Histograme in Python

November 9, 2020

Catalin Stoean

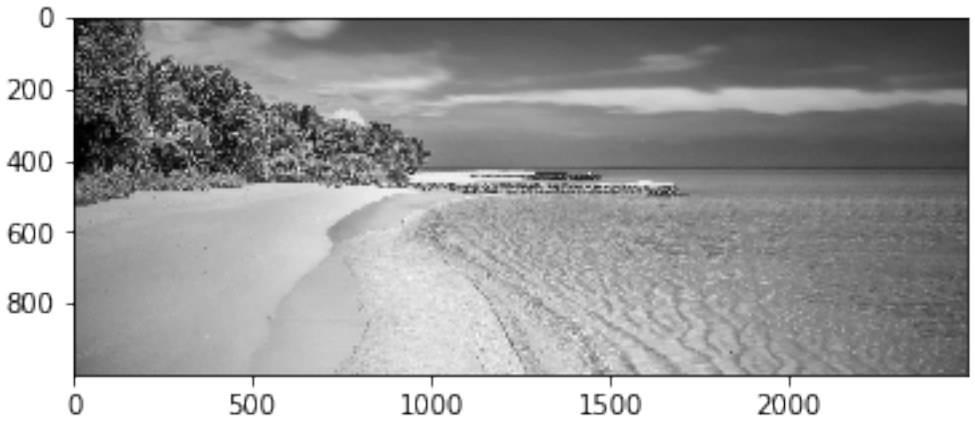
catalin.stoean@inf.ucv.ro

http://inf.ucv.ro/~cstoean

1 Calculam histograma pentru o poza grayscale

[22]:
```
import cv2
from matplotlib import pyplot as plt

img = cv2.imread('D:/pic.jpg', 0)

plt.imshow(img, cmap='gray')

hist = cv2.calcHist([img],[0],None,[256],[0,256])

print('Numarul de elemente din hist:', len(hist))
print('Elementele de la 100 la 110\n', hist[100:110])
```

```
Numarul de elemente din hist: 256
Elementele de la 100 la 110
 [[12000.]
 [12180.]
 [12960.]
 [12989.]
 [12857.]
 [13131.]
 [13495.]
 [13561.]
 [14154.]
 [14097.]]
```

