

Tratamiento de Imagen Digital

Portafolio N°3

ALEJANDRO ALBERT GRAMAJE

ÍNDICE

1. Procesamiento de mejora de una imagen	3
Matricula 1:.....	3
Matricula 2:.....	4
Matricula 3.....	5
Matricula 4:.....	6
2. Superposición de imagen sobre un vídeo	7
Variante	8

1. Procesamiento de mejora de una imagen

Se carga una imagen de una matrícula y se pasa a escala de grises para trabajar solo con intensidades. Luego se redimensiona a un tamaño fijo para que todas tengan las mismas dimensiones. Después se aplica umbralización binaria para separar los caracteres del fondo: una automática con Otsu y otra manual, aproximando. Así se puede comparar cuál separa mejor los números del fondo.

Matricula 1:

Código:

```
import cv2

img = cv2.imread('Tema 4.4/Ejercicio_1/img/matricula.jpg')
imgGris = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
res = cv2.resize(imgGris, dsize=(54, 140), interpolation=cv2.INTER_CUBIC)

(thresh, im_bw) = cv2.threshold(imgGris, 128, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
thresh = 165
im_bw1 = cv2.threshold(imgGris, thresh, 255, cv2.THRESH_BINARY)[1]

cv2.imshow("imagenOriginal",img)
cv2.imshow("imagenGris", imgGris)
# cv2.imwrite('imagenAutomatica.jpg', im_bw)
cv2.imshow("imagenAutomatica", im_bw)
cv2.imshow("ImagenThresh", im_bw1)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Resultado:



Matricula 2:

Código:

```
import cv2

img = cv2.imread('Tema 4.4/Ejercicio_1/img/matricula2.jpg')
imgGris = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
res = cv2.resize(imgGris, dsize=(54, 140), interpolation=cv2.INTER_CUBIC)

(thresh, im_bw) = cv2.threshold(imgGris, 128, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
thresh = 180
im_bw1 = cv2.threshold(imgGris, thresh, 255, cv2.THRESH_BINARY)[1]

cv2.imshow("imagenOriginal",img)
cv2.imshow("imagenGris", imgGris)
# cv2.imwrite('imagenAutomatica.jpg', im_bw)
cv2.imshow("imagenAutomatica", im_bw)
cv2.imshow("ImagenThresh", im_bw1)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Resultado:



Matricula 3

Código:

```
import cv2

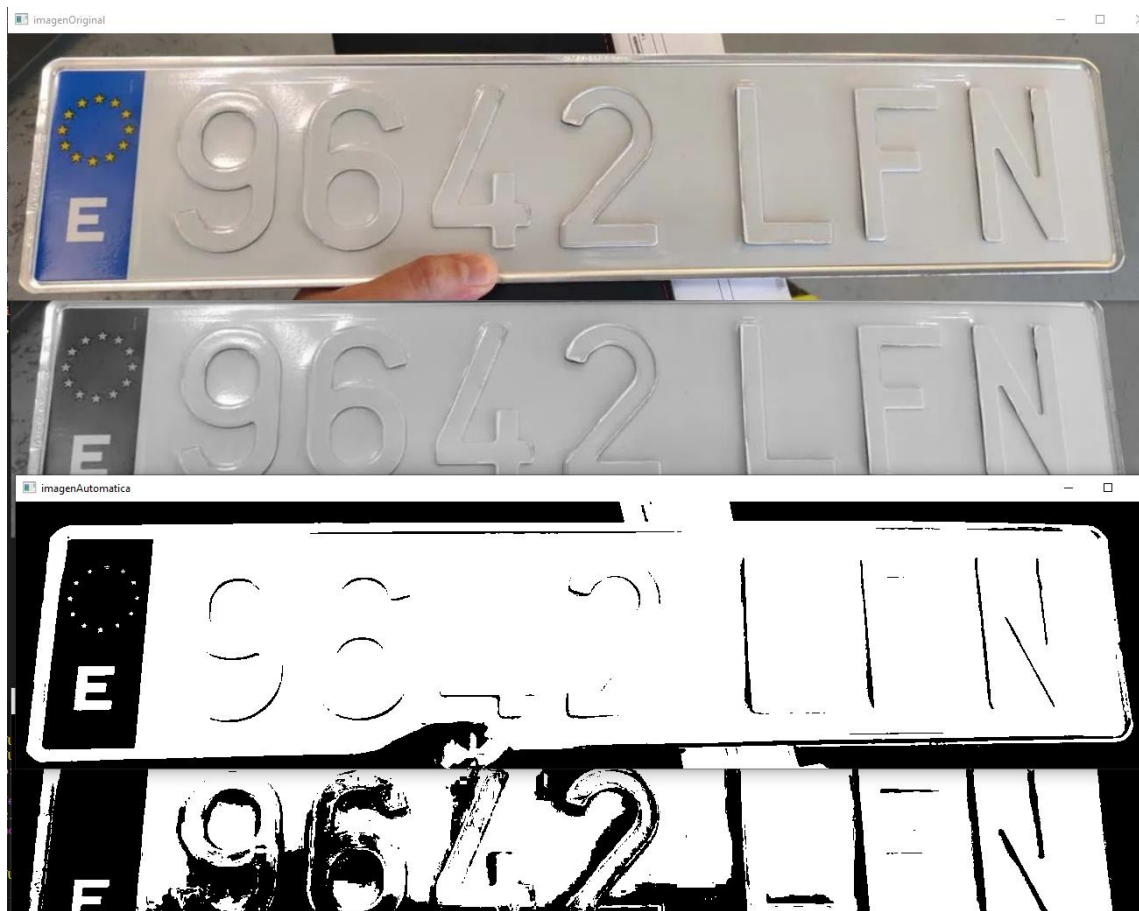
img = cv2.imread('Ejercicio_1/img/matricula3.jpg')
imgGris = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
res = cv2.resize(imgGris, dsize=(54, 140), interpolation=cv2.INTER_CUBIC)

(thresh, im_bw) = cv2.threshold(imgGris, 128, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
thresh = 176
im_bw1 = cv2.threshold(imgGris, thresh, 255, cv2.THRESH_BINARY)[1]

cv2.imshow("imagenOriginal",img)
cv2.imshow("imagenGris", imgGris)
# cv2.imwrite('imagenAutomatica.jpg', im_bw)
cv2.imshow("imagenAutomatica", im_bw)
cv2.imshow("ImagenThresh", im_bw1)

cv2.waitKey(0)
cv2.destroyAllWindows() |
```

Resultado:



Matricula 4:

Código:

```
import cv2

img = cv2.imread('Tema 4.4/Ejercicio_1/img/matricula4.jpg')
imgGris = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
res = cv2.resize(imgGris, dsize=(54, 140), interpolation=cv2.INTER_CUBIC)

(thresh, im_bw) = cv2.threshold(imgGris, 128, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
thresh = 57
im_bw1 = cv2.threshold(imgGris, thresh, 255, cv2.THRESH_BINARY)[1]

cv2.imshow("imagenOriginal",img)
cv2.imshow("imagenGris", imgGris)
# cv2.imwrite('imagenAutomatica.jpg', im_bw)
cv2.imshow("imagenAutomatica", im_bw)
cv2.imshow("ImagenThresh", im_bw1)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Resultado:



2. Superposición de imagen sobre un vídeo

Se carga un vídeo y se muestra frame a frame. En cada uno se coloca una imagen, en este caso un soldado imperial y un dinosaurio, pegada en la esquina. Esto se hace copiando los píxeles del logo directamente sobre el vídeo.

Código:

```
import numpy as np
import cv2 as cv

cap = cv.VideoCapture('Tema 4.4/Ejercicio_2/Video/videoKiddKeo.mp4')
imga = cv.cvtColor(cv.imread("Tema 4.4/Ejercicio_2/img/SoldadoImperio.jpg"), cv.COLOR_BGR2HSV)
imga=cv.resize(imga,(300,300))
img2=imga[0:500:2,0:1600:2,:]

while(cap.isOpened()):
    ret, frame = cap.read()
    rows,cols,channels = img2.shape
    roi = frame[0:rows, 0:cols]
    lowerBlue = np.array([85,40,50])
    upperBlue = np.array([135,255,255])
    mask = cv.inRange(img2, lowerBlue, upperBlue)
    mask_inv = cv.bitwise_not(mask)
    frame_bg = cv.bitwise_and(roi,roi,mask = mask)
    img2_fg = cv.cvtColor(cv.bitwise_and(img2,img2,mask = mask_inv),cv.COLOR_HSV2BGR)
    dst = cv.add(frame_bg,img2_fg)
    frame[0:rows, 0:cols ] = dst
    cv.imshow("img",frame)
    if cv.waitKey(30) & 0xFF == ord("q"):
        break

cap.release()
cv.destroyAllWindows()
```

Resultado:



Variante

Código:

```
import numpy as np
import cv2 as cv

cap = cv.VideoCapture('Ejercicio_2/Video/videoKiddKeo.mp4')
imga = cv.cvtColor(cv.imread("Ejercicio_2/img/Dinosaurio.jpg"), cv.COLOR_BGR2HSV)
Imga=cv.resize(imga,(150,150))
img2=Imga[0:500:2,0:1600:2,:]

while(cap.isOpened()):
    ret, frame = cap.read()
    rows,cols,channels = img2.shape
    roi = frame[0:rows, 0:cols]
    lowerBlue = np.array([35,40,50])
    upperBlue = np.array([90,255,255])
    mask = cv.inRange(img2, lowerBlue, upperBlue)
    mask_inv = cv.bitwise_not(mask)
    frame_bg = cv.bitwise_and(roi,roi,mask = mask)
    img2_fg = cv.cvtColor(cv.bitwise_and(img2,img2,mask = mask_inv),cv.COLOR_HSV2BGR)
    dst = cv.add(frame_bg,img2_fg)
    frame[0:rows, 0:cols ] = dst
    cv.imshow("img",frame)
    if cv.waitKey(30) & 0xFF == ord("q"):
        break

cap.release()
cv.destroyAllWindows()
```

Resultado:

