

## OpenCV References

Prof. Dario Maio, Dott. Francesco Turrone



### Strutture Base:

Rect : rappresenta un generico rettangolo 2D con i seguenti parametri: top-left corner, larghezza e altezza del rettangolo. É possibile specificare il tipo T del Rettangolo (interi, float, etc.) con Rect\_<T>.

Mat : rappresenta una generica matrice il cui tipo viene definito in fase di allocazione.

vector<T> : rappresenta un vettore i cui elementi hanno tipo T. Per ottenere il numero di elementi contenuti nel vettore usare il metodo size(). Per ottenere l'i-esimo elemento del vettore procedere con l'usuale modo di accesso agli array ([i]).

IplImage : struttura che rappresenta una generica immagine. É possibile definire dimensioni, profondità, canali e molti altri parametri. Consultare la documentazione ufficiale OpenCV per maggiori dettagli. Ricordiamo qui come creare una generica immagine di BYTE (unsigned 8bit integer):

```
IplImage img = cvCreateImage(cvSize(w,h), IPL_DEPTH_8U, 1);      // ad 1 canale
```

```
IplImage img = cvCreateImage(cvSize(w,h), IPL_DEPTH_8U, 3);      // a 3 canali
```

### Metodi:

IplImage cvCreateImage(cvSize size, int depth, int channels) : crea un'immagine con le dimensioni profondità e numero di canali (da 1 a 4) specificati. Le profondità possono essere: IPL\_DEPTH\_8U - uint 8-bit integers IPL\_DEPTH\_8S - signed 8-bit integers IPL\_DEPTH\_16U - uint 16-bit integers IPL\_DEPTH\_16S - signed 16-bit integers IPL\_DEPTH\_32S - signed 32-bit integers IPL\_DEPTH\_32F - single precision floating-point numbers IPL\_DEPTH\_64F - double precision floating-point numbers.

void rectangle(Mat& img, Rect rec, const Scalar& color, int thickness=1) : funzione di disegno di un semplice rettangolo. I parametri sono:

img – immagine.

rec – rettangolo da disegnare.

color – colore del rettangolo o tonalità (grayscale). Bianco = CV\_RGB(255, 255, 255)

thickness – Spessore della linea.

void cvResize(const CvArr\* src, CvArr\* dst, int interpolation=CV\_INTER\_LINEAR):

questa funzione ridimensiona una immagine *src* in modo che sia contenuta esattamente in *dst*. Viene eseguita l'interpolazione specificata in input che può essere: CV\_INTER\_NN (nearest-neighbor), CV\_INTER\_LINEAR (bilineare), CV\_INTER\_AREA (ricampiona utilizzando la relazione pixel area), CV\_INTER\_CUBIC (bicubica). *Il tipo CvArr può essere ad esempio una IplImage.*

void cvMerge(const CvArr\* src0, const CvArr\* src1, const CvArr\* src2, const CvArr\*

src3, CvArr\* dst): funzione che compone un'array multicanale a partire da diversi array a canale singolo. *src0*, *src1*, *src2* e *src3* sono le immagini che si vogliono fondere. Se ad esempio voglio fondere tre immagini, allora *src0*, *src1* e *src2* saranno diversi da NULL, mentre *src3* sarà NULL. L'immagine destinazione avrà tanti canali quante sono le immagini che voglio fondere. *Il tipo CvArr può essere ad esempio una IplImage.*

void cvCvtColor(const CvArr\* src, CvArr\* dst, int code) : funzione che converte un'immagine da uno spazio di colore ad un altro. I parametri sono:

*src* – Immagine sorgente ad 8-bit (8u), 16-bit (16u) or single-precision floating-point (32f).

*dst* – Immagine destinazione dello stesso tipo dell'immagine sorgente. Il numero dei canali può essere differente.

*code* – operazione di conversione colore specificata dal codice CV\_\*src\_color\_space\* 2 \*dst\_color\_space\*. Ad esempio:

CV\_YCrCb2RGB = conversione da YCrCb ad RGB

Mat cvarrToMat(const CvArr\* src) : converte una CvMat o IplImage o CvMatND nel tipo Mat.

void cvReleaseImage(const CvArr\* src) : rilascia le risorse utilizzate.

## Visualizzatore per immagini:

cvNamedWindow("camcvWin", CV\_WINDOW\_AUTOSIZE) : creazione.

imshow("camcvWin", gray) : visualizzazione di un'immagine

## Detection:

Classe: CascadeClassifier

Metodi:

bool load(const string& filename)

Carica un classificatore dal file specificato in input.

void detectMultiScale(const Mat& image, vector<Rect>& objects, double scaleFactor=1.1, int minNeighbors=3, int flags=0, Size minSize=Size())

Rileva oggetti di differenti dimensioni all'interno dell'immagine in input. Gli oggetti rilevati vengono restituiti all'interno di un vettore di rettangoli (Rect). Nella definizione sono specificati anche i valori di default. I parametri sono:

image – matrice di tipo CV\_8U contenente un'immagine nella quale trovare gli oggetti.

objects – vettore di rettangoli dove ogni rettangolo rappresenta un oggetto localizzato.

scaleFactor – parametro che specifica quanto l'immagine deve essere ridotta ad ogni scala. Un buon valore per questo parametro è 1.1.

minNeighbors – parametro che specifica quanti vicini deve avere ogni candidato rettangolo affinché esso sia mantenuto.

flags – parametro di detection. Si consiglia di usare la costante CV\_HAAR\_SCALE\_IMAGE perchè viene scalata l'immagine piuttosto che il detector. La face detection è ottimizzata con questo flag impostato anzichè il valore di default 0.

minSize – minima dimensione possibile di ogni oggetto. Oggetti più piccoli vengono ignorati. Un valore consigliato per questo parametro è Size(80, 80) oppure Size(30,30).