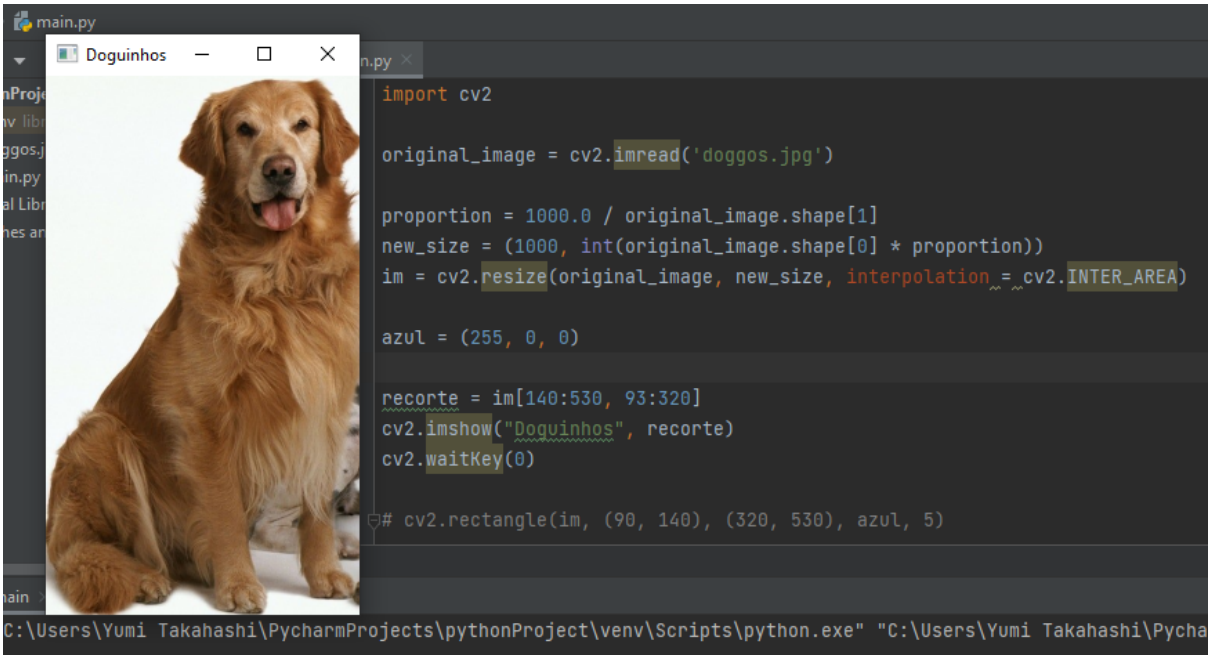
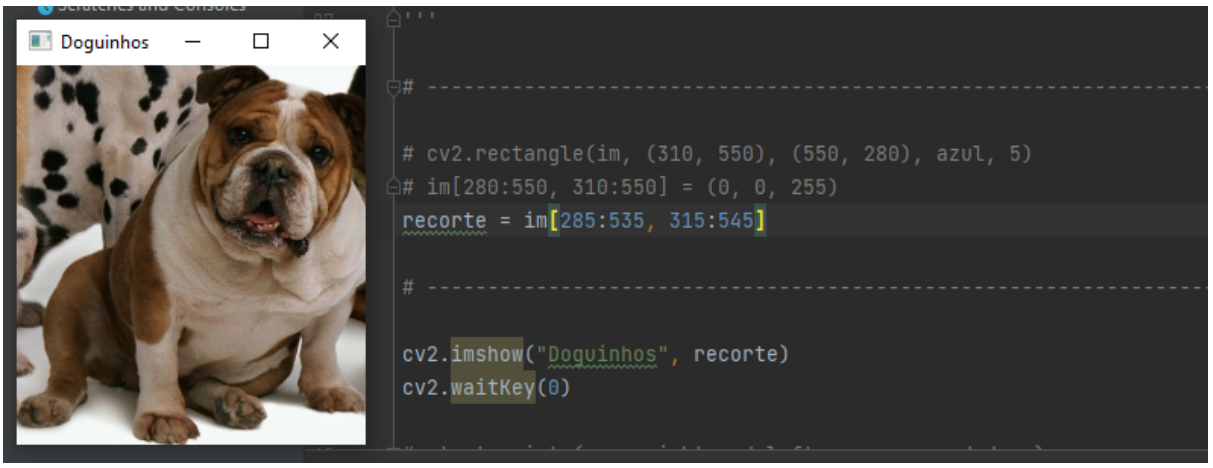
