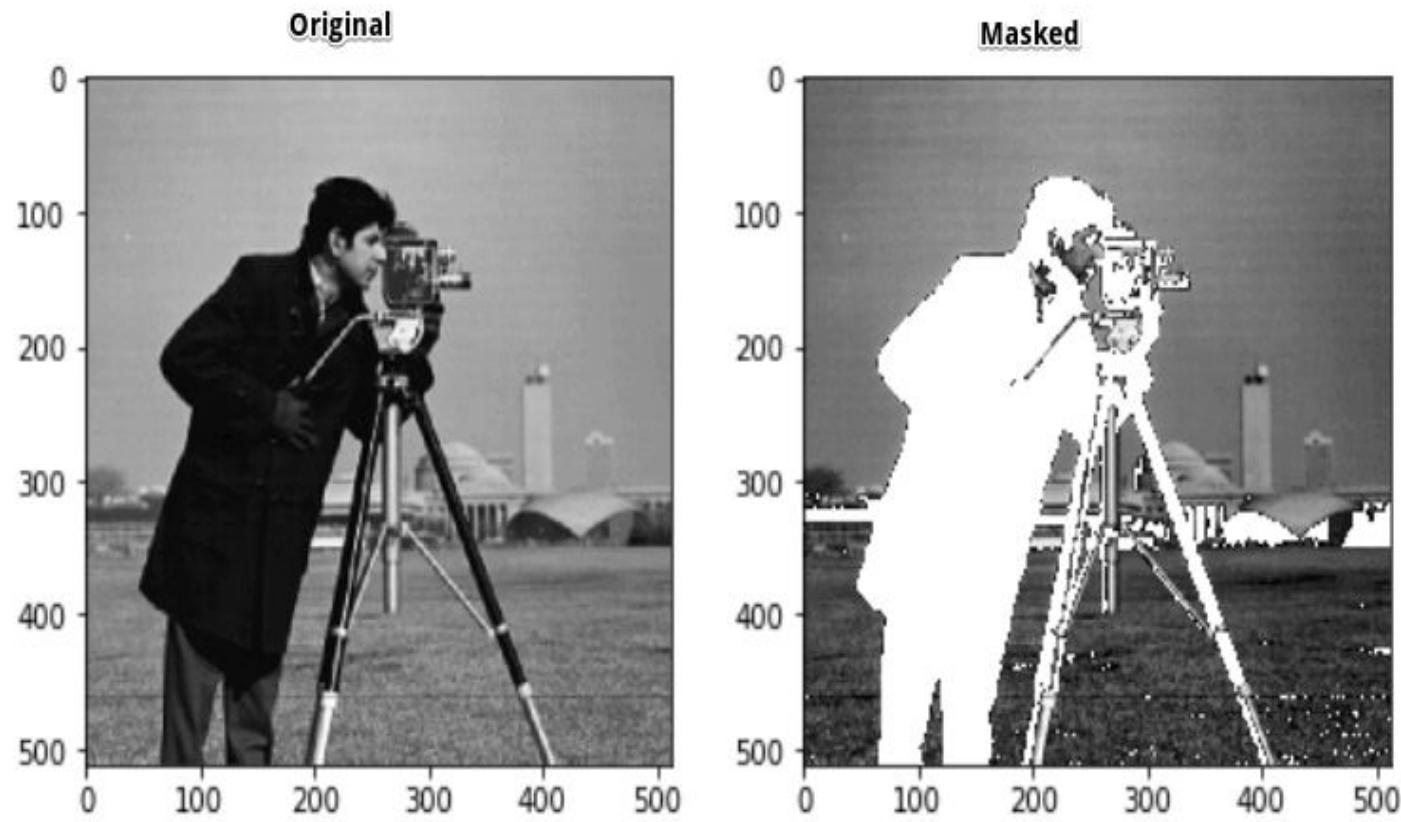
Numpy

Les opérations NumPy de base, telles que le découpage en tranches (slicing), les masques et toute sorte d'indexation, nous pouvons modifier les valeurs de pixel d'une image.

```
import numpy as np
from skimage import data
import matplotlib.pyplot as plt
%matplotlib inline

image = data.camera()
type(image) #Image est un array numpy

mask = image < 87
image[mask]=255
plt.imshow(image, cmap='gray')
```



Traitements d'images en Python

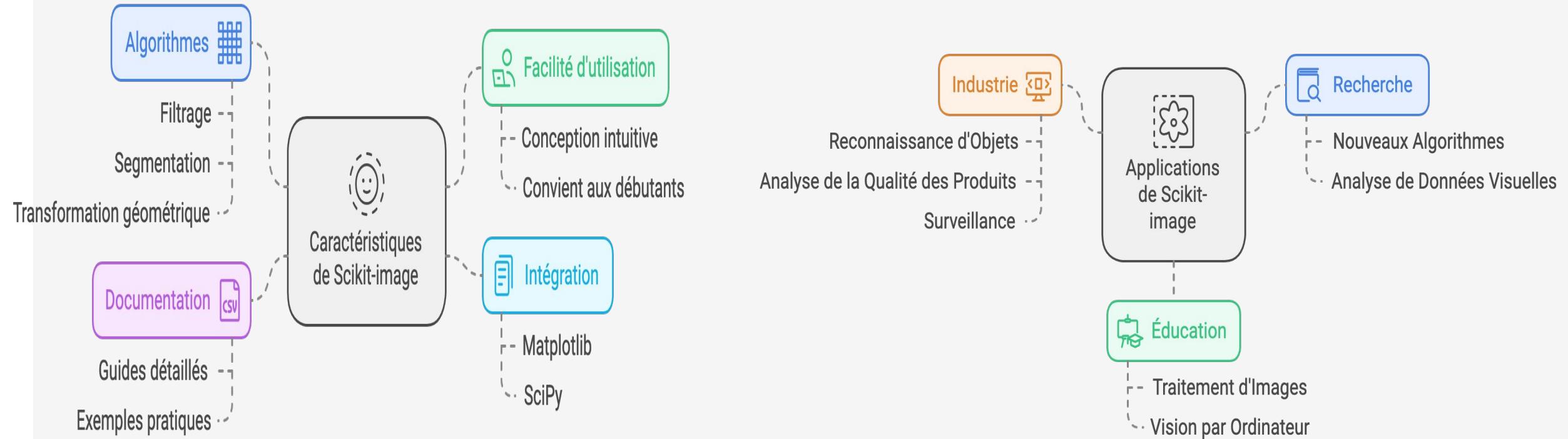
Scikit Image

Scikit-image est un paquet open source Python qui facilite le traitement d'images en utilisant des tableaux numpy. Ses applications dans divers domaines, ainsi que sa simplicité d'utilisation, ce qui en fait un choix idéal pour les chercheurs, les éducateurs et les professionnels de l'industrie.

