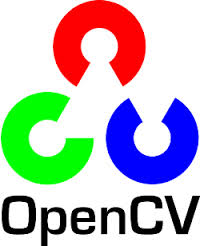
** OpenCV Python References**

**Prof. Dario Maio, Dott. Antonio Magnani**

* cv2.**imwrite**(filename, img[, params]) → retval

Salve un’immagine in un file specificato. Parametri:

* + filename: nome del file
  + img: immagine da salvare.
  + params: fare riferimento alla [documentazione](https://docs.opencv.org/2.4/modules/highgui/doc/reading_and_writing_images_and_video.html).
* cv2.**imshow**(winname, mat) → None

Mostra un’immagine di una finestra specificata. Parametri:

* + Winname: nome finestra.
  + Mat: immagine da mostrare.
* cv2.**waitKey**([delay]) → retval

Attende la pressione di un tasto. L’unico parametro opzionale è l’eventuale ritardo in millisecondi.

* cv2.**rectangle**(img, pt1, pt2, color[, thickness[, lineType[, shift]]]) → None

Funzione di disegno. Permette di disegnare un rettangolo secondo le specifiche definite dai parametri:

* + img: input image su cui applicare il disegno.
  + pt1: vertice del rettangolo.
  + pt2: vertice opposto del rettangolo.
  + color: colore del rettangolo (nella forma (R,G,B)).
  + thickness: parametro opzionale per definire lo spessore della linea. Un valore negativo implica il filling del rettangolo.
  + lineType: parametro opzionale per definire la tipologia di linea.
  + shift: parametro opzionale per definire uno scostamento rispetto alle coordinate originali.
* cv2.**putText**(img, text, org, fontFace, fontScale, color[, thickness[, lineType[, bottomLeftOrigin]]]) → None

Funzione di disegno. Permette l’aggiunta di una stringa di testo all’immagine. Parametri:

* + img: immagine
  + text: testo.
  + org: angolo inferiore-sinistro della stringa nell’immagine.
  + fontFace: tipo di font. Specificare uno tra FONT\_HERSHEY\_SIMPLEX, FONT\_HERSHEY\_PLAIN, FONT\_HERSHEY\_DUPLEX, FONT\_HERSHEY\_COMPLEX, FONT\_HERSHEY\_TRIPLEX, FONT\_HERSHEY\_COMPLEX\_SMALL, FONT\_HERSHEY\_SCRIPT\_SIMPLEX, o FONT\_HERSHEY\_SCRIPT\_COMPLEX
  + fontScale: fattore di scala che è moltiplicato per la dimensione base dello specifico font utilizzato.
  + color: colore del testo (nella forma (R,G,B)).
  + thickness: parametro opzionale per definire lo spessore delle linee utilizzate per disegnare testo.
  + lineType: parametro opzionale per definire la tipologia di linea.
  + bottomLeftOrigin: parametro opzionale. Quando è true l’origine è fissata nel corner inferiore sinistro. Altrimenti, nel corner superiore sinistro.
* cv2.**cvtColor**(src, code[, dst[, dstCn]]) → dst
  + Converte un’immagine da uno spazio colore ad un altro. Parametri:
  + src: input image
  + dst: output image della medesima dimensione di src
  + code: descrizione dello spazio colore. Nel caso di conversione RGB-grayscale (e viceversa) si possono utilizzare: CV\_BGR2GRAY, CV\_RGB2GRAY, CV\_GRAY2BGR, CV\_GRAY2RGB. Utilizzare la trasformazione più appropriata rispetto al formato di canale in ingresso.
  + dstCn: parametro opzionale che può essere utilizzato per definire il numero di canali nell’immagine destinazione. Se non specificato, viene derivato automaticamente da src e code.
* cv2.**GaussianBlur** (src, ksize, sigmaX[, dst[, sigmaY[, borderType]]]) → dst

Operazione di blurring (sfocatura) utilizzando un filtro gaussiano. Parametri:

