



[OpenCV](#) est la bibliothèque [OpenSource](#) de référence en ce qui concerne la [vision par ordinateur](#) (computer vision). En 23 ans d'existence, elle a accumulé plus de 2500 algorithmes optimisés issus de la littérature scientifique tels que [SIFT](#), la [transformée de Hough](#),... Elle permet entre autre :

- L'ouverture, la modification, l'enregistrement et l'affichage de fichiers images et vidéos.
- L'extraction d'informations sur la répartition statistiques des pixel dans une image : moyenne, variance, histogramme.
- La segmentation d'image
- La localisation et l'extraction d'objets.
- L'extraction d'informations (aire, périmètre, couleurs, ...) sur les objets contenus dans une images.



[OpenCV](#) est la bibliothèque [OpenSource](#) de référence en ce qui concerne la [vision par ordinateur](#) (computer vision). En 23 ans d'existence, elle a accumulé plus de 2500 algorithmes optimisés issus de la littérature scientifique tels que [SIFT](#), la [transformée de Hough](#),... Elle permet entre autre :

- La détection de contours (Filtre de Sobel, laplacien, Scharr, ...), de ligne ou de cercle (Transformée de Hough).
- Opérations d'image usuelles (redimensionnement, rotation, opérateurs morphologiques, filtrage, etc).
- Détection et suivi en temps réels d'objets (yeux, voiture, balle, ...)
- La mise en œuvre d'algorithmes usuels du machine learning tels que par le perceptron multicouches ou encore les arbres de décisions.



OpenCV bibliothèque favorite?

L'adoption massive d'OpenCV par la communauté de l'analyse et du traitement d'images s'explique par les facteurs suivants :

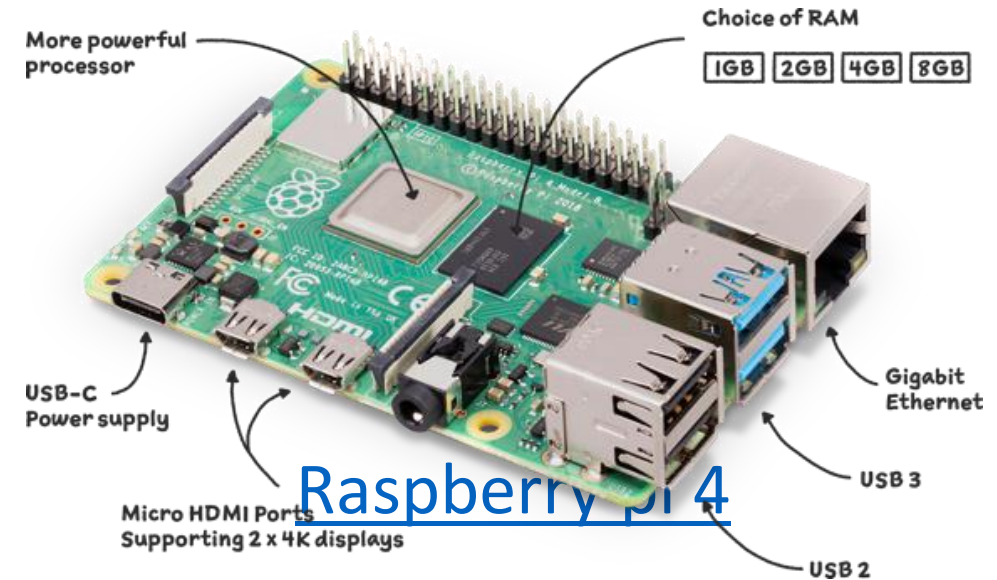
- Il s'agit d'une bibliothèque gratuite et simple d'utilisation
- Un grand nombre de ressources gratuites en ligne (articles, tutoriels, livres, etc ...)
- Ses codes sources sont ouverts . La communauté peut donc participer de manière active à son amélioration et à l'ajout de nouvelles fonctionnalités.
- Elle est disponible dans de nombreux langages différents (Python, Java, Matlab, C#, C++, JavaScript, etc.).
- Elle est portable et s'exécute donc sur de nombreuses plateformes différentes telles que les [systèmes de vision embarqués](#) ([Raspberry pi 4](#), [Intel latte pandas](#), [NVIDIA Jetson Nano](#), etc...)

OpenCV bibliothèque favorite?



Jetson Nano Developer Kit

NVIDIA® Jetson Nano™ Developer Kit is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. All in an easy-to-use platform that runs in as little as 5 watts.