Caractéristiques principales

Applications

Scikit-image est utilisé dans divers domaines, notamment :

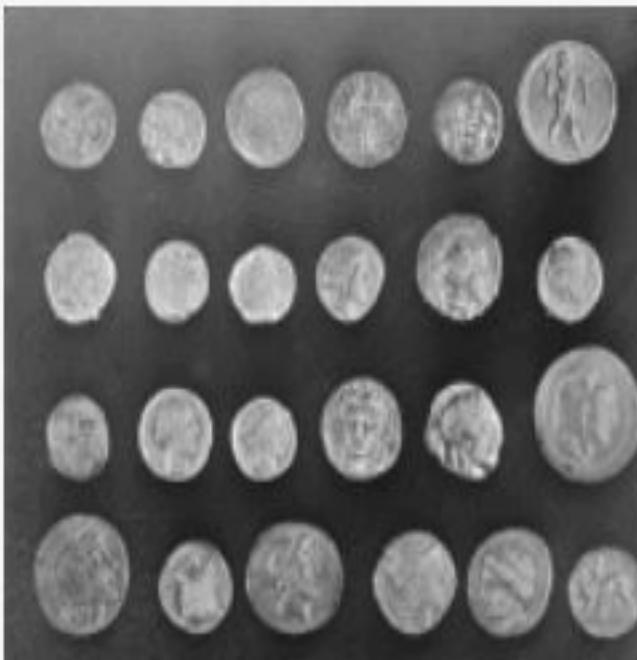


Traitement d'images en Python

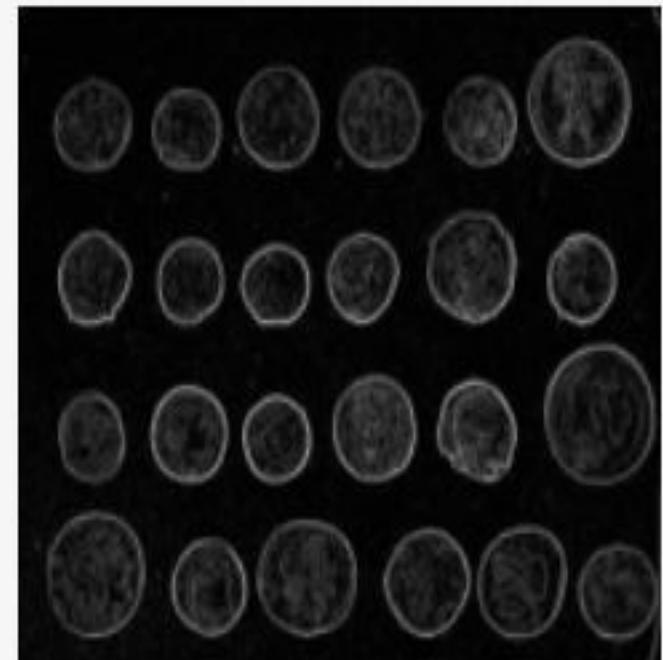
Scikit Image -Exemples

Filtre d'image :

```
import matplotlib.pyplot as plt  
%matplotlib inline  
  
from skimage import data,filters  
  
image = data.coins()  
plt.imshow(image)  
  
edges = filters.sobel(image)  
plt.imshow(edges)
```



Original



Filtered

Traitement d'images en Python

Scikit Image -Exemples

Correspondance de modèles :

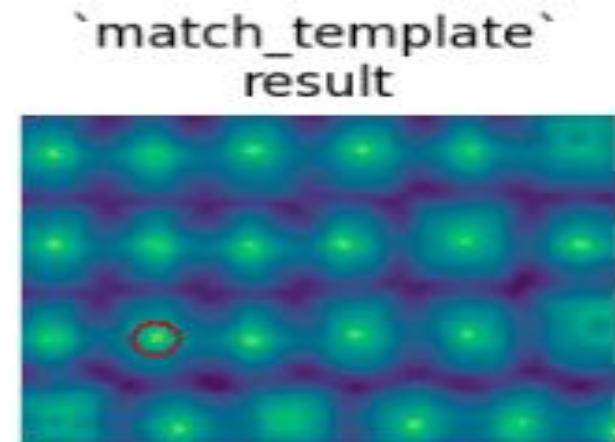
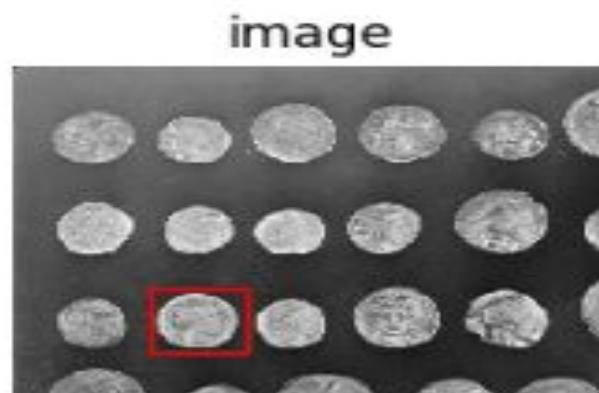
Identifier l'occurrence d'un patch d'image

```
import numpy as np
import matplotlib.pyplot as plt

from skimage import data
from skimage.feature import match_template

image = data.coins()
coin = image[170:220, 75:130]

result = match_template(image, coin)
```



Traitement d'images en Python

PIL/Pillow

PIL (Python Imaging Library) et son fork moderne, Pillow, sont des bibliothèques Python puissantes et flexibles pour le traitement d'images. Elles permettent de manipuler des images de manière simple et efficace, offrant ainsi des fonctionnalités Clés ,en utilisant les opérations courantes que l'on peut réaliser avec cette bibliothèque.

Caractéristiques Clés

- Ouverture d'Images
- Modification d'Images
- Création d'Images
- Enregistrement d'Images

Opérations Courantes

- Redimensionnement
- Recadrage
- Filtrage
- Dessin



Traitement d'images en Python

PIL/Pillow Exemple

Transformations géométriques

La classe `PIL.Image.Image` contient des méthodes pour `resize()` et `rotate()` une image. La première prend un tuple donnant la nouvelle taille, la seconde l'angle en degrés dans le sens inverse des aiguilles d'une montre.



Tourner une image

```
out = im.resize((128, 128))  
out = im.rotate(45) # degrees counter-clockwise
```



Transposer une image

```
out = im.transpose(Image.Transpose.FLIP_LEFT_RIGHT)
```



```
out = im.transpose(Image.Transpose.ROTATE_180)
```



```
out = im.transpose(Image.Transpose.FLIP_TOP_BOTTOM)
```



Traitement d'images en Python

PIL/Pillow Exemple

Redimensionnement

	<code>thumbnail()</code>	<code>contain()</code>	<code>cover()</code>	<code>fit()</code>	<code>pad()</code>
Taille donnée	(100, 150)	(100, 150)	(100, 150)	(100, 150)	(100, 150)
Image résultante					
Taille résultante	100×100	100×100	150×150	100×150	100×150

Traitement d'images en Python

PIL/Pillow Exemple

Amélioration de l'image

Application des

```
from PIL import ImageFilter  
out = im.filter(ImageFilter.DETAIL)
```



Opérations ponctuelles

```
# multiply each pixel by 20  
out = im.point(lambda i: i * 20)
```



