

Taller 2: Transformaciones y Preproceso

Técnicas de Inteligencia Artificial: Visión de Máquina

Prof. Flavio Prieto
email: faprieto@unal.edu.co

Ingeniería Mecatrónica
Facultad de Ingeniería
Universidad Nacional de Colombia Sede Bogotá



29 de febrero de 2023

Redimensionamiento.¹

Esta función permite cambiar el tamaño de una imagen:

`cv.Resize(im-fuente, (width, height), Método de interpolación);`

Metodos de Interpolación:

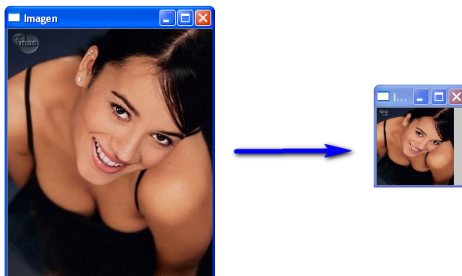
- ▶ Vecino más cercano (`cv2.INTER_NN`)
- ▶ Bilineal (`cv2.INTER_LINEAR`)
- ▶ Píxel área de re-muestreo (`cv2.INTER_AREA`)
- ▶ Bicúbica (`cv2.INTER_CUBIC`)

¹http://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html#resize

Transformación de Imágenes.

Ejemplo:

```
img = cv2.imread('fig/Notebook/Escudo_UN.png',0)  
height, width = img.shape[:2]  
img2 = cv2.resize(img, (2*width, height/2), interpolation =  
cv2.INTER_CUBIC)
```



Transformación de Imágenes.

Conversión espacios de color.

Estas función permite la conversión entre diferentes espacios de color.

```
cv2.cvtColor(img_Fuente, cv2.COLOR_BGR2RGB)
```

- ▶ Grises (8,16 bits)
- ▶ RGB
- ▶ XYZ
- ▶ HSV
- ▶ HLS
- ▶ LUV
- ▶ YCrCb

Transformación de Imágenes.

Conversion code	Meaning
CV_BGR2RGB CV_RGB2BGR CV_RGBA2BGRA CV_BGRA2RGBA	Convert between RGB and BGR color spaces (with or without alpha channel)
CV_RGB2RGBA CV_BGR2BGRA	Add alpha channel to RGB or BGR image
CV_RGBA2RGB CV_BGRA2BGR	Remove alpha channel from RGB or BGR image
CV_RGB2BGRA CV_RGBA2BGR CV_BGRA2RGB CV_BGR2RGBA	Convert RGB to BGR color spaces while adding or removing alpha channel
CV_RGB2GRAY CV_BGR2GRAY	Convert RGB or BGR color spaces to grayscale
CV_GRAY2RGB CV_GRAY2BGR CV_RGBA2GRAY CV_BGRA2GRAY	Convert grayscale to RGB or BGR color spaces (optionally removing alpha channel in the process)
CV_GRAY2RGBA CV_GRAY2BGRA	Convert grayscale to RGB or BGR color spaces and add alpha channel
CV_RGB2BGR565 CV_BGR2BGR565 CV_BGR5652RGB CV_BGR5652BGR CV_RGBA2BGR565 CV_BGRA2BGR565 CV_BGR5652RGBA CV_BGR5652BGRA	Convert from RGB or BGR color space to BGR565 color representation with optional addition or removal of alpha channel (16-bit images)
CV_GRAY2BGR565 CV_BGR5652GRAY	Convert grayscale to BGR565 color representation or vice versa (16-bit images)

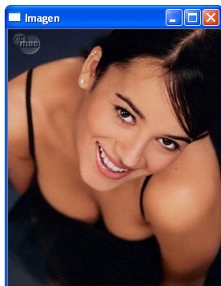
Conversion code	Meaning
CV_RGB2BGR555 CV_BGR2BGR555 CV_BGR5552RGB CV_BGR5552BGR CV_RGBA2BGR555 CV_BGRA2BGR555 CV_BGR5552RGBA CV_BGR5552BGRA	Convert from RGB or BGR color space to BGR555 color representation with optional addition or removal of alpha channel (16-bit images)
CV_GRAY2BGR555 CV_BGR5552GRAY	Convert grayscale to BGR555 color representation or vice versa (16-bit images)
CV_RGB2XYZ CV_BGR2XYZ CV_XYZ2RGB CV_XYZ2BGR	Convert RGB or BGR image to CIE XYZ representation or vice versa (Rec 709 with D65 white point)
CV_RGB2YCrCb CV_BGR2YCrCb CV_YCrCb2RGB CV_YCrCb2BGR	Convert RGB or BGR image to luma-chroma (aka YCC) color representation
CV_RGB2HSV CV_BGR2HSV CV_HSV2RGB CV_HSV2BGR	Convert RGB or BGR image to HSV (hue saturation value) color representation or vice versa
CV_RGB2HLS CV_BGR2HLS CV_HLS2RGB CV_HLS2BGR	Convert RGB or BGR image to HLS (hue lightness saturation) color representation or vice versa
CV_RGB2Lab CV_BGR2Lab CV_Lab2RGB CV_Lab2BGR	Convert RGB or BGR image to CIE Lab color representation or vice versa
CV_RGB2Luv CV_BGR2Luv CV_Luv2RGB CV_Luv2BGR	Convert RGB or BGR image to CIE Luv color representation
CV_BayerBG2RGB CV_BayerGB2RGB CV_BayerRG2RGB CV_BayerGR2RGB CV_BayerBG2BGR CV_BayerGB2BGR CV_BayerRG2BGR CV_BayerGR2BGR	Convert from Bayer pattern (single-channel) to RGB or BGR image

Transformación de Imágenes.