TP1



Installer Python et OpenCV sur Windows

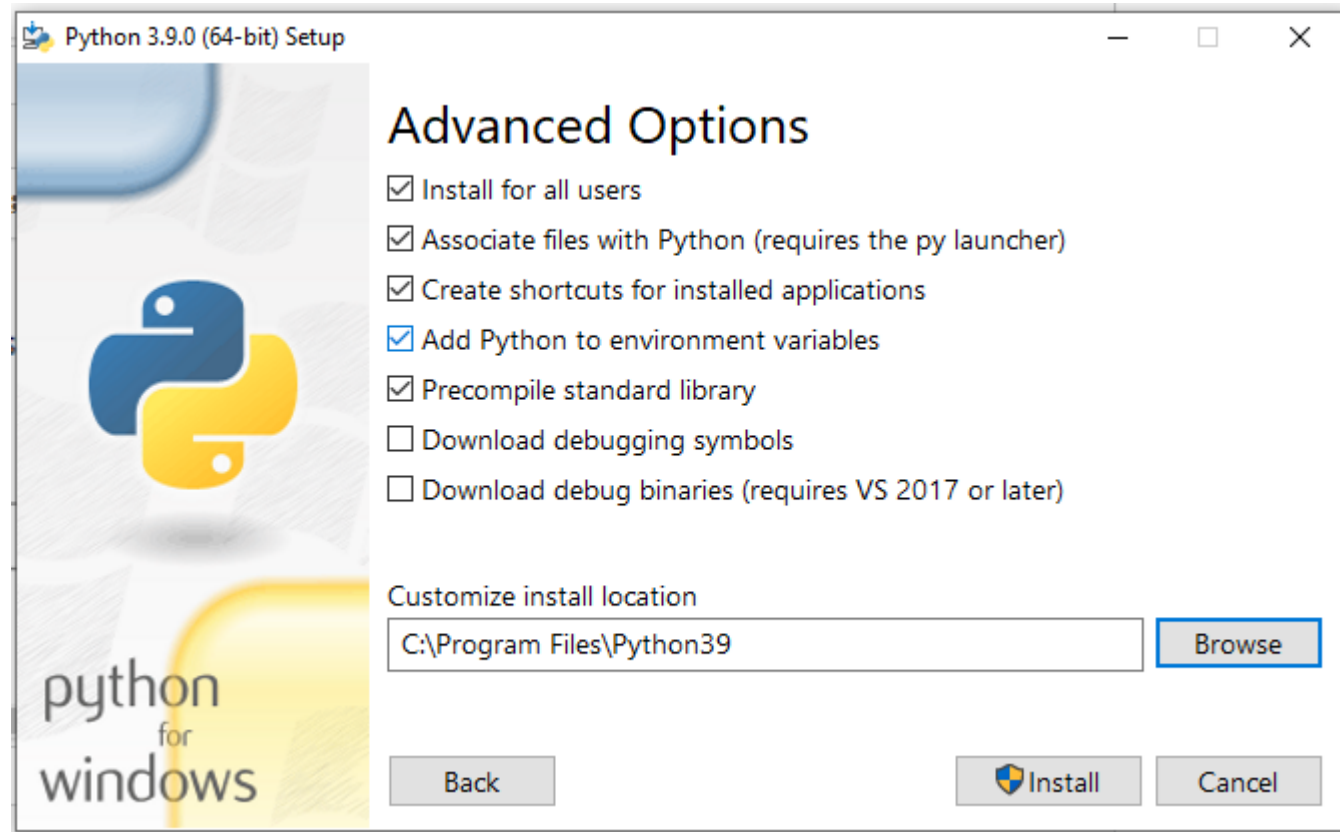
#1 Installer Python 3

Rendez-vous sur <https://www.python.org/downloads/>

Téléchargez la dernière version de Python disponible en 64bits

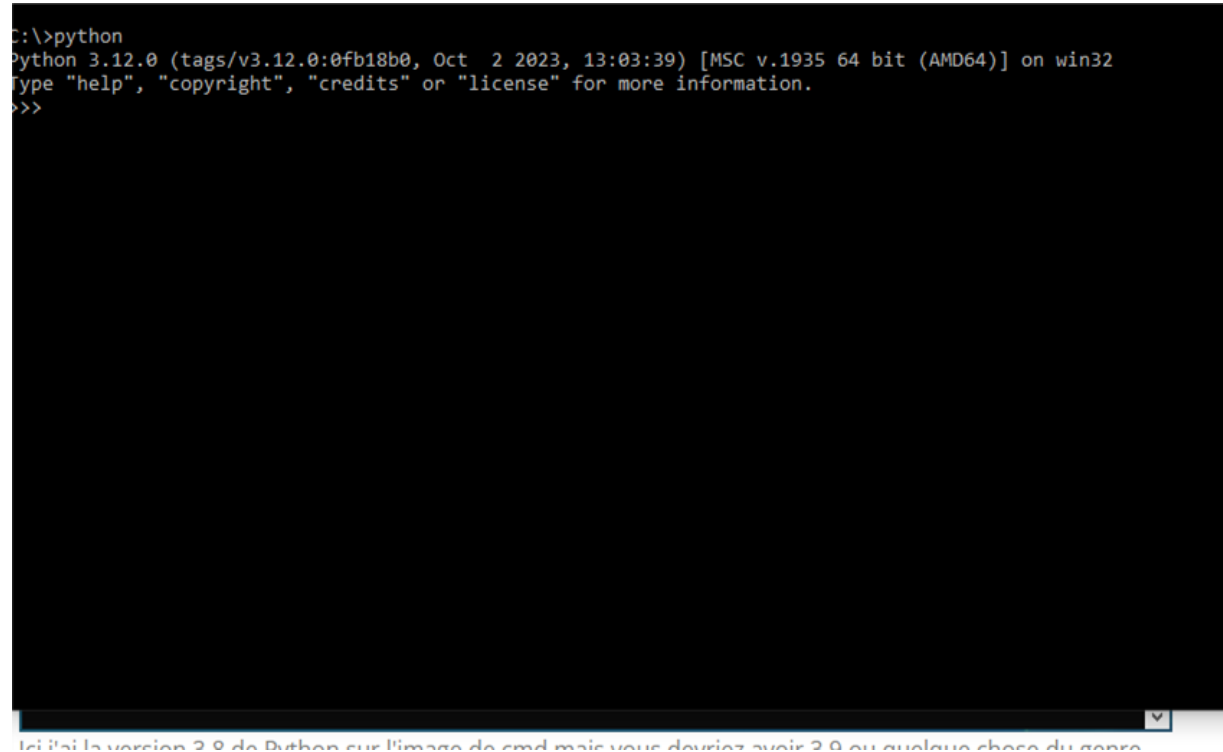
A screenshot of the Python.org website homepage. The header features the Python logo, a search bar with the text "Chercher", and buttons for "Faire un don", "ALLER", and "Socialiser". Below the header is a navigation bar with links: "À propos", "Téléchargements", "Documentation", "Communauté", "Réussites", "Nouvelles", and "Événements". The main content area has a heading "Téléchargez la dernière version pour Windows" and a button "Télécharger Python 3.9.1". Below this, there is text about downloading Python for other operating systems (Windows, Linux / UNIX, Mac OS X, autre) and links to "Avant-premières" and "images Docker". At the bottom, there is text about Python 2.7 and links to "spécifiques". The background of the main content area features an illustration of two parachutes with cargo boxes hanging from them, set against a blue sky with clouds.

Double-cliquez sur le fichier fraîchement téléchargé. Cliquez sur "Customize installation", "Advanced Options", cochez "Install for all users", "Associate files with Python", "Add Python to environment variables" and "Precompile standard library".



Cliquez sur Install, une nouvelle s'affiche pour vous dire que Python s'est bien installé puis appuyez sur "Close".

- Accédez à l'invite de commande en tapant la commande **cmd**
- tapez **python**



```
C:\>python
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

- Tapez "quit()" (toujours sans ") et passez à l'étape #2 :

#2 Installer Open CV

Accéder à l'invite de commandes et tapez les ligne suivantes :

python -m pip install numpy