Amélioration des images

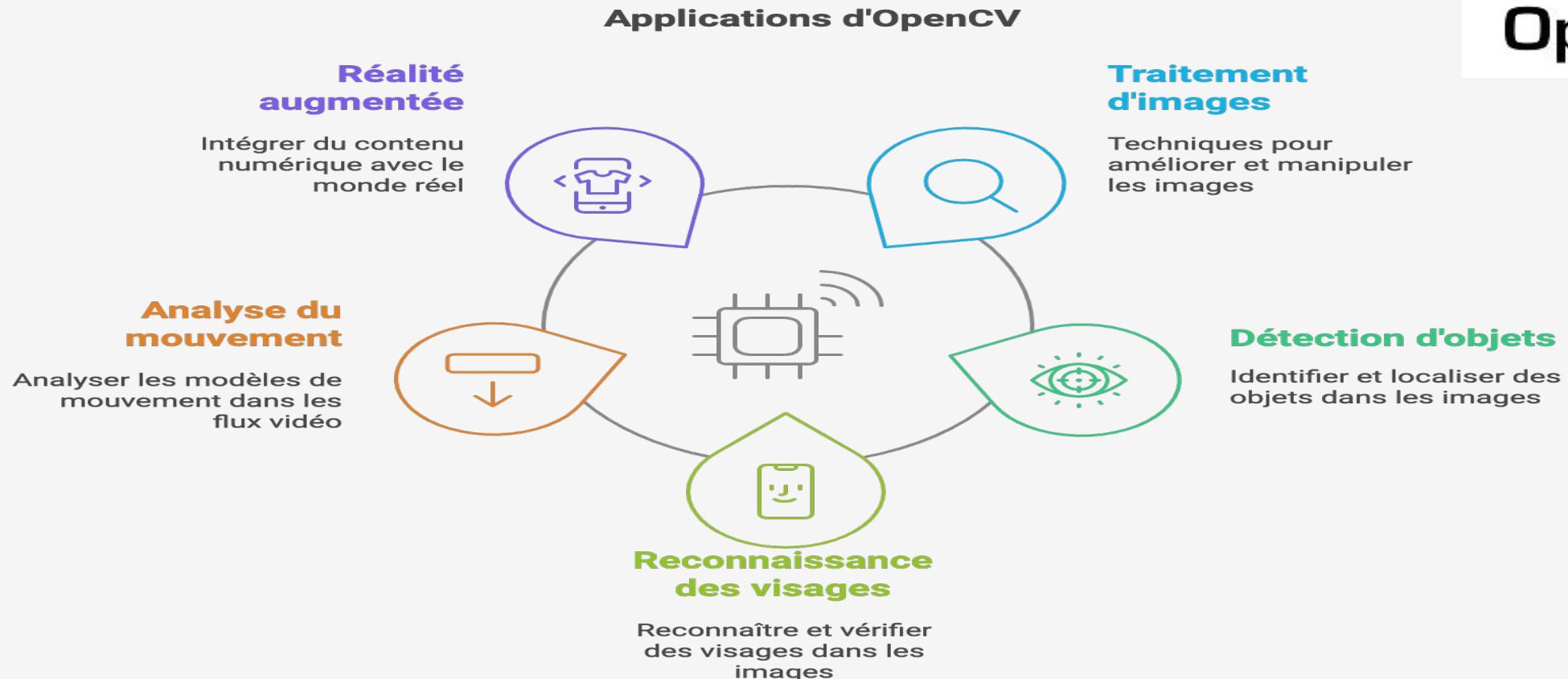
```
from PIL import ImageEnhance  
  
enh = ImageEnhance.Contrast(im)  
enh.enhance(1.3).show("30% more contrast")
```



Traitement d'images en Python

OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque open source très populaire pour le traitement d'images et la vision par ordinateur. Elle contient plus de 2500 algorithmes optimisés, qui peuvent être utilisés pour un large éventail de tâches, notamment :

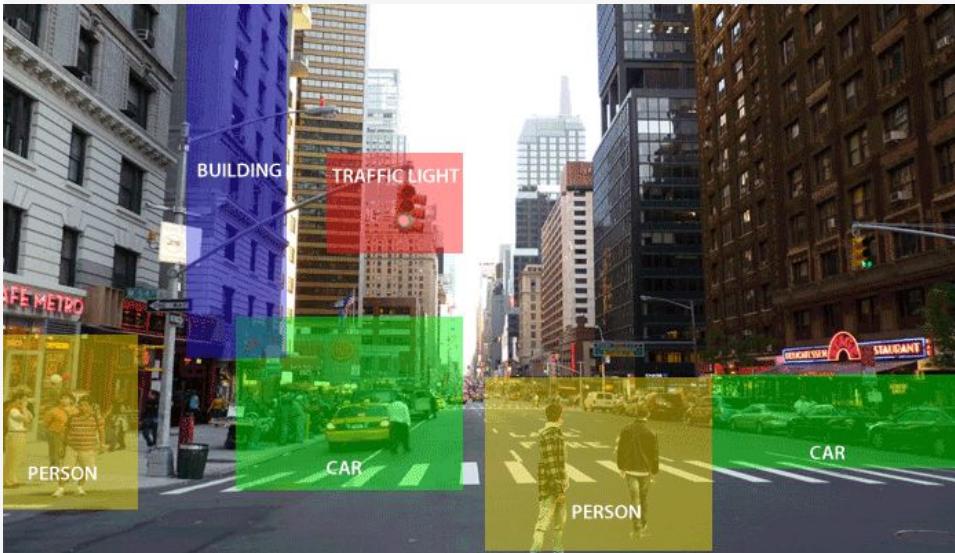


Traitements d'images en Python

OpenCV



Classification des objets :



Identification de l'objet :



Traitement d'images en Python

OpenCV - Accès aux fichiers image

Lire l'image et la visualiser en

```
img1 = cv2.imread("imageA.png")
plot.imshow(img1)
plot.title("lenna - original color image")
```



La convertir en niveau de gris et l'afficher

```
img2 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
plot.imshow(img2, cmap='gray')
plot.title("lenna - gray visualisation")
```



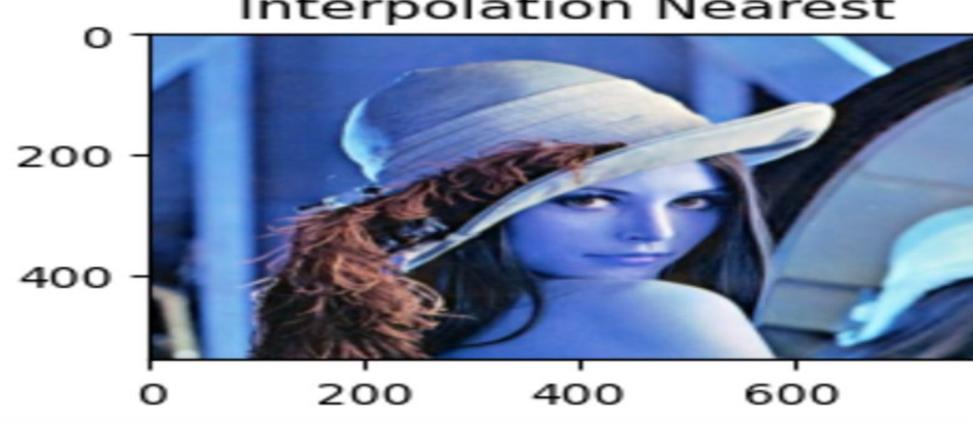
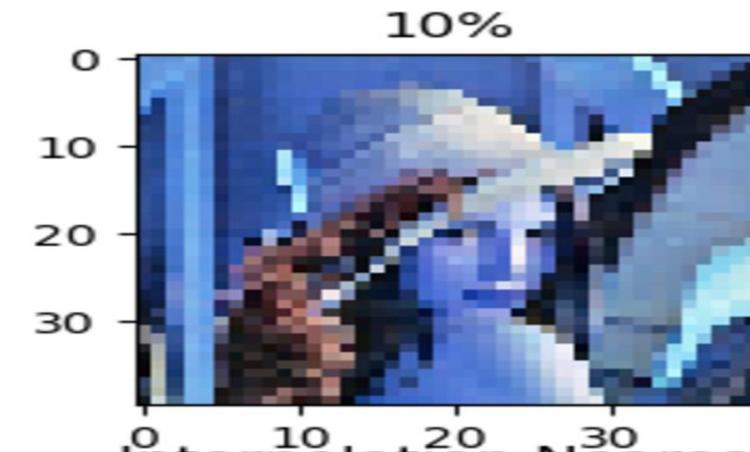
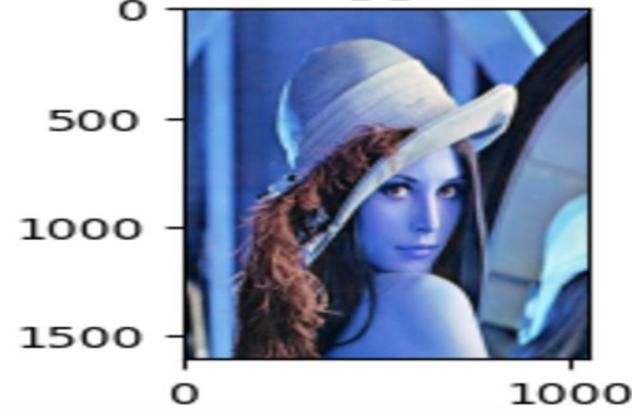
sauvegarder une image

```
# pour sauvegarder l'image, il suffit de mettre la ligne suivante:
cv2.imwrite("nouvelleimage.jpg", image)
```