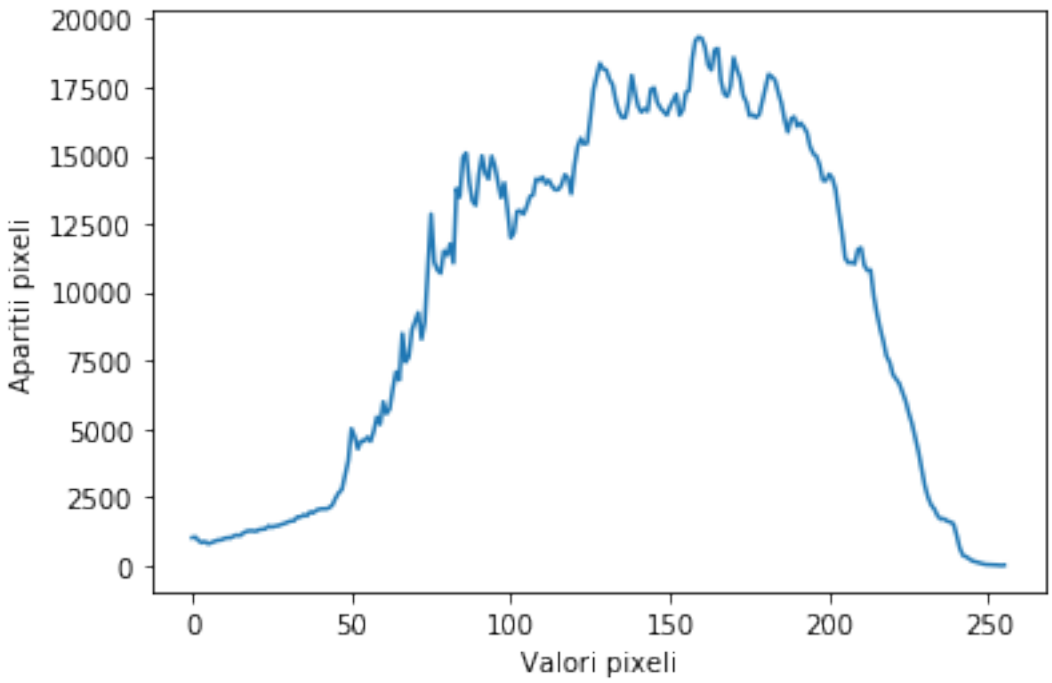


void cv::calcHist(InputArrayOfArrays images, const std::vector< int > & channels, InputArray mask, OutputArray hist, const std::vector< int > & histSize, const std::vector< float > & ranges, bool accumulate = false)
Python: hist = cv.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])

2 Afisam histograma pentru o poza grayscale

[25]:
```
from matplotlib import pyplot as plt

plt.plot(hist)

plt.xlabel('Valori pixeli')
plt.ylabel('Aparitii pixeli')
```

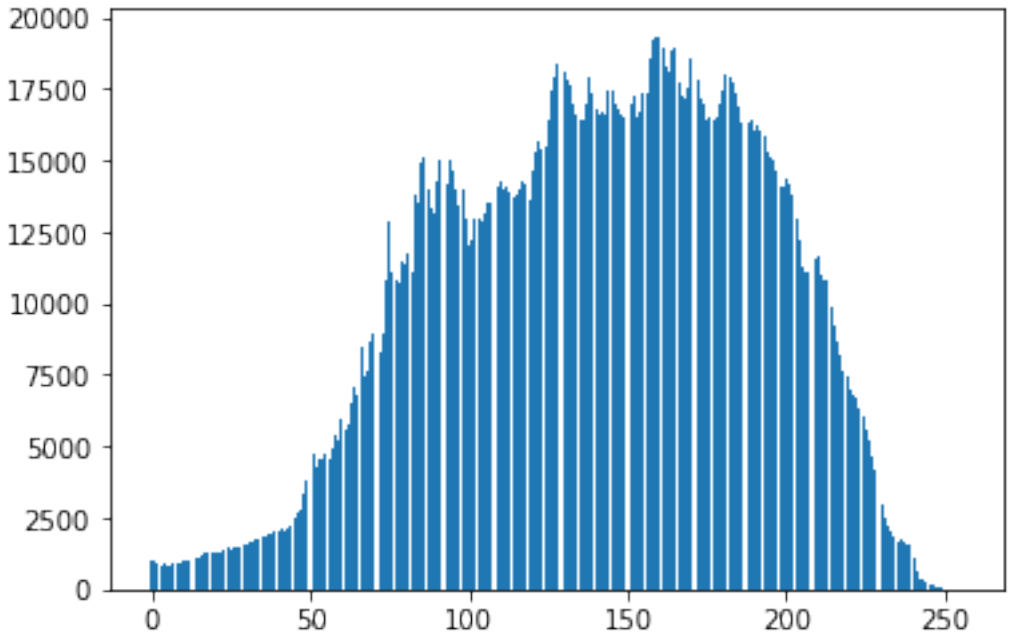
[25]:
```
Text(0, 0.5, 'Aparitii pixeli')
```



[26]:
```
x = list(range(0, 256))
valoriHist = [item for sublist in hist for item in sublist]

print('Primele 10 valori din x:', x[:10])
print('Primele 10 valori din hist:\n', hist[:10])
print('Primele 10 valori din valoriHist:', valoriHist[:10])

plt.bar(x, valoriHist)
```

