**TP4 : Prise en main de la vidéo avec OpenCV**

**Objectif :**

* Capturer une vidéo depuis la **webcam ou un fichier**
* Lire image par image (frame)
* Afficher en temps réel
* Appliquer des traitements simples (niveaux de gris, détection de mouvement, etc.)

**1. Lire une vidéo depuis la webcam**

import cv2

# Ouvrir la webcam (index 0 = caméra par défaut)

cap = cv2.VideoCapture(0)

if not cap.isOpened():

    print("Erreur : Impossible d'ouvrir la webcam")

    exit()

while True:

    ret, frame = cap.read()  # Lire une image (ret = True si lecture OK)

    if not ret:

        print("Erreur lors de la lecture de la frame")

        break

    # Afficher l’image (frame)

    cv2.imshow("Webcam", frame)

    # Quitter si on appuie sur 'q'

    if cv2.waitKey(1) == ord('q'):

        break

# Libérer les ressources

cap.release()

cv2.destroyAllWindows()

* cv2.VideoCapture(0) : démarre la webcam.
* ret, frame = cap.read() : lit une image de la vidéo (ou webcam).
* cv2.imshow(...) : affiche chaque frame.
* cv2.waitKey(1) : attend 1 ms. Si on appuie sur 'q', la boucle s'arrête.
* cap.release() : libère la caméra.
* destroyAllWindows() : ferme toutes les fenêtres.

**2. Lire une vidéo depuis un fichier .mp4**

import cv2

# Ouvrir la webcam (index 0 = caméra par défaut)

cap = cv2.VideoCapture(r"C:\Users\win11\Videos\VID-20250501-WA0016.mp4")

while cap.isOpened():

    ret, frame = cap.read()

    if not ret:

        break

    cv2.imshow("Vidéo", frame)

    if cv2.waitKey(25) == ord('q'):

        break

cap.release()

cv2.destroyAllWindows()

**3. Convertir la vidéo en niveaux de gris (traitement live)**

import cv2

# Ouvrir la webcam (index 0 = caméra par défaut)

cap = cv2.VideoCapture(r"C:\Users\win11\Videos\VID-20250501-WA0016.mp4")

while cap.isOpened():

    ret, frame = cap.read()

    if not ret:

        break

    # Traitement : conversion en niveaux de gris

    gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

    cv2.imshow("Vidéo", gray)

    if cv2.waitKey(25) == ord('q'):

        break

cap.release()

cv2.destroyAllWindows()

**4. Enregistrer la vidéo dans un fichier (ex. output.avi)**

import cv2

cap = cv2.VideoCapture(0)

# Définir le codec et créer l'objet VideoWriter

fourcc = cv2.VideoWriter\_fourcc(\*'XVID')

out = cv2.VideoWriter('output.avi', fourcc, 20.0, (640, 480))  # 20 FPS, taille 640x480

while True:

    ret, frame = cap.read()

    if not ret:

        break

    # Écriture dans le fichier

    out.write(frame)

    cv2.imshow('Webcam - Enregistrement', frame)

    if cv2.waitKey(1) == ord('q'):

        break

cap.release()

out.release()

cv2.destroyAllWindows()

**Remarques :**

* Le fichier .avi sera enregistré dans le dossier courant.
* cv2.VideoWriter\_fourcc(\*'XVID') : définit le codec pour compression vidéo.

**5. Détection de mouvement (soustraction de fond simple)**

import cv2

cap = cv2.VideoCapture(0)

ret, frame1 = cap.read()

ret, frame2 = cap.read()

while cap.isOpened():

    # Différence entre 2 images

    diff = cv2.absdiff(frame1, frame2)

    gray = cv2.cvtColor(diff, cv2.COLOR\_BGR2GRAY)

    blur = cv2.GaussianBlur(gray, (5,5), 0)

    \_, thresh = cv2.threshold(blur, 20, 255, cv2.THRESH\_BINARY)

    cv2.imshow("Détection de mouvement", thresh)

    frame1 = frame2

    ret, frame2 = cap.read()

    if not ret or cv2.waitKey(1) == ord('q'):

        break

cap.release()

cv2.destroyAllWindows()

**Résumé des fonctions utiles**

|  |  |
| --- | --- |
| **Fonction** | **Rôle** |
| cv2.VideoCapture() | Lire la webcam ou une vidéo |
| cap.read() | Lire une frame |
| cv2.imshow() | Afficher une image |
| cv2.waitKey() | Attendre un temps ou une touche |
| cv2.VideoWriter() | Enregistrer la vidéo |
| cv2.cvtColor(..., GRAY) | Convertir en niveaux de gris |
| cv2.absdiff() | Calculer la différence entre deux frames |