* + src: input image
  + dst: output image della medesima dimensione di src
  + ksize: dimensione del kernel gaussiano. Altezza e larghezza del filtro possono differire ma devono essere entrambi positivi e dispari.
  + sigmaX: deviazione standard del kernel gaussiano rispetto ad X. In questa esercitazione utilizzare 0.
  + sigmaY: parametro opzionale. deviazione standard del kernel gaussiano rispetto ad Y. In questa esercitazione può non essere specificato.
  + borderType: parametro opzionale. In questa esercitazione può non essere specificato.
* cv2.**accumulateWeighted**(src, dst, alpha[, mask]) → None

Permette l’esecuzione di una media corrente tramite un’immagine di input ed un’immagine accumulatore. La destinazione diviene il nuovo accumulatore. Parametri:

* + src: input image.
  + dst: immagine accumulatore.
  + alpha: peso da applicare all’immagine di input per effettuare la media pesata.
  + mask: parametro opzionale (non considerato in questa esercitazione).
* cv2.**absdiff**(src1, src2[, dst]) → dst

Calcola il valore assoluto (element-based) tra due array. Parametri:

* + src1: primo array.
  + src2: secondo array.
  + dst: array di output che ha la stessa dimensione e tipo degli array di input.
* cv2.**threshold**(src, thresh, maxval, type[, dst]) → retval, dst

Applica una soglia fissa ad ogni elemento dell’array. Questa funzione restituisce due variabili, retval rappresenta un valore di check. Parametri:

* + src: array di input.
  + dst: output array della medesima dimensione di src.
  + thresh: valore di soglia.
  + maxval: valore di soglia massimo
  + type: tipo di sogliatura. Tra i più comuni: THRESH\_BINARY, THRESH\_BINARY\_INV. Fare riferimento alla [documentazione ufficiale](https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html).
* cv2.**dilate**(src, kernel[, dst[, anchor[, iterations[, borderType[, borderValue]]]]]) → dst

Applica un’operazione di dilatazione all’immagine in input. Parametri:

* + src: input image.
  + dst: output image della medesima dimensione di src.
  + kernel: definisce uno specifico kernel da applicare all’operazione di dilating. In questa esercitazione non è stato usato uno specifico kernel (utilizzare None).
  + iterations: numero di volte in cui il dilating viene applicato. Maggiore è il numero, maggiore è la dilatazione. Testare quale sia il valore ideale rispetto al contesto ripreso dalla telecamera, oltre a quello specificato nel codice.
  + borderType, borderValue e anchor: sono parametri opzionali ignorati in questa esercitazione. Definisco il comportamento del dilating rispetto ai bordi dell’immagine e la posizione dell’anchor rispetto all’elemento. Non specificando nulla utilizziamo il centro immagine come posizione.
* cv2.**findContours**(image, mode, method[, contours[, hierarchy[, offset]]]) → contours, hierarchy

Trova i contorni in un’immagine binaria. Parametri:

* + image: input image
  + countours: contorni rilevati. Ogni contorno è definito come un vettore di punti.
  + hierarchy: vettore di output opzionale contenente informazioni riguardanti la topologia dell’immagine. In questa esercitazione è stato ignorato.
  + mode: definisce la modalità di recupero dei contorni. In questa esercitazione è consigliato cv2.RETR\_EXTERNAL (fare riferimento alla [documentazione](https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?)).
  + method: definisce la modalità di approssimazione dei contorni. In questa esercitazione è consigliato cv2.CHAIN\_APPROX\_SIMPLE (fare riferimento alla [documentazione](https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?)).
* cv2.**contourArea**(contour[, oriented]) → retval

Determina l’area di un contorno. Parametri:

* + contour: vettore di punti 2D rappresentanti i vertici del contorno.
  + oriented: parametro opzionale che permette di determinare l’orientazione di un contorno. In questa esercitazione non è stato considerato.
* cv2.**boundingRect**(points) → retval

Calcola il rettangolo minimo che delimita un insieme di punti, fornendo il punto estremo superiore destro, larghezza e altezza del rettangolo (i.e., una tupla nella forma x,y,w,h). Parametri:

* + points: insieme di punti 2D.