Reconhecimento de Placas de Regulamento de Trânsito

Passo-a-passo Geral

- Ler a imagem de Teste colorida
- Usar a Transformada de Hough para círculo apenas na camada vermelha
  - Aqui existe a dificuldade de encontrar a quantidade esperada de círculos
- Ler o modelo com as placas que se quer reconhecer
- Usar o Template Matching para identificar qual é a placa

Pré-tarefas

- Antes de se fazer o script final, vamos fazer duas pré-tarefas:
  - Entender os parâmetros da Transformada de Hough para círculos
  - Compreensão do Template Matching

Pré-tarefa 1: Transformada de Hough para Círculo

- http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghcircles/py_houghcircles.html#
- http://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=houghcircles

Pré-tarefa 1: Transformada de Hough para Círculo

- pratos = cv2.imread('teste7.jpg')
- cimg = pratos.copy()
- gray = cv2.cvtColor(pratos,cv2.COLOR_BGR2GRAY)
- **# Faça testes com a camada vermelha e depois faça o teste com o cinza**
- **# Faça testes variando a distância de 20 para 2 ou 200**
- circles = cv2.HoughCircles(pratos[:,:,2],cv2.HOUGH_GRADIENT,1,20)
- circles = np.uint16(np.around(circles))
- for i in circles[0,:]:
- # desenha o círculo
- cv2.circle(cimg,(i[0],i[1]),i[2],(0,255,0),2)
- # desenha o centro do círculo
- cv2.circle(cimg,(i[0],i[1]),2,(0,0,255),3)
- cv2.imwrite('res_pratos.png', cimg)

Pré-tarefa 2: Template Matching

- http://docs.opencv.org/3.2.0/d4/dc6/tutorial_py_template_matching.html
- http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html?highlight=template%20matching

Pré-tarefa 2: Template Matching

- img = cv2.imread('placas_modelo.jpg',0)
- img2 = img.copy()
- template = cv2.imread('byc.png',0)
- w, h = template.shape[::-1]
- methods = ['cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED', 'cv2.TM_CCORR', 'cv2.TM_CCORR_NORMED', 'cv2.TM_SQDIFF', 'cv2.TM_SQDIFF_NORMED']
- for meth in methods:
- img = img2.copy()
- method = eval(meth)

Pré-tarefa 2: Template Matching

```
# Apply template Matching
  res = cv2.matchTemplate(img,template,method)
  min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)
  # If the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
  if method in [cv2.TM_SQDIFF, cv2.TM_SQDIFF_NORMED]:
      top_left = min_loc
  else:
      top_left = max_loc
  bottom_right = (top_left[0] + w, top_left[1] + h)
  cv2.rectangle(img,top_left, bottom_right, 255, 2)
  plt.subplot(121),plt.imshow(res,cmap = 'gray')
  plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
  plt.subplot(122),plt.imshow(img,cmap = 'gray')
  plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
  plt.suptitle(meth)
  plt.show()
```

Passo-a-passo Detalhado

- Ler a imagem de Teste colorida
- Leia a quantidade de círculos que se quer encontrar
- Usar a Transformada de Hough para círculo apenas na camada vermelha
  - Varie o parâmetro para que se encontre a quantidade de círculos indicado pelo usuário
  - Para cada uma das placas encontradas, fazer o recorte da placa, fazer o resize para o tamanho 60,60 e gravar em um arquivo.

Passo-a-passo Detalhado

- Ler o modelo com as placas 'placas_modelo.jpg'
- Para cada um dos arquivos de placa gerado no slide anterior:
  - Usar o Template Matching para identificar qual é a placa