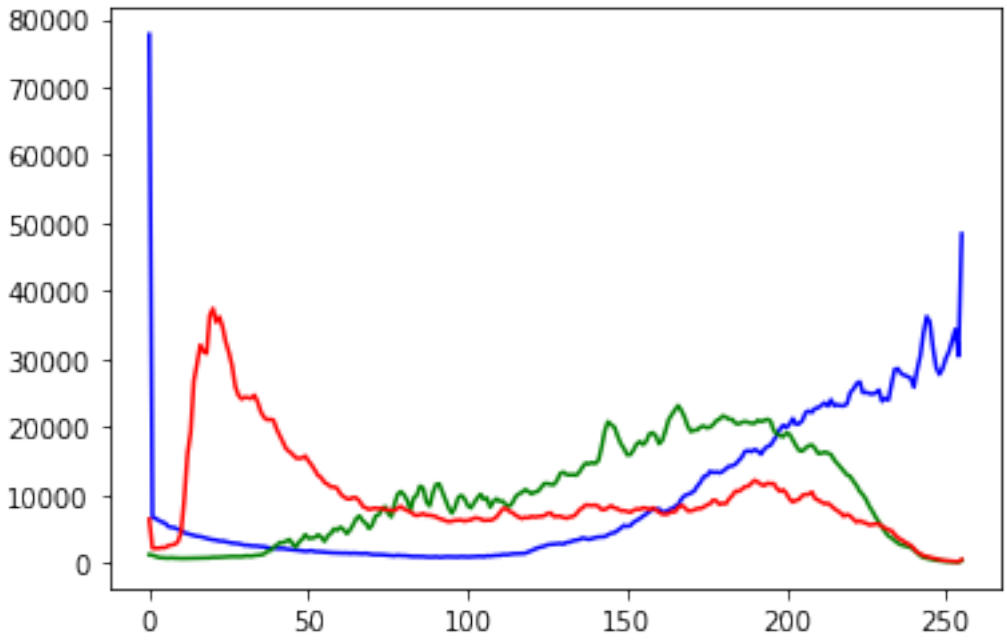
```
Primele 10 valori din x: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Primele 10 valori din hist:
 [[1025.]
 [1051.]
 [ 919.]
 [ 840.]
 [ 884.]
 [ 800.]
 [ 843.]
 [ 883.]
 [ 934.]
 [ 941.]]
Primele 10 valori din valoriHist: [1025.0, 1051.0, 919.0, 840.0, 884.0, 800.0,
843.0, 883.0, 934.0, 941.0]
```

[26]:
```
<BarContainer object of 256 artists>
```



3 Afisam histograma pentru o poza color

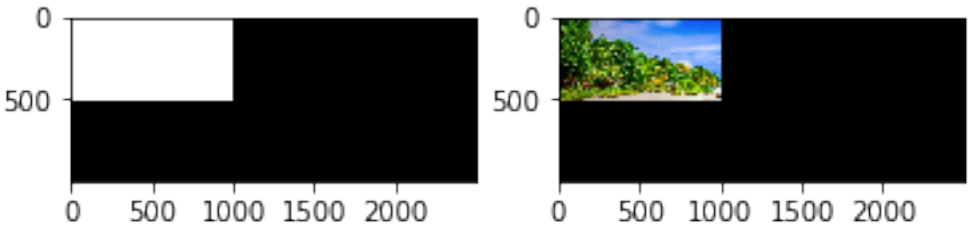
[28]:
```
from matplotlib import pyplot as plt
import numpy as np

imgColor = cv2.imread('D:/pic.jpg')

color = ['b','g','r']
for i, col in enumerate(color):
    histColor = cv2.calcHist([imgColor],[i],None,[256],[0,256])
    plt.plot(histColor, color = col)
plt.show()
```