Ejemplo:

RGB a escala de grises.

```
img = cv2.imread('fig/Notebook/Escudo_UN.png',1) # Carga  
imagen en color  
img1 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
img2 = cv2.cvtColor(img1, cv2.COLOR_RGB2GRAY)
```



Transformación de Imágenes.

Umbralización.

Esta función permite umbralizar una imagen respecto a un valor determinado (umbral), en escala de grises.

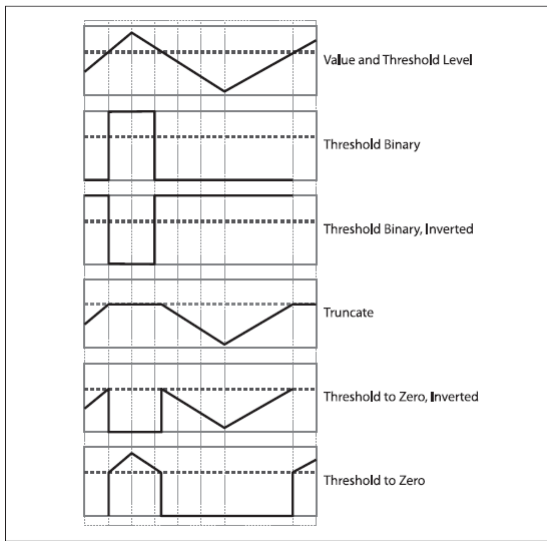
```
cv2.threshold(im-fuente, umbral,max-valor,Tipo-umbralización))
```

Tipos de umbralización:

1. cv2.THRESH_BINARY
2. cv2.THRESH_BINARY_INV
3. cv2.THRESH_TRUNC
4. cv2.THRESH_TOZERO_INV
5. cv2.THRESH_TOZERO

Transformación de Imágenes.

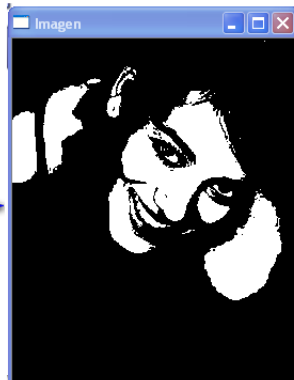
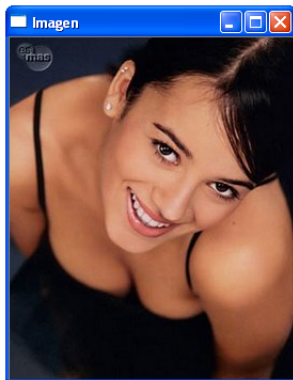
Tipos de Umbralización



Transformación de Imágenes.

Ejemplo:

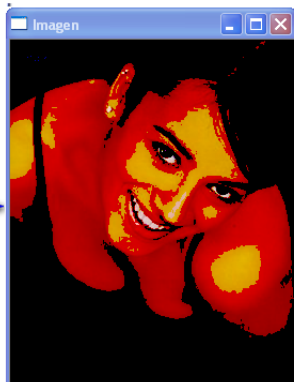
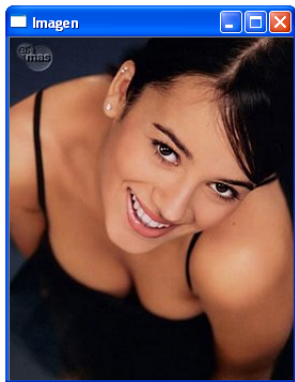
Umbralización Binaria



Transformación de Imágenes.

Ejemplo:

Umbralización Imagen color- Umbral a cero



Transformación de Imágenes.

Umbral Adaptativo.

cv2.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C[, dst])

Métodos ADP en OpenCV.

Existen dos métodos implementados en OpenCV para el cálculo del umbral de forma automática.

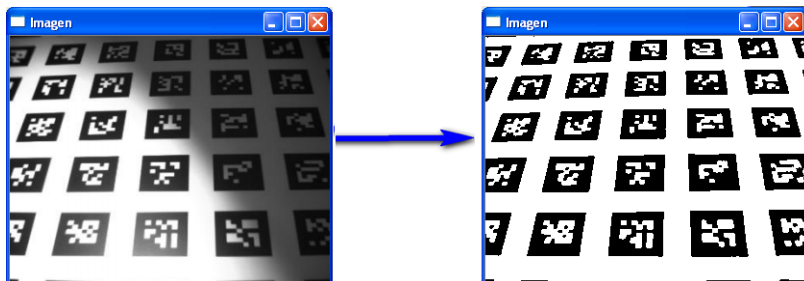
- ▶ Mediana: cv2.ADAPTIVE_THRESH_MEAN_C
- ▶ Gausiano: cv2.ADAPTIVE_THRESH_GAUSSIAN_C

blockSize.

