Laporan Tugas PCD

Sari Rahmawati 1103130236

Soal 1

Memisahkan gambar per plan RGB , yaitu plan blue , plan green , plan red

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('rgb.png')

b,g,r = cv2.split(img)

cv2.imshow('image',img)