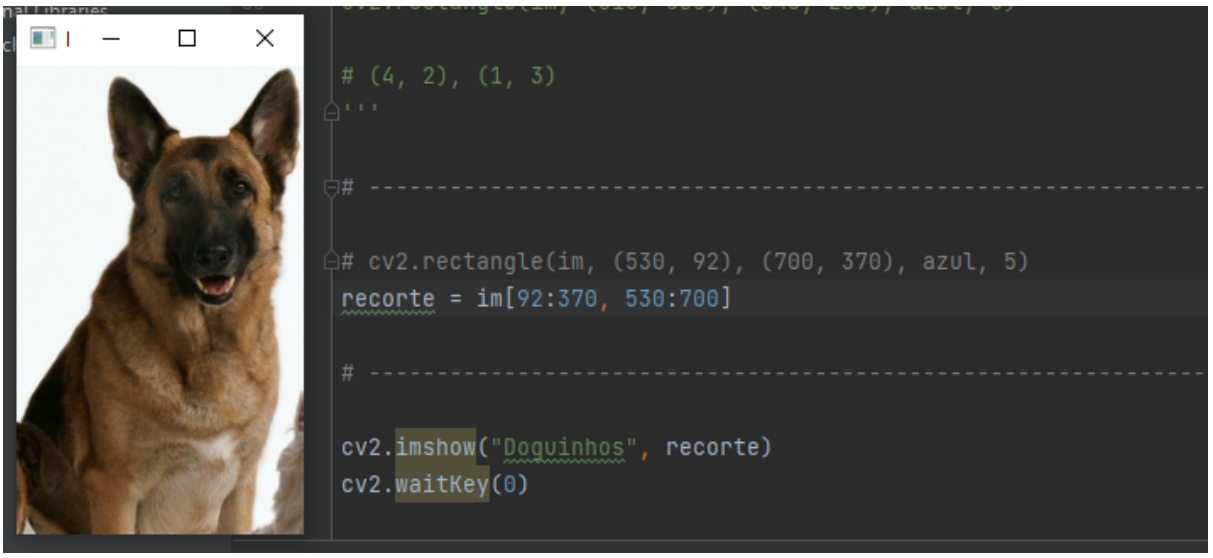
Roberta Yumi Romero Takahashi **1903220**