Tamaño del vecindario de píxeles sobre los cuales se calcula el valor del umbral, puede ser 3, 5, 7, etc.

Transformación de Imágenes.

Ejemplo:



Sitio de interés:

`http://docs.opencv.org/2.4/modules/imgproc/
doc/miscellaneous_transformations.html?
highlight=threshold`

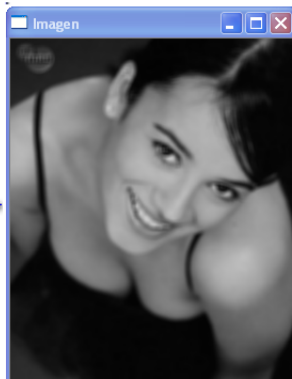
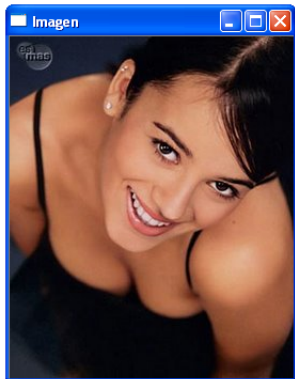
Filtros Lineales.

Este grupo de funciones nos permiten realizar operaciones como la convolución:

`cv2.filter2D(im-Origen, ddepth, Máscara (Kernel))`

- ▶ **ddepth:** cuando se fija en '-1' la imagen de salida tiene la misma profundidad (tipo de datos), que la imagen de entrada.
- ▶ Definición de la Máscara:
 - ▶ `kernel1 = np.ones((5, 5)) / 25`
 - ▶ `kernel2 = np.array([[1, 1, 1], [0, 0, 0], [-1, -1, -1]])`

Ejemplo:



Filtros de suavizado.

El suavizado permite reducción de ruido en la imagen.

1. **cv2.blur**(img, Skernel)

- ▶ El tamaño del Kernel (SKernel) es de la forma: '(3,3) o (5,5), ...'

2. **cv2.medianBlur**(img, Skernel)

- ▶ El tamaño del kernel es de la forma: 3 o 5 o 7,...)

3. **cv2.GaussianBlur**(img, Skernel, Desv. est. X, Desv. est. Y)

- ▶ El tamaño del Kernel (SKernel) es de la forma: '(3,3) o (5,5), ...'
- ▶ Si Desv. est. Y = 0, se hace igual a Desv. est. X.
- ▶ Si las dos desviaciones son cero, ellas se calculan del kernel.

4. **cv2.bilateralFilter**(img, d, sigmaColor, sigmaSpace)

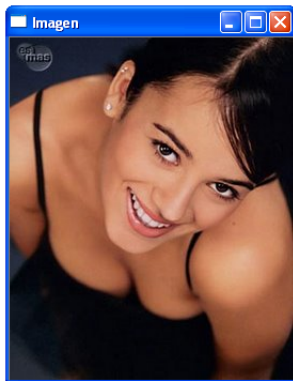
- ▶ **d**: Diámetro del vecindario al píxel que es usado durante el proceso de filtrado.
- ▶ **sigmaColor**: Desviación estándar del filtro en el espacio de color.
- ▶ **sigmaSpace**: Desviación estándar del filtro en el espacio de coordenadas.

Los siguientes filtros resaltan el contorno de las imágenes.

- ▶ Sobel.
- ▶ Laplaciano.
- ▶ Canny.

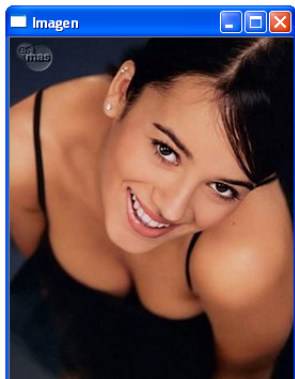
Sobel.

`cv2.Sobel(im-Fuente, tipo dato, Orden derivada X, Orden derivada Y, ksize=3);`



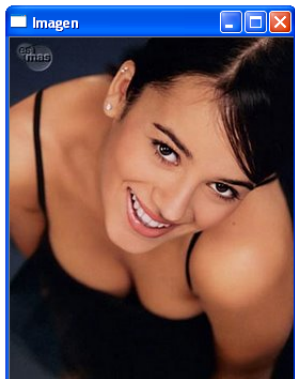
Laplaciano

`cv2.Laplacian(im-Fuente, Tipo de dato);`



Canny.

`cv2.Canny(im-Fuente, UmbralBajo, UmbralAlto);`



- ▶ Implementar, como una función de OpenCV, el filtro de Prewitt de 8 direcciones.
- ▶ Estudiar, evaluar y discutir las funciones vistas en el taller, cuando son aplicadas a imágenes reales.
Preferiblemente use imágenes del proyecto que está desarrollando para el curso, en caso de no tener aun estas imágenes use las imágenes del taller anterior.