OpenCV 4 Cheat Sheet

HTTPS://PYMOTION.COM

Entrée / Sortie

Ouvrir une image :

- img = cv2.imread(chemin)

Afficher une image :

- cv2.imshow(fenetre, img)
- cv2.waitKey(t) // avec t en mS - si t=0 : en continu

Enregistrer une image :

- cv2.imwrite(chemin, img)

Lire une vidéo :

- cap = videoCapture(chemin)
- ret, img = cap.read() // Pour chaque image
- cap.release()

Enregistrer une vidéo :

- out = cv2.VideoWriter(chemin, codec, fps, (w, h))
- out.write(img) // Pour chaque image
- out.release()

Informations d'une image

Obtenir la taille d'une image:

- Taille = img.shape

Obtenir le nombre de pixels :

- Nbr = img.size

Obtenir la valeur d'un pixel :

- Valeur = img[x,y,canal]
- Valeur = img.item(x,y,canal)

Informations d'une vidéo

Obtenir les informations :

- Info = cap.get(propId)

Différentes informations (propId) :

Position en mS	cv2.CAP_PROP_POS_MSEC
Position en # image	cv2.CAP_PROP_POS_FRAMES
Position en ratio [0 -> 1]	cv2.PROP_POS_AVI_RATIO
Largeur de l'image (w)	cv2.CAP_PROP_FRAME_WIDTH
Hauteur de l'image (h)	cv2.CAP_PROP_FRAME_HEIGHT
FPS	cv2.CAP_PROP_FPS
Codec	cv2.CAP_PROP_FOURCC
Nombre total d'image	cv2.CAP_PROP_FRAME_COUNT

Manipulation d'image

Pivoter une image :

- M = cv2.getRotationMatrix2D(centre, angle, échelle)
- img_rot = cv2.warpAffine(img, M, (w, h))

Déplacer une image :

- M = np.float32([[1,0,tx],[0,1,ty]]) // tx/ty : déplacement en x/y
- img_trans = cv2.warpAffine(img, M, (w, h))

Découper une partie d'image :

- zone = img[y:y+h,x:x+w]

Redimensioner une image

- img = cv2.resize(img, None, fx, fy, interpolation)

Interpolations :

Proche voisin	cv2.INTER_NEAREST
Bilinéaire	cv2.INTER_LINEAR
4x4 voisinage	cv2.INTER_CUBIC

Appliquer un filtre moyenneur :

- median = cv2.Blur(img,(5,5))

Appliquer un filtre médian :

- median = cv2.medianBlur(img,5)

Appliquer un filtre Gaussien

- blur = cv2.GaussianBlur(img,(5,5),0)

Seuiller une image en niveaux de gris :

- ret, thresh = cv2.threshold(img,val_bas,val_haut,sseuil_baseuil)

Seuils :

Binaire	cv2.THRESH_BINARY
Tronqué	cv2.THRESH_TRUNC
A zéro	cv2.THRESH_TOZERO

Seuiller une image couleur :

- cv2.inrange(chemin, img)

Appliquer une opérations morphologiques:

- Erosion = cv2.erode(img,kernel,iterations = 1)
- Dilation = cv2.dilate(img,kernel,iterations = 1)
- Kernel = cv2.getStructuringElement(forme,(tx,ty))
- Ouv = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
- Ferm = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)

Formes d'éléments structurants :

Rectangulaire	cv2.MORPH_RECT
Croix	cv2.MORPH_CROSS
Elipse	cv2.MORPH_ELLIPSE

Dessin

Tracer un rectangle :

- img = cv2.rectangle(img, pt1, pt2, coul, épaisseur)

Tracer un cercle :

- img = cv2.circle(img, centre, rayon, coul, épaisseur)

Tracer une ligne :

- img = cv2.line(img, pt1, pt2, coul, épaisseur)

Ajouter du texte:

- img = cv2.putText(img, texte, orig, police, taille, coul, épaisseur)

Variables:

pt1,pt2	Positions (x,y) des points hautgauche et basdroit
centre	Position (x,y) du centre du cercle
org	Position en bas à gauche du texte: (x,y)
police	Police d'écriture. ex : cv2.FONT_HERSHEY_SIMPLEX
taille	Taille d'écriture
coul	Couleur de la forme : (R,G,B)
épaisseur	Épaisseur de la forme

Conversion colorimétrique

Changer d'espace colorimétrique:

- img = cv2.cvtColor(img, Espace)

Espace colorimétrique

BGR <-> Gris	cv2.COLOR_BGR2GRAY
	cv2.COLOR_GRAY2BGR
BGR <-> RGB	cv2.COLOR_BGR2RGB
	cv2.COLOR_RGB2BGR
BGR <-> HSV	cv2.COLOR_BGR2HSV
	cv2.COLOR_HSV2BGR
BGR <-> LAB	cv2.COLOR_BGR2LAB
	cv2.COLOR_LAB2BGR

Reconnaissance de forme

Détecter les contours:

- Cnts, hier = cv2.findContours(img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

Tracer le contours :

- cv2.drawContours(img, Cnts, #contours, coul, épaisseur) // si #contours=-1, trace tous les contours

Calculer le périmètre d'un contours :

- Perimetre = cv2.arcLength(cnt, closed= TRUE)