Escolher uma imagem com vários cães, mais que 5.

1 - Localizar cada cão e gerar imagens menores, individuais de cada cão e apresentar cada uma delas.

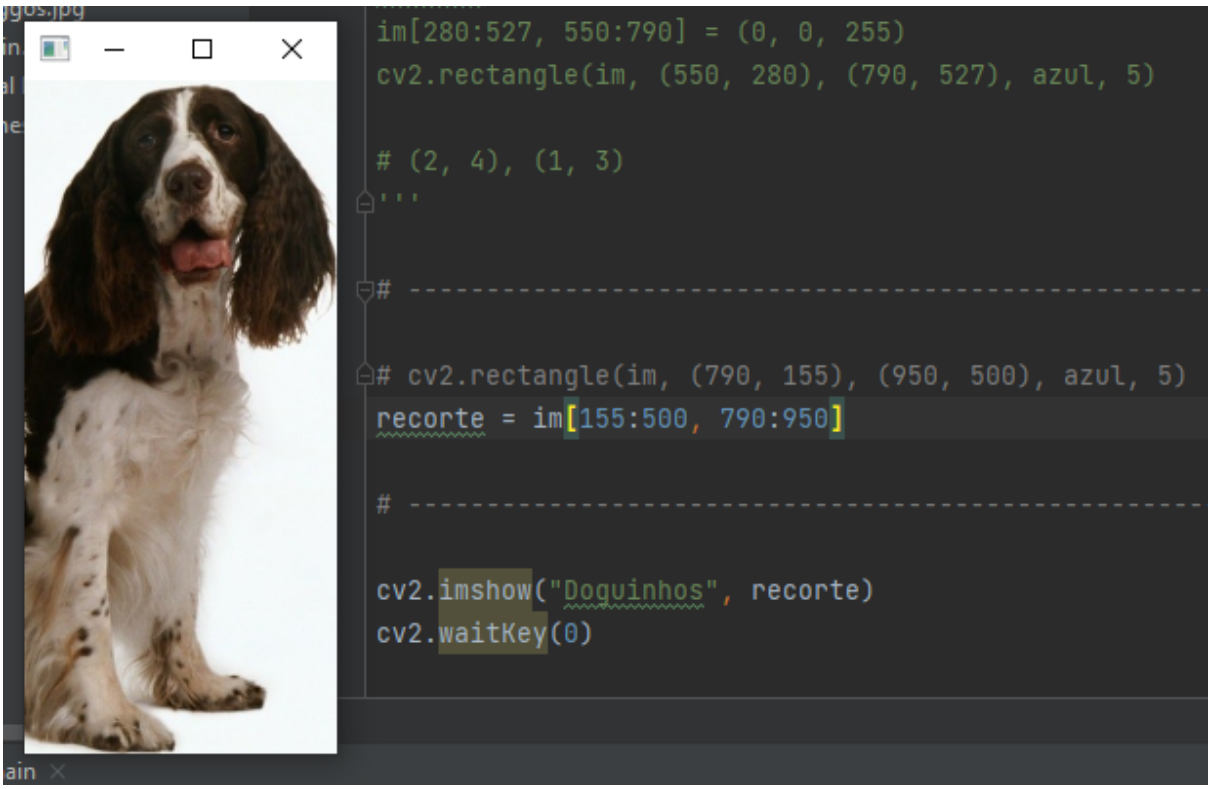






```
# (4, 2), (1, 3)
'''

# ------------------------------------------------------------

# cv2.rectangle(im, (530, 92), (700, 370), azul, 5)
recorte = im[92:370, 530:700]

# ------------------------------------------------------------

cv2.imshow("Doguinhos", recorte)
cv2.waitKey(0)
```

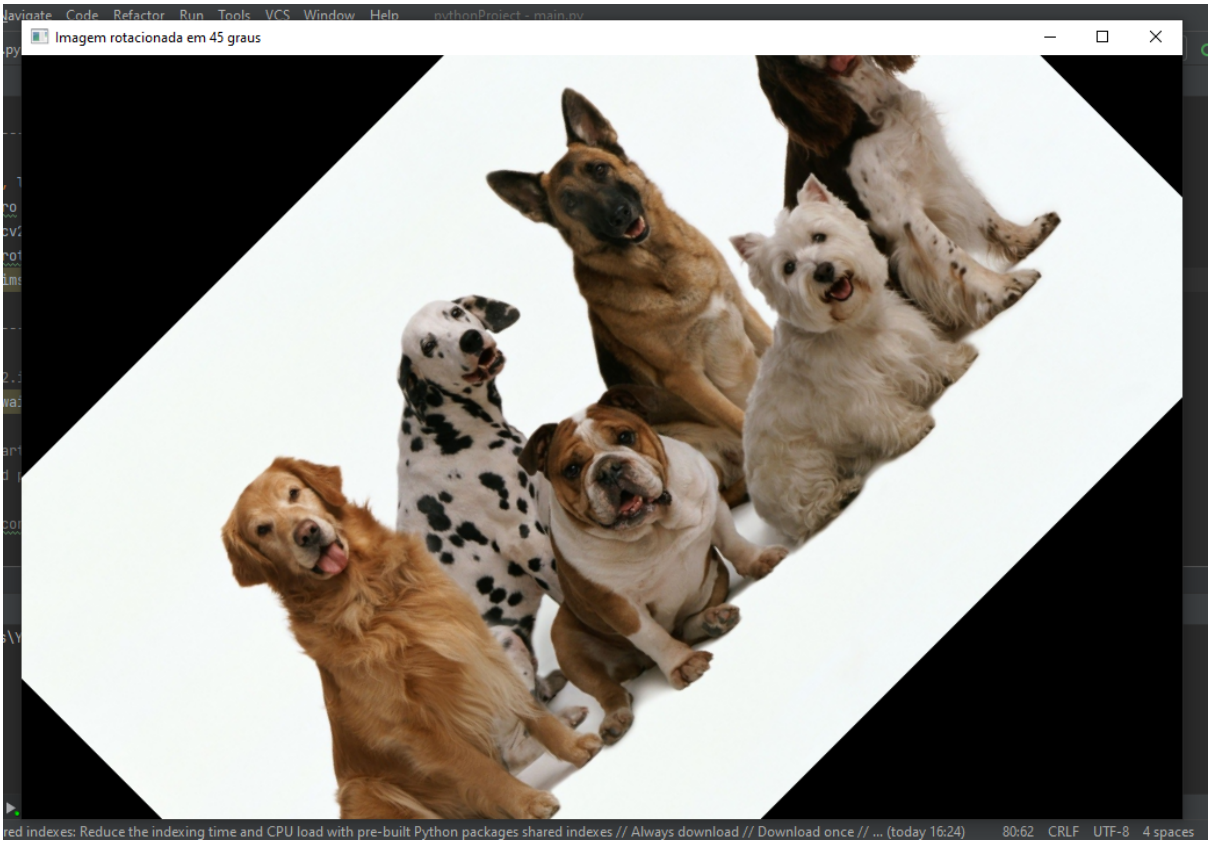
```
# ------------------------------------------------------------

# cv2.rectangle(im, (550, 280), (790, 527), azul, 5)
recorte = im[280:527, 550:790]

# ------------------------------------------------------------

cv2.imshow("Doguinhos", recorte)
cv2.waitKey(0)

# start point (goes right and left, goes up and down)
# end point(goes right and left, goes up and down)
```



2 - Na mesma imagem (original), localizar todos os cães e remover o resto da imagem, colocando o fundo na cor preta, somente os cães devem ficar na imagem.

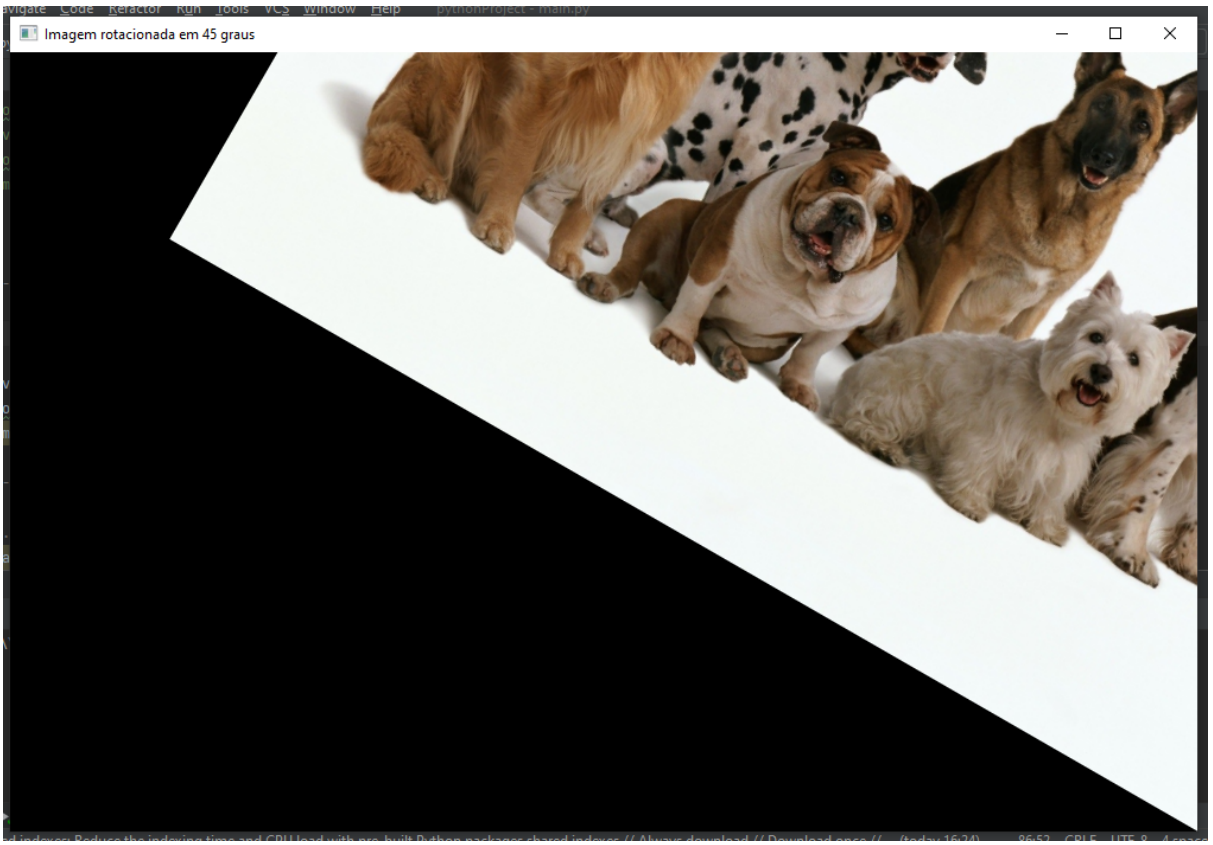
```python
mask = np.zeros(im.shape[:2], dtype="uint8")
cv2.rectangle(mask, (90, 92), (950, 535), azul, -1)
masked = cv2.bitwise_and(im, im, mask=mask)
cv2.imshow("Mask Applied to Image", masked)
```

3 - Rotacionar a imagem com relação ao centro da imagem em 45 graus.



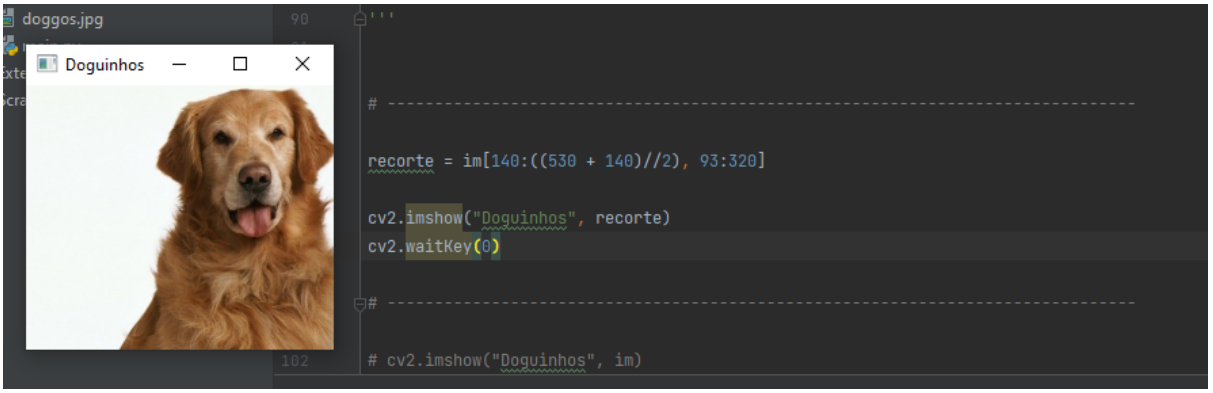

```python
(alt, lar) = im.shape[:2] #captura altura e largura
centro = (lar // 2, alt // 2) #acha o centro
M = cv2.getRotationMatrix2D(centro, 45, 1.0) #30 graus
img_rotacionada = cv2.warpAffine(im, M, (lar, alt))
cv2.imshow("Imagem rotacionada em 45 graus", img_rotacionada)
```