Traitements d'images en Python

OpenCV- redimensionner

```
mini = cv2.resize(image, (0, 0), fx = 0.1, fy = 0.1)
bigger = cv2.resize(image, (1050, 1610))
stretch_near = cv2.resize(image, (780, 540), interpolation = cv2.INTER_LINEAR)
```



Traitements d'images en Python

OpenCV- Lire une vidéo

```
# Create a VideoCapture object and read from input file
cap = cv2.VideoCapture('A 12-year-old app developer .mp4')

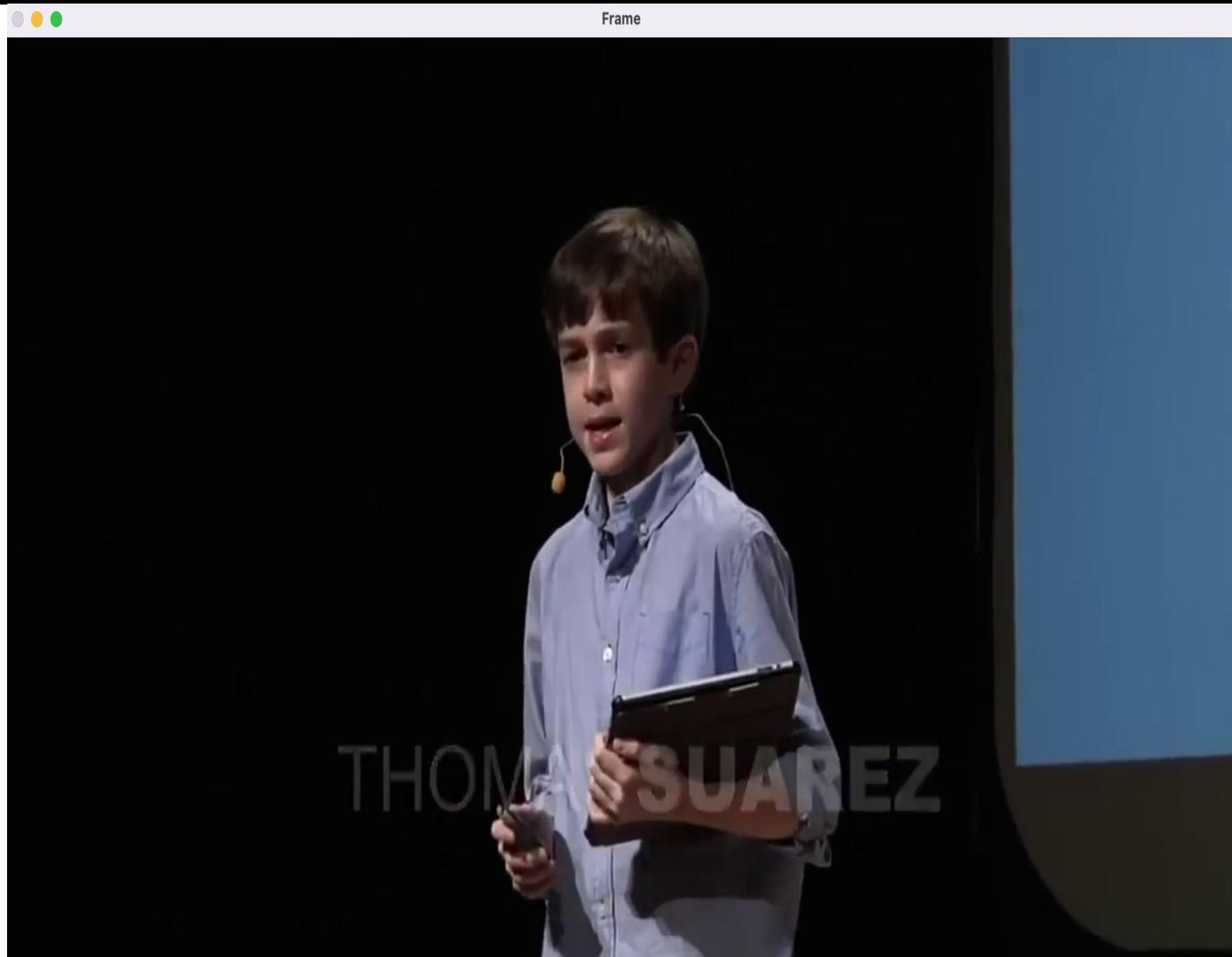
# Check if camera opened successfully
if (cap.isOpened()== False):
    print("Error opening video file")

# Read until video is completed
while(cap.isOpened()):

    # Capture frame-by-frame
    ret, frame = cap.read()
    if ret == True:
        # Display the resulting frame
        cv2.imshow('Frame', frame)

    # Press Q on keyboard to exit
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break

    # Break the loop
else:
    break
```

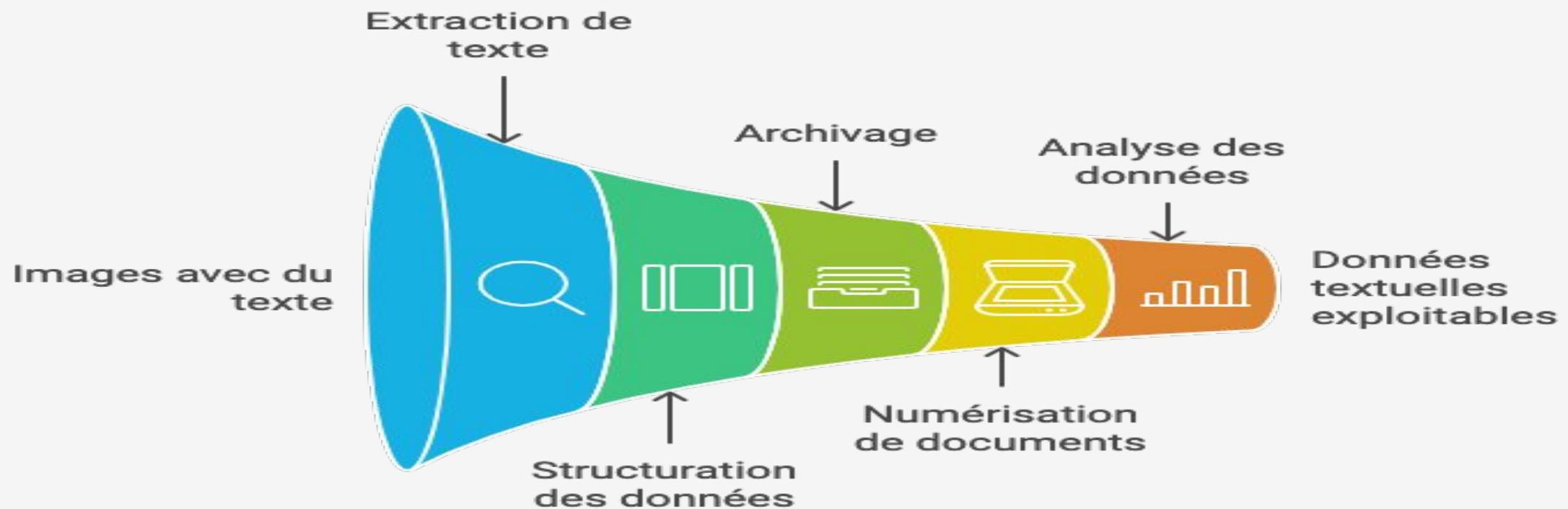


Traitements d'images en Python

OCR (Optical Character Recognition)

La reconnaissance optique de caractères (ROC, ou OCR pour l'anglais optical character recognition), ou océrisation, désigne les procédés informatiques pour la traduction d'images de textes imprimés ou dactylographiés en fichiers de texte.

Transformation des images en données textuelles

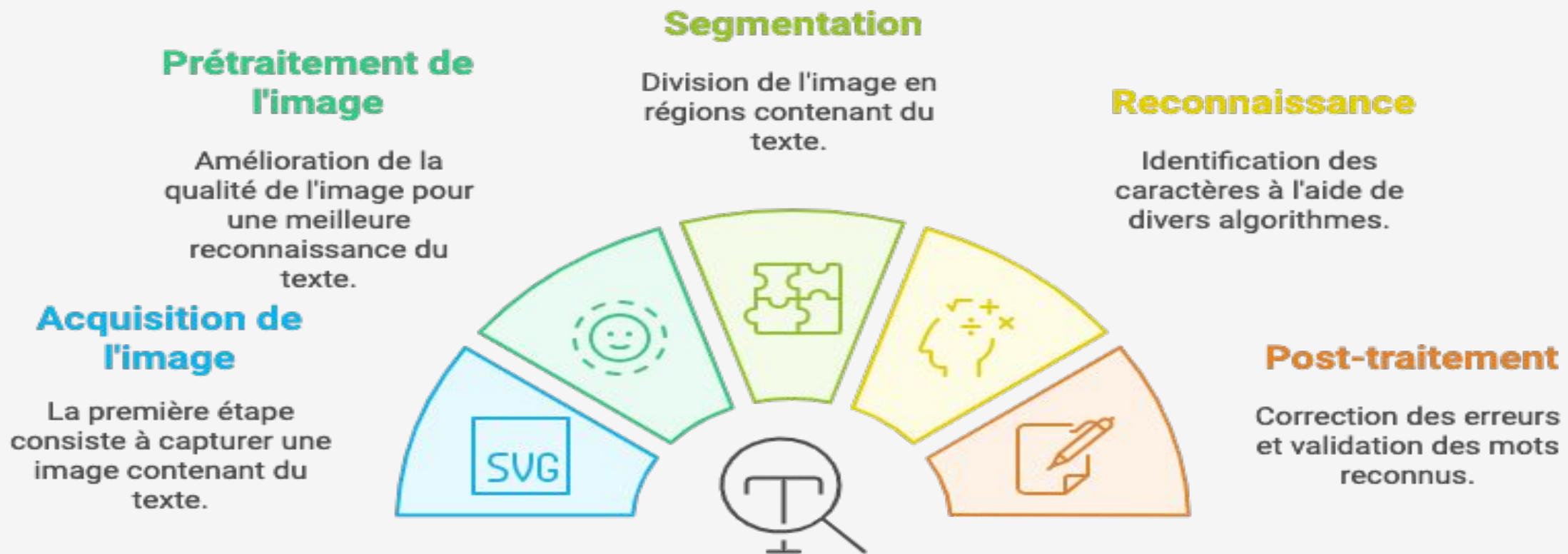


Traitement d'images en Python

OCR - Principe

La reconnaissance de texte repose sur plusieurs étapes clés :

Processus de reconnaissance de texte

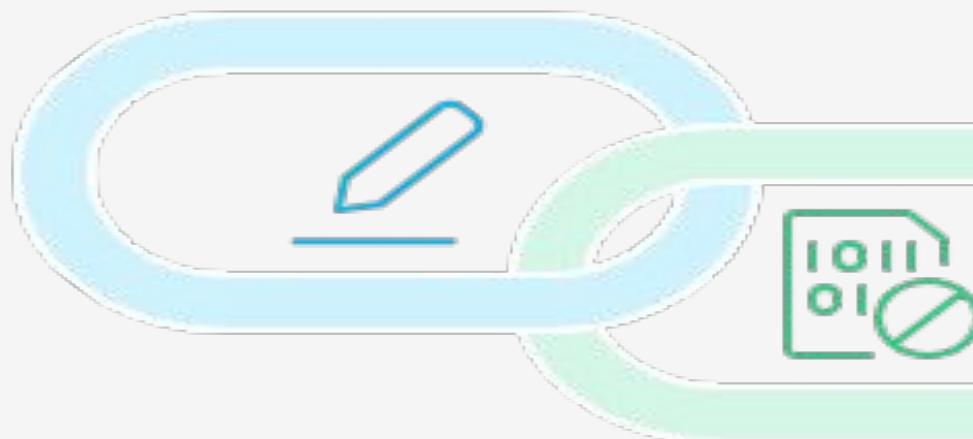


Traitement d'images en Python

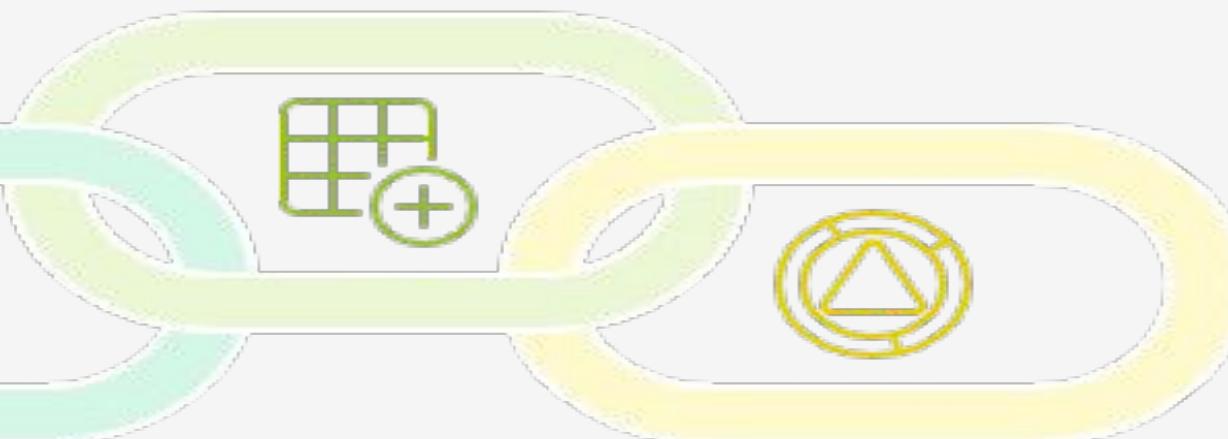
OCR – Pré-traitement de l'image

Pour que l'extraction des données soit précise, la qualité de l'image doit être améliorée. Le processus d'amélioration des images est également connu sous le nom de phase de prétraitement des images. Plus l'image ou le document numérisé est clair et de bonne qualité, plus la sortie de données est précise. Lors de l'étape de pré-traitement, le moteur OCR recherche automatiquement les erreurs et corrige les problèmes. Les techniques souvent utilisées pour améliorer les images ou les documents numérisés sont les suivantes :

Redressement



Zonage



Binarisation



Normalisation



Traitements d'images en Python

OCR – Pré-traitement de l'image- Redressement

Processus par lequel une photo ou un document numérisé est redressé et l'angle corrigé.:

Input
Skewed



Output
De-skewed

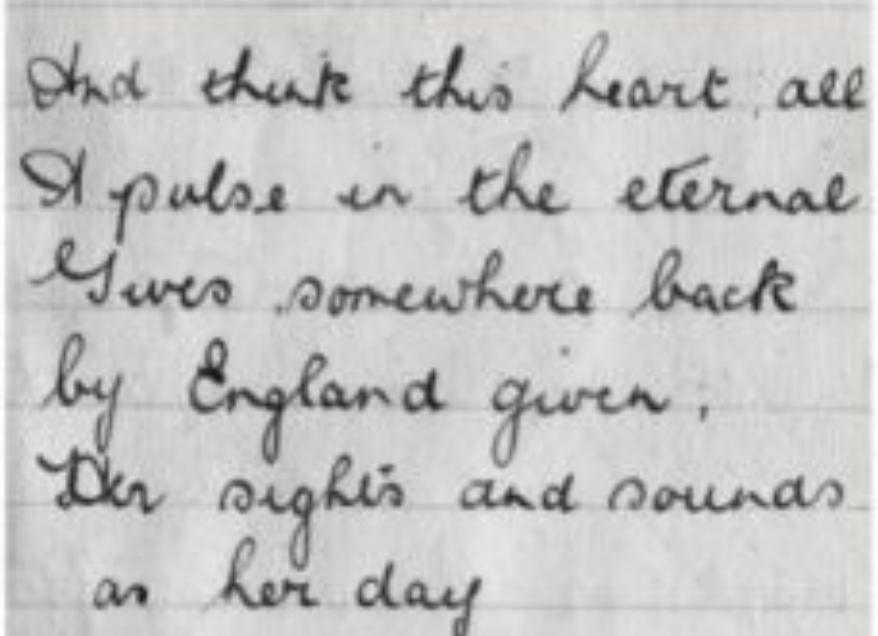


Traitement d'images en Python

OCR – Pré-traitement de l'image- Binarisation

Processus par lequel une image ou un document numérisé est converti en noir et blanc. La binarisation permet de séparer plus précisément le texte de l'arrière-plan.

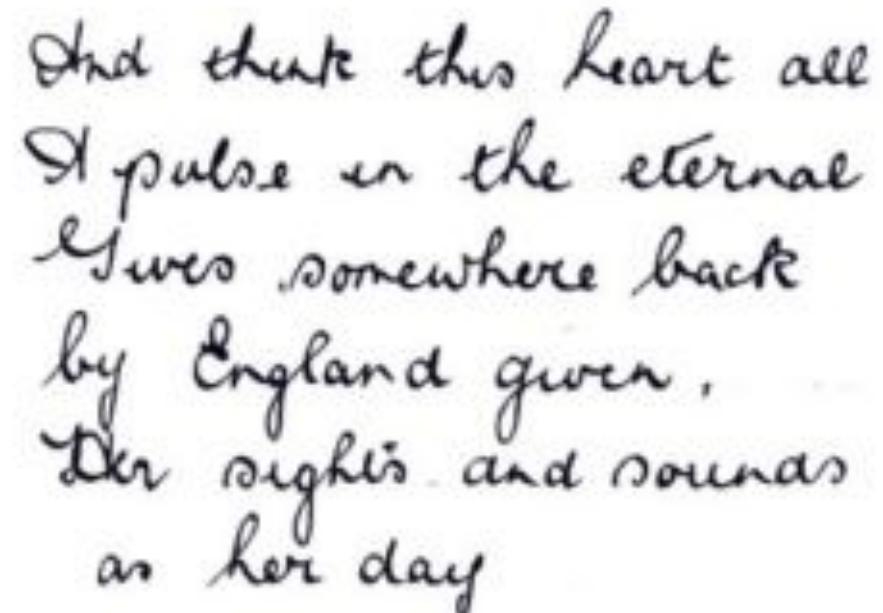
Input



Dad that this heart all
I pulse in the eternal
I was somewhere back
by England given.
The sights and sounds
as her day



Output



Dad that this heart all
I pulse in the eternal
I was somewhere back
by England given.
The sights and sounds
as her day

Traitement d'images en Python

OCR – Pré-traitement de l'image- Zonage

Également connu sous le nom d'analyse de la mise en page, utilisé pour identifier les colonnes, les rangées, les blocs, les légendes, les paragraphes, les tableaux et autres éléments.

The image shows a Purchase Order document from Klappa B.V. on the left and its processed version on the right. The original document contains company details, delivery address, and a table of items. The processed version highlights the 'Purchase Order' number with a green box and the text 'OCR ZONING' below it.

Purchase Order

Klappa B.V.
Löbeckweg 2
9723 HE Groningen
The Netherlands
+31 502119631

Delivered to:
Ditzain B.V.
Europaweg 197
7766 AH Drenthe
The Netherlands
+31 71 853 7837

QTY	Description	Unit Price	Amount
1	Item 1	€1000,00	€1000,00
1	Item 2	€2300,00	€2300,00
2	Item 3	€100,00	€200,00
	Total:		€3500,00

Jane Doe - Financial Controller

PO number

Purchase Order: 5141240124

OCR ZONING

Traitement d'images en Python

OCR – Pré-traitement de l'image- Normalisation

Processus dédié à la réduction du bruit en ajustant la valeur de l'intensité des pixels aux valeurs moyennes des pixels environnants.

Input
Noisy

2xBitburger alkoholfre	*	3,60	7,20
1xCoca-Cola 0,5L	*	3,95	3,95
1xProsciutto e Funghi	*	10,90	10,90
1xConchiglie Granchi	*	11,95	11,95
Total in EUR			
134,00			
Zahlungsartsart:			
Bar		34,00	
MwSt %	Netto	MwSt	Brutto
8,5 %	91,76	1,00	93,00



Output
Normalized

2xBitburger alkoholfre	*	3,60	7,20
1xCoca-Cola 0,5L	*	3,95	3,95
1xProsciutto e Funghi	*	10,90	10,90
1xConchiglie Granchi	*	11,95	11,95
Total in EUR			
134,00			
Zahlungsartsart:			
Bar		34,00	
MwSt %	Netto	MwSt	Brutto
8,5 %	91,76	1,00	93,00

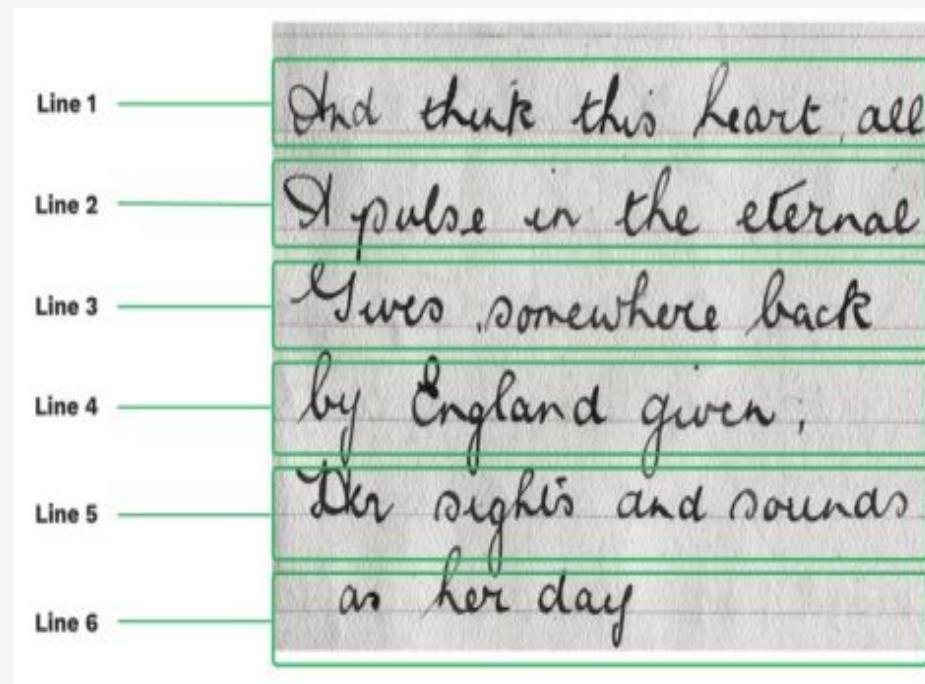
Traitements d'images en Python

OCR – Segmentation

La segmentation est le processus de reconnaissance d'une ligne de texte à la fois. La segmentation comprend les étapes suivantes :

Détection des mots et des lignes de texte

Il s'agit de l'identification des lignes de texte et des mots qui leur appartiennent.



Reconnaissance du script

Processus d'identification du script à partir de documents, de pages, de lignes de texte, de paragraphes, de mots et de caractères.



Traitement d'images en Python

OCR – Reconnaissance de caractères

Deux approches sont utilisées dans l'étape de reconnaissance des caractères :

La mise en correspondance de matrices

Chaque caractère est comparé à une bibliothèque de matrices de caractères. Le modèle OCR effectue une comparaison pixel par pixel afin d'identifier le caractère qui lui correspondant

Reconnaissance des caractéristiques

Le processus identifie les caractéristiques des caractères à partir d'images. Par exemple, la taille, la hauteur, la forme, les lignes et la structure d'un caractère sont comparées à celles de la bibliothèque existante.

Étapes de la reconnaissance de texte

Reconnaissance des caractéristiques

Identifier les motifs de texte et les caractéristiques des caractères

Correspondance de matrices

Comparer les caractères à une bibliothèque de matrices



Traitement d'images en Python

OCR – Post-traitement du résultat

Cette étape concerne les techniques et les algorithmes qui améliorent la précision de l'extraction des données pour un résultat optimal. Tout d'abord, les données sont détectées, puis corrigées si nécessaire. Les données extraites sont comparées à un vocabulaire ou à une bibliothèque de caractères pour des vérifications grammaticales et des considérations contextuelles afin de compléter la phase de post-traitement.

And think this heart all
A pulse in the eternal
Gives somewhere back
by England given,
Her sights and sounds
on her day

And thik this heart all
A pulse in the eternal
Yives somewhere back
by England given,
Her sighis and sounds
on her day

And think this heart all
A pulse in the eternal
Gives somewhere back
by England given,
Her sights and sounds
on her day

Traitements d'images en Python

OCR -Cas d'utilisation

Lecture de plaques d'immatriculation :

L'OCR est utilisé pour la lecture des plaques d'immatriculation des voitures dans des environnements de sécurité, tels que les parkings, les péages et les systèmes de surveillance de la circulation.

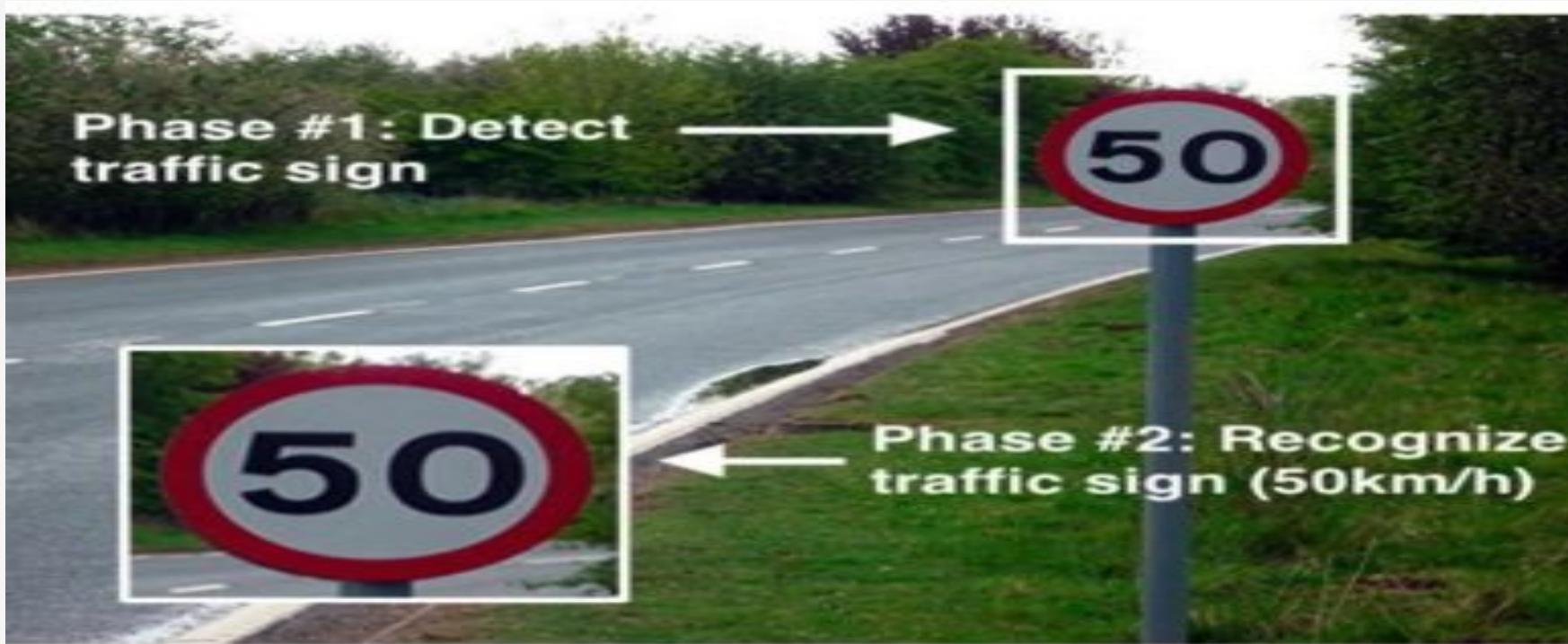


Traitement d'images en Python

OCR –Cas d'utilisation

La reconnaissance de panneaux de signalisation et de marquages routiers :

Les caméras montées sur les voitures autonomes peuvent être utilisées pour capturer des images de panneaux de signalisation et de marquages routiers, et l'OCR peut être utilisé pour convertir les caractères imprimés sur ces images en format numérique. Cela permet à la voiture de comprendre les limites de vitesse, les arrêts et les directions, entre autres informations..



Traitement d'images en Python

OCR – Les limites

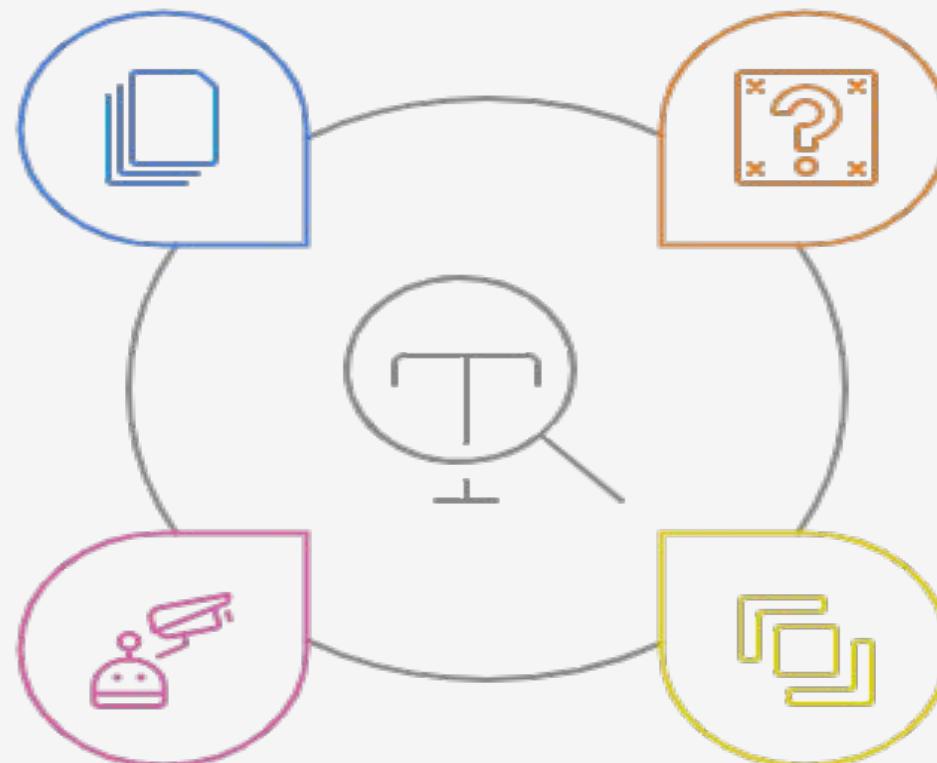
Limitations de l'OCR traditionnel

Variété de documents

L'OCR traditionnel a du mal avec des documents divers et non structurés.

Manque d'automatisation

La création manuelle de règles pour l'extraction de données entrave l'automatisation.



Qualité de l'image

La clarté et la résolution de l'image d'entrée affectent la précision de la reconnaissance de texte.

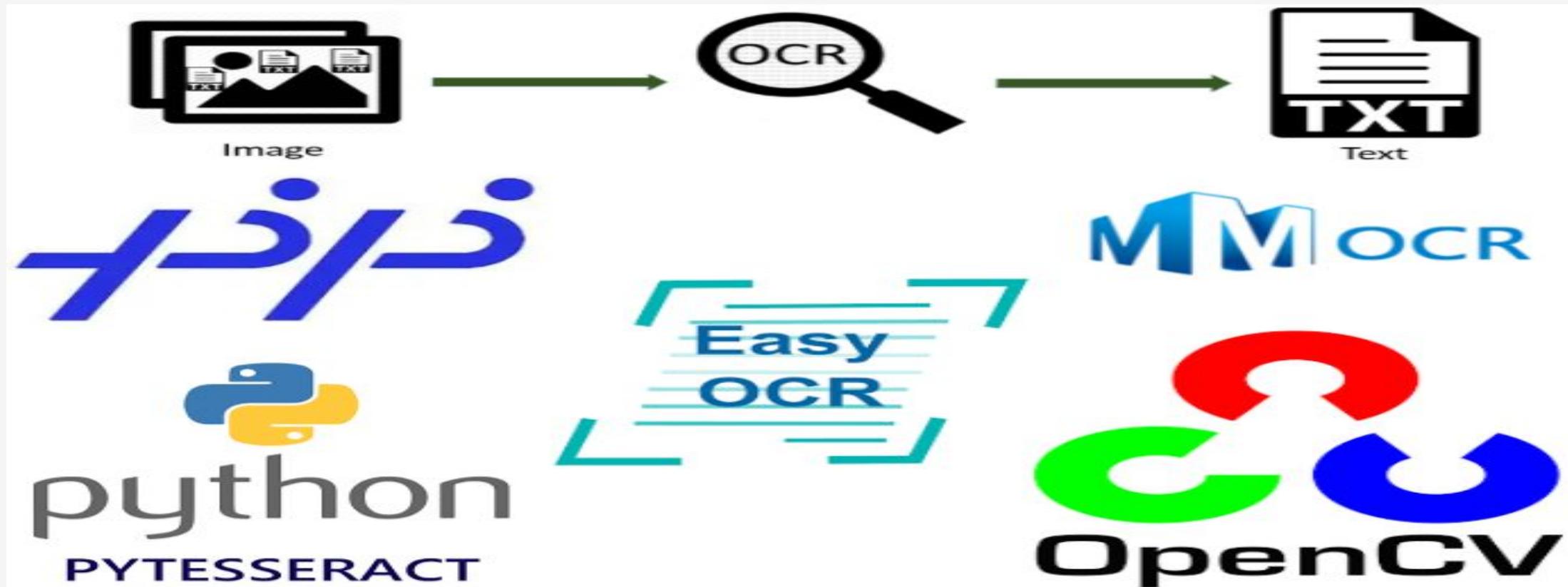
Modèles et règles

Le besoin de modèles et de règles prédéfinis limite la flexibilité de l'OCR.

Traitements d'images en Python

OCR en Python

En Python, il existe plusieurs bibliothèques et outils pour effectuer de la reconnaissance optique de caractères (OCR) sur des images ou des documents numérisés. Voici les deux outils les plus populaires :



Traitement d'images en Python

OCR avec OpenCV

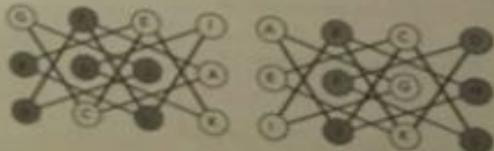
OpenCV est une bibliothèque de traitement d'image open-source qui peut être utilisée pour prétraiter les images avant l'OCR. Elle peut être utilisée pour supprimer le bruit, améliorer la qualité de l'image, détecter les contours ainsi que l'inversion des couleurs de l'image, comme affiché dans cet exemple où on a séparé le fond de l'avant-plan d'une image en implémentant La méthode de seuillage adaptatif de Gauss en utilisant openCV :

```
➊ img = cv2.imread('/content/book_adaptative.jpg')
cv2.imshow(img)
```



Troca de moedas

A primeira figura mostra 6 moedas de prata, A, C, E, G, I e K e 6 moedas de ouro, B, D, F, H, J e L. A sua tarefa é mover as moedas para a disposição mostrada na segunda figura. Cada movimento deve consistir em uma troca entre uma moeda de prata e uma moeda de ouro adjacente; duas moedas são adjacentes se estiverem unidas por uma linha reta. Sabe-se que o menor número de movimentos que resolve este quebra-cabeça é 17. Você consegue encontrar uma solução em 17 jogadas?



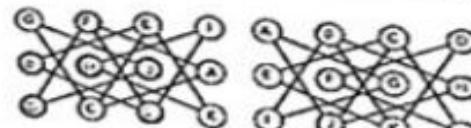
Mova as moedas da primeira posição para a segunda.

```
➋ gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
adaptive_gaussian = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY)
cv2.imshow(adaptive_gaussian)
```



Troca de moedas

A primeira figura mostra 6 moedas de prata, A, C, E, G, I e K e 6 moedas de ouro, B, D, F, H, J e L. A sua tarefa é mover as moedas para a disposição mostrada na segunda figura. Cada movimento deve consistir em uma troca entre uma moeda de prata e uma moeda de ouro adjacente; duas moedas são adjacentes se estiverem unidas por uma linha reta. Sabe-se que o menor número de movimentos que resolve este quebra-cabeça é 17. Você consegue encontrar uma solução em 17 jogadas?



Mova as moedas da primeira posição para a segunda.

Traitement d'images en Python

OCR avec Tesseract

Tesseract est une bibliothèque OCR open-source développée par Google, prenant en charge plus de 100 langues et offrant une grande précision. Elle fonctionne avec Python via la bibliothèque pytesseract. Le processus OCR de Tesseract commence par la conversion de l'image en binaire, suivie d'une analyse des composants pour identifier les contours, qui sont ensuite organisés en lignes et mots. La reconnaissance des mots se fait en deux étapes : une première tentative de reconnaissance suivie d'une analyse supplémentaire pour identifier les mots non reconnus initialement. Enfin, une dernière passe permet de résoudre les ambiguïtés et de localiser les petites majuscules

```
simage = r'image_texte.png'  
image_original = cv2.imread(simage)  
print(pytesseract.image_to_string(image_original, lang='fra'))  
plt.imshow(image_original, 'gray')
```

jaur ceci est un test

Bonjour ceci est un test