Pip est un système de gestion de paquets utilisé pour installer et gérer des librairies écrites en Python

python -m pip install imutils

python -m pip install opencv-contrib-python

C'est terminé :) ! OpenCV est installé sur votre ordinateur Windows !!

Testez votre installation

```
import cv2
import numpy as np
img=cv2.imread('C:/Users/intel i7/Desktop/image.jpeg',1)
img = cv2.resize(img,(500,500))
cv2.imshow("image", img)
cv2.waitKey(0)
```

Read color image using imread()

The syntax of `cv2.imread()` function is given below.

`cv2.imread(/path/to/image, flag)`

where `/path/to/image` has to be the complete absolute path to the image. The flag is optional and one of the following possible values can be passed for the flag.

- **`cv2.IMREAD_COLOR`** reads the image with RGB colors. This is the default value for the flag when no value is provided as the second argument for `cv2.imread()`.
- **`cv2.IMREAD_GRAYSCALE`** reads the image as grey image. If the source image is color image, grey value of each pixel is calculated by taking the average of color channels, and is read into the array.
- **`cv2.IMREAD_UNCHANGED`** reads the image as is from the source. If the source image is an RGB, it loads the image into array with Red, Green and Blue channels. If the source image is ARGB, it loads the image with three color components along with the alpha or transparency channel.

1. Read color image using imread()

```
import cv2
#read image
img = cv2.imread('D:/image-1.png')
#print its shape
print('Image Dimensions :', img.shape)
```

Output : Image Dimensions : (400, 640, 3)

2. Read image as greyscale

```
import cv2
img = cv2.imread('D:/image-1.png', cv2.IMREAD_GRAYSCALE)

print('Image Dimensions :', img.shape)
```

Output : Image Dimensions : (400, 640)

Examples

3. Read image with transparency channel

```
import cv2  
img = cv2.imread('D:/image-1.png', cv2.IMREAD_UNCHANGED)  
print('Image Dimensions :', img.shape)
```

We have read all the four channels of the image. Namely Red, Green, Blue and Transparency.

imread() and Color Channels

imread() decodes the image into a matrix with the color channels stored in the order of Blue, Green, Red and (Transparency) respectively.

If (400, 640, 4) is the shape of the image, then

- (:, :, 0) represents Blue channel
- (:, :, 1) represents Green channel
- (:, :, 2) represents Red channel
- (:, :, 3) represents Transparency channel

TP1

Save Image



The syntax of imwrite() function is: **cv2.imwrite(path, image)**

- where path is the complete path of the output file to which you would like to write the image numpy array.
- cv2.imwrite() returns a boolean value. True if the image is successfully written and False if the image is not written successfully to the local path specified.

Example

```
import cv2
```

```
#read image as grey scale
```

```
img = cv2.imread('D:/image-1.png')
```

```
#do some transformations on img and #save matrix/array as image file
```

```
isWritten = cv2.imwrite('D:/image-2.png', img)
```

```
if isWritten:
```

```
    print('Image is successfully saved as file.')
```

Resize Image

Syntax : `cv2.resize(src, dsize[, dst[, fx[, fy[, interpolation]]])` , where

- `src` is the source, original or input image in the form of numpy array
- `dsize` is the desired size of the output image, given as tuple
- `fx` is the scaling factor along X-axis or Horizontal axis
- `fy` is the scaling factor along Y-axis or Vertical axis
- `interpolation` could be one of the following values : `INTER_NEAREST`, `INTER_LINEAR`, `INTER_AREA`, `INTER_CUBIC`, `INTER_LANCZOS`

TP1

Resize Image



Example 1 : Resize Image using cv2.resize()

```
import cv2
src = cv2.imread('D:/cv2-resize-image-original.png', cv2.IMREAD_UNCHANGED)
scale_percent = 50
#calculate the 50 percent of original dimensions
width = int(src.shape[1] * scale_percent / 100)
height = int(src.shape[0] * scale_percent / 100)
dsize = (width, height)
# resize image
output = cv2.resize(src, dsize)
cv2.imwrite('D:/cv2-resize-image-50.png',output)
```

TP1

Resize Image



Example 2 : Resize image only horizontally

```
import cv2
```

```
src = cv2.imread('D:/cv2-resize-image-original.png', cv2.IMREAD_COLOR)
```

```
# set a new width in pixels
```

```
new_width = 300
```

```
# dsize
```

```
dsize = (new_width, src.shape[0])
```

```
# resize image
```

```
output = cv2.resize(src, dsize, interpolation = cv2.INTER_AREA)
```

```
cv2.imwrite('D:/cv2-resize-image-width.png',output)
```



TP1

Resize Image



Example 2 : Resize image only vertically

```
import cv2
```

```
src = cv2.imread('D:/cv2-resize-image-original.png', cv2.IMREAD
```

```
# set a new height in pixels
```

```
new_height = 200
```

```
# dsize
```

```
dsize = (src.shape[1], new_height)
```

```
# resize image
```

```
output = cv2.resize(src, dsize, interpolation = cv2.INTER_AREA)
```

```
cv2.imwrite('D:/cv2-resize-image-height.png',output)
```



TP1



Convert color image to black and white

```
import cv2
#read image
img_grey = cv2.imread('D:/original.png', cv2.IMREAD_GRAYSCALE)
# define a threshold, 128 is the middle of black and white in grey scale
thresh = 128
# threshold the image
img_binary = cv2.threshold(img_grey, thresh, 255, cv2.THRESH_BINARY)
#save image
cv2.imwrite('D:/black-and-white.png',img_binary)
```



Convert color image to grey scale image

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Lecture d'une sequence vidéo

```
import cv2
# connexion à la camera
cap = cv2.VideoCapture(0)
while (True):
    # lecture d'une image (frame) du flux vidéo
    ret, frame = cap.read()
    # affichage de l'image dans une fenetre nommée Vidéo
    cv2.imshow('Video', frame)
    #Arrêt flux caméra
    if cv2.waitKey(1) == 27:
        break
# fermeture de tous les fenêtres d'affichage
cap.release()cv2.destroyAllWindows()
```

Lecture d'une sequence vidéo

```
import cv2
cap = cv2.VideoCapture(0)
i=1
while (True):
    ret, frame = cap.read()
    cv2.imwrite('captureImg'+str(i)+'.jpg', frame)
    cv2.imshow('Video', frame)
    i+=1
    if cv2.waitKey(1) == 27:
        break
cap.release()
cv2.destroyAllWindows()
```