4 - Rotacionar a imagem com relação ao centro do cão localizado mais a direita e abaixo, em -30 graus.



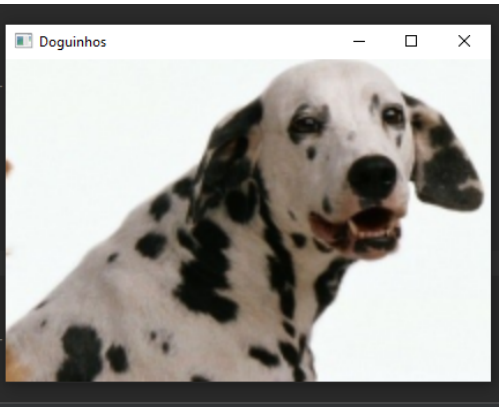

```python
(alt, lar) = im.shape[:2] #captura altura e largura
pin = (lar, alt) #acha o centro
M = cv2.getRotationMatrix2D(pin, -30, 1.0) #30 graus
img_rotacionada = cv2.warpAffine(im, M, (lar, alt))
cv2.imshow("Imagem rotacionada em 45 graus", img_rotacionada)
```

5 - Reduzir a imagem do maior cão pela metade.



6 - Ampliar a imagem do menor cão em 2 vezes.

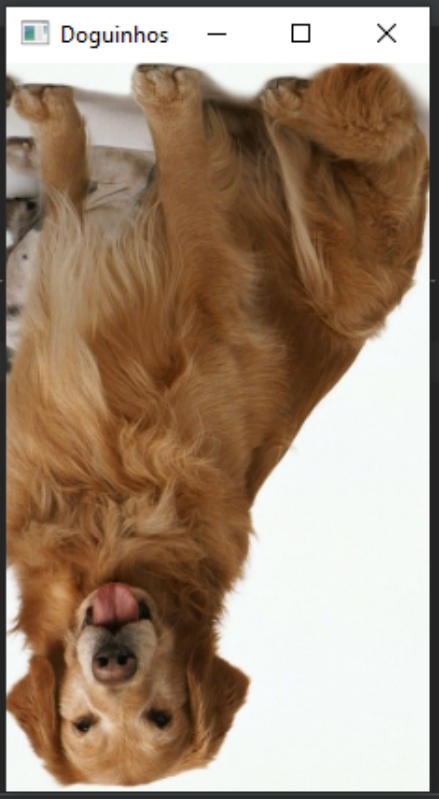
```
image = cv2.flip(im, 1)
'''

# ----------------------------------------------------------

def zoom(img, zoom_factor):
    return cv2.resize(img, None, fx=zoom_factor, fy=zoom_factor)


recorte_menor = im[140:280, 320:530]

zoomed = zoom(recorte_menor, 2)

# ----------------------------------------------------------

cv2.imshow("Doguinhos", zoomed)
```



7 - Fazer um flip de -1, 0 e 1 com a imagem individual do maior e menor cão.

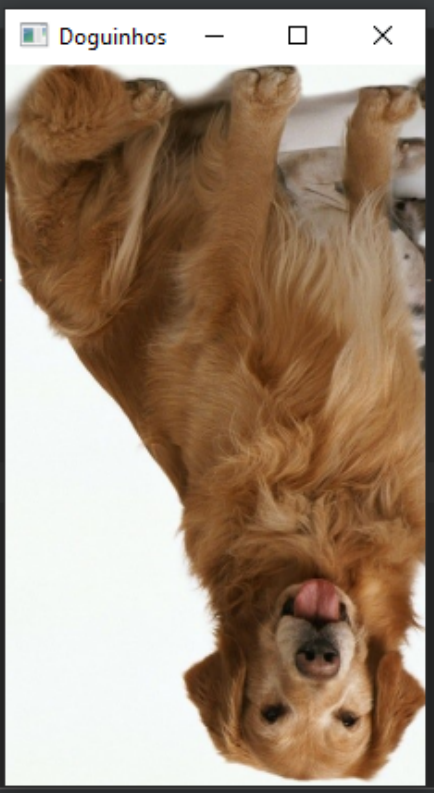
main.py

```
[EXERC 7]
image = cv2.flip(im, -1)
image = cv2.flip(im, 0)
image = cv2.flip(im, 1)
'''

# ----------------------------------------------------------

recorte_maior = im[140:530, 93:320]
recorte_menor = im[140:280, 320:530]

image = cv2.flip(recorte_maior, -1)

'''

image = cv2.flip(recorte_maior, 0)
image = cv2.flip(recorte_maior, 1)

image = cv2.flip(recorte_menor, -1)
```



main.py

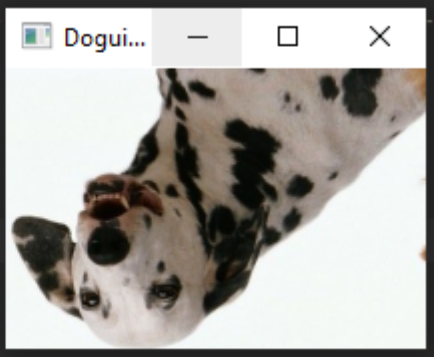
```
103     [EXERC 7]
104     image = cv2.flip(im, -1)
105     image = cv2.flip(im, 0)
106     image = cv2.flip(im, 1)
107     '''
108
109     # ----------------------------------------
110
111     recorte_maior = im[140:530, 93:320]
112     recorte_menor = im[140:280, 320:530]
113
114     image = cv2.flip(recorte_maior, 0)
115
116     '''
117
118
119     image = cv2.flip(recorte_maior, 1)
120
121     image = cv2.flip(recorte_menor, -1)
```



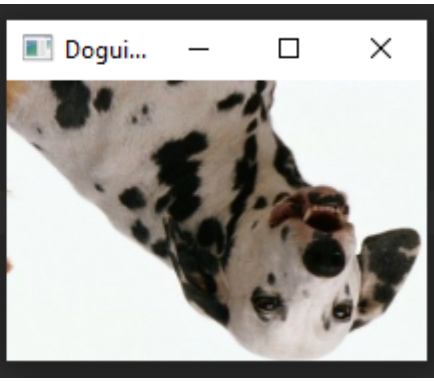
main.py

```
103     [EXERC 7]
104     image = cv2.flip(im, -1)
105     image = cv2.flip(im, 0)
106     image = cv2.flip(im, 1)
107     '''
108
109     # ----------------------------------------
110
111     recorte_maior = im[140:530, 93:320]
112     recorte_menor = im[140:280, 320:530]
113
114     image = cv2.flip(recorte_maior, 1)
115
116     '''
117
118
119
120
121     image = cv2.flip(recorte_menor, -1)
```

```
# ----------------------------------------

recorte_maior = im[140:530, 93:320]
recorte_menor = im[140:280, 320:530]

image = cv2.flip(recorte_menor, -1)

'''
image = cv2.flip(recorte_menor, 0)
```




```
recorte_maior = im[140:530, 93:320]
recorte_menor = im[140:280, 320:530]

image = cv2.flip(recorte_menor, 0)

'''
```




```
# ----------------------------------------

recorte_maior = im[140:530, 93:320]
recorte_menor = im[140:280, 320:530]

image = cv2.flip(recorte_menor, 1)

# ----------------------------------------
```