4 Masca in OpenCV

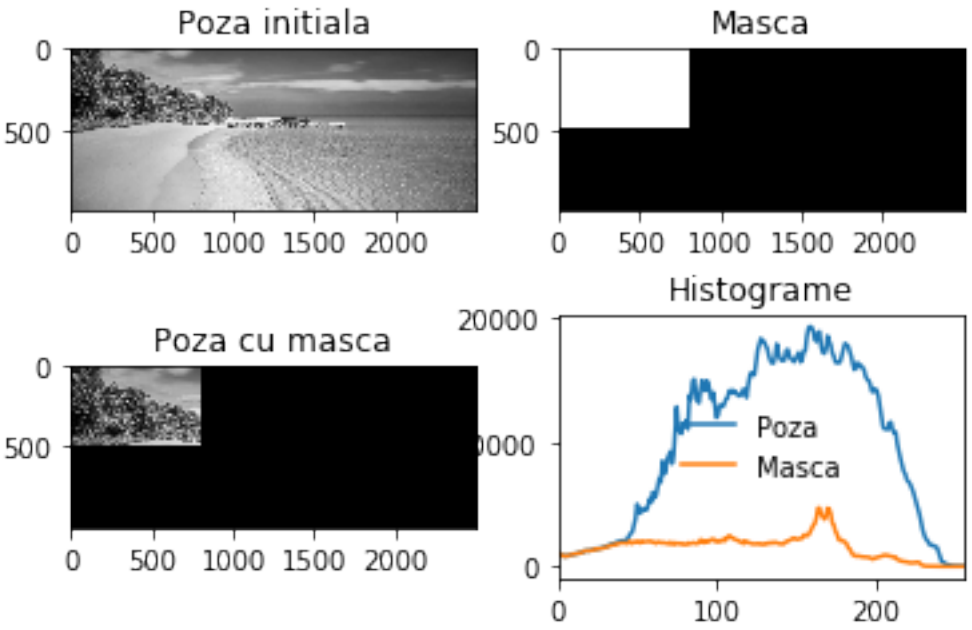
[22]:
```
import numpy as np
im = cv2.imread('D:/pic.jpg')

# Cream o imagine neagra
masca = np.zeros(im.shape, dtype = "uint8")

# Desenam un dreptunghi alb plin pe imaginea masca
cv2.rectangle(masca, (0, 0), (1000, 500), (255, 255, 255), -1)
im = cv2.cvtColor(im, cv2.COLOR_BGR2RGB)
# Aplicam masca si afisam masca si rezultatul
#Era OK sa scriem si ca in comentariul de mai jos
# imagineaMascata = im & masca
imagineaMascata = cv2.bitwise_and(im, masca)
fig = plt.figure()
ax1 = fig.add_subplot(121)
ax1.imshow(masca)
ax2 = fig.add_subplot(122)
ax2.imshow(imagineaMascata)
```

[22]:
```
<matplotlib.image.AxesImage at 0x20c07493c48>
```



5 Aplicam o masca pentru o imagine si calculam histograma

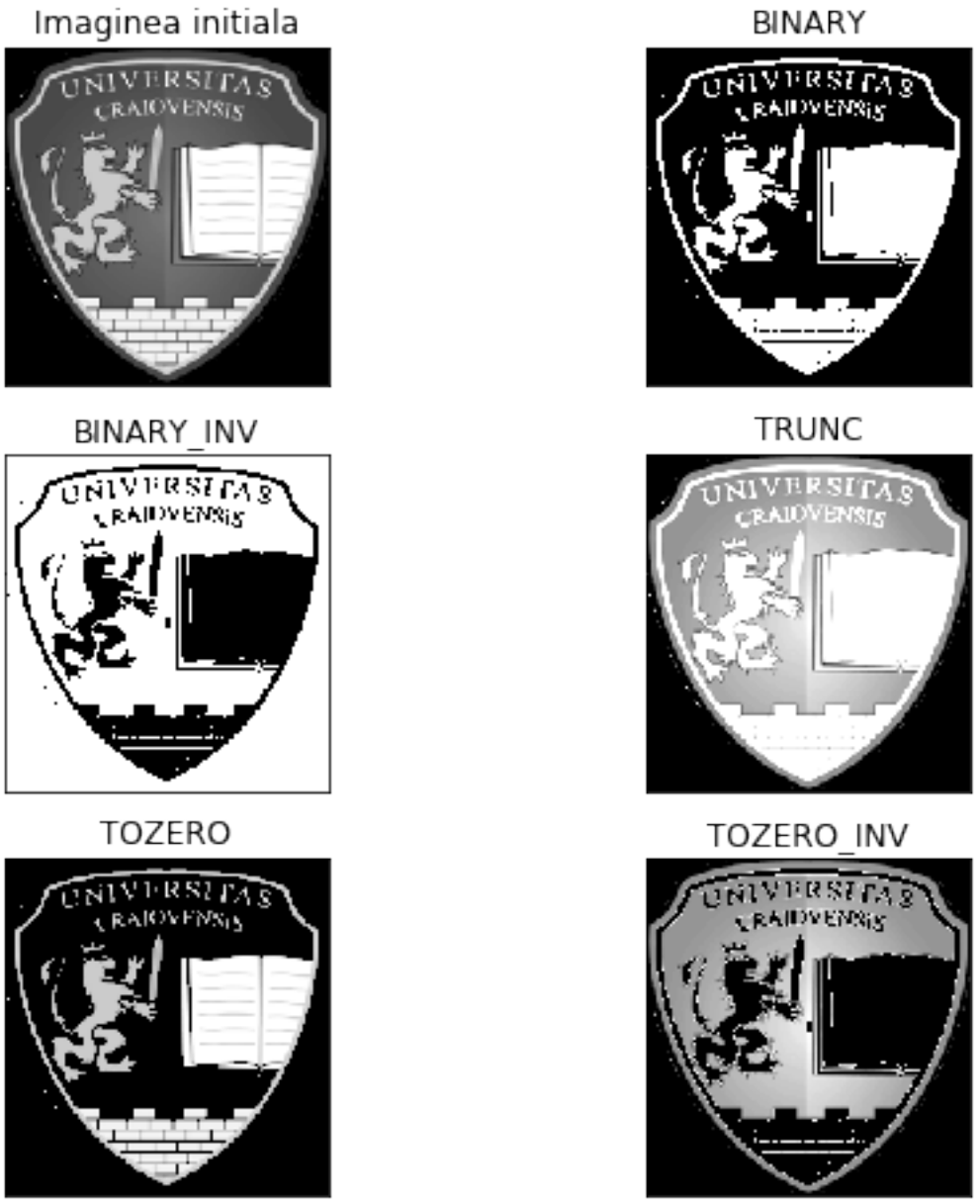
[30]:
```
import numpy as np

img = cv2.imread('D:/pic.jpg', 0)
# cream o masca

masca = np.zeros(img.shape[:2], np.uint8)
masca[0:500, 0:800] = 255
mascaImg = cv2.bitwise_and(img, masca)

# Calculam histograma cu si fara masca
histPoza = cv2.calcHist([img],[0],None,[256],[0,256])
histMasca = cv2.calcHist([img],[0],masca,[256],[0,256])

plt.subplot(221)
plt.imshow(img, 'gray')
plt.title('Poza initiala')
plt.subplot(222)
plt.imshow(masca,'gray')
plt.title('Masca')
plt.subplot(223)
plt.imshow(mascaImg, 'gray')
plt.title('Poza cu masca')
plt.subplot(224)
plt.plot(histPoza, label='Poza')
plt.plot(histMasca, label='Masca')
plt.legend(frameon=False)
plt.title('Histograme')
plt.xlim([0,256])
plt.show()
```



6 Thresholding

[31]:
```
logo = cv2.imread('D:/ucv.png', 0)
h, w = logo.shape

print('w = {}, h = {}'.format(w, h))

ret,thresh1 = cv2.threshold(logo, 127, 255, cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(logo, 127, 255, cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(logo, 127, 255, cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(logo, 141, 255, cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(logo, 127, 255, cv2.THRESH_TOZERO_INV)
titles = ['Imaginea␣
 ↪initiala','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [logo, thresh1, thresh2, thresh3, thresh4, thresh5]
fig = plt.figure(figsize=(8, 8))
for i in range(6):
    plt.subplot(3,2,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])
plt.show()
```

```
w = 200, h = 208
```

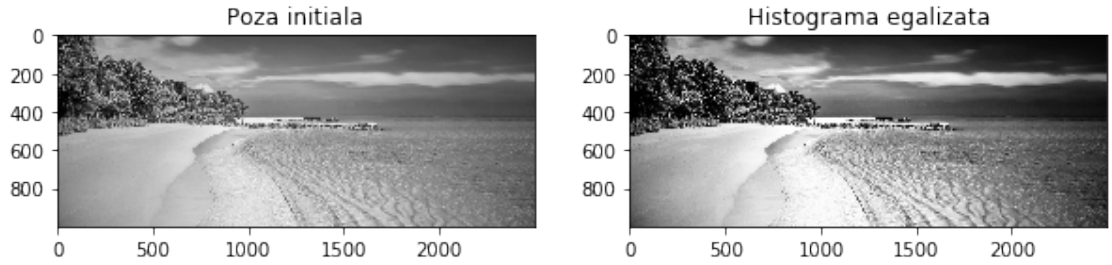


7 Egalizarea histogramei

[33]:
```
img = cv2.imread('D:/pic.jpg', 0)
equ = cv2.equalizeHist(img)

f = plt.figure(figsize=(10,4))
ax = f.add_subplot(121)
ax2 = f.add_subplot(122)
```

```
ax.imshow(img, 'gray')
ax.set_title('Poza initiala')
ax2.imshow(equ, cmap='gray')
ax2.set_title('Histograma egalizata')
```

[33]: Text(0.5, 1.0, 'Histograma egalizata')



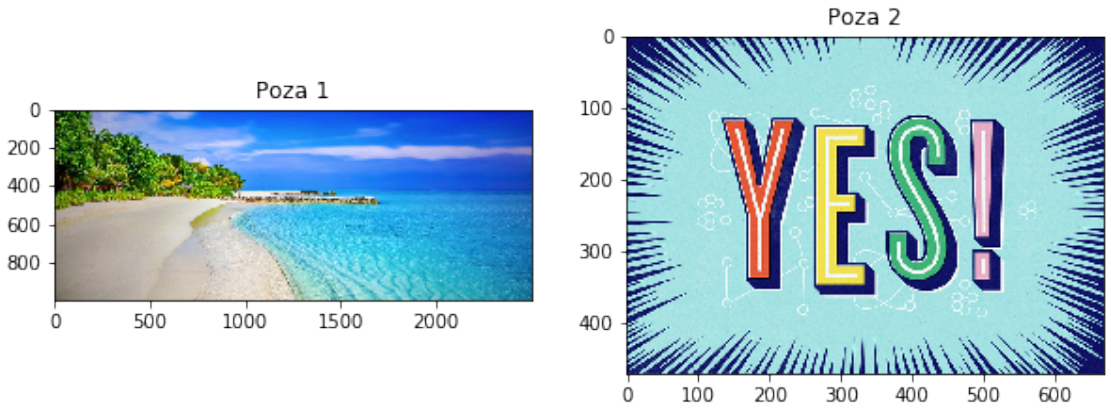
8 Comparare histograme

[33]:
```
import cv2
from matplotlib import pyplot as plt

img1 = cv2.imread('D:/pic.jpg')
img2 = cv2.imread('D:/poza.jpg')

# Calculam histogramele color
hist_img1 = cv2.calcHist([img1], [0, 1, 2], None, [256, 256, 256], [0, 256, 0,␣
↪256, 0, 256])
hist_img2 = cv2.calcHist([img2], [0, 1, 2], None, [256, 256, 256], [0, 256, 0,␣
↪256, 0, 256])

#calculam diferenta
diferenta = cv2.compareHist(hist_img1, hist_img2, cv2.HISTCMP_BHATTACHARYYA)

print('Diferenta folosind Bhattacharyya:', diferenta)

#Afisam si imaginile
img1RGB = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img2RGB = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)

f = plt.figure(figsize=(10,4))
ax = f.add_subplot(121)
ax2 = f.add_subplot(122)
ax.imshow(img1RGB)
ax.set_title('Poza 1')
```

```
ax2.imshow(img2RGB)
ax2.set_title('Poza 2')
```

```
Diferenta folosind Bhattacharyya: 0.9771913519893658
```

[33]:
```
Text(0.5, 1.0, 'Poza 2')
```



In loc de cv2.HISTCMP_BHATTACHARYYA se pot folosi alte masuri:

HISTCMP_CORREL: Correlation

HISTCMP_CHISQR: Chi-Square

HISTCMP_INTERSECT: Intersection

HISTCMP_BHATTACHARYYA: Bhattacharyya distance

HISTCMP_CHISQR_ALT: Alternative Chi-Square

HISTCMP_HELLINGER: Synonym for CV_COMP_BHATTACHARYYA

HISTCMP_KL_DIV: Kullback-Leibler divergence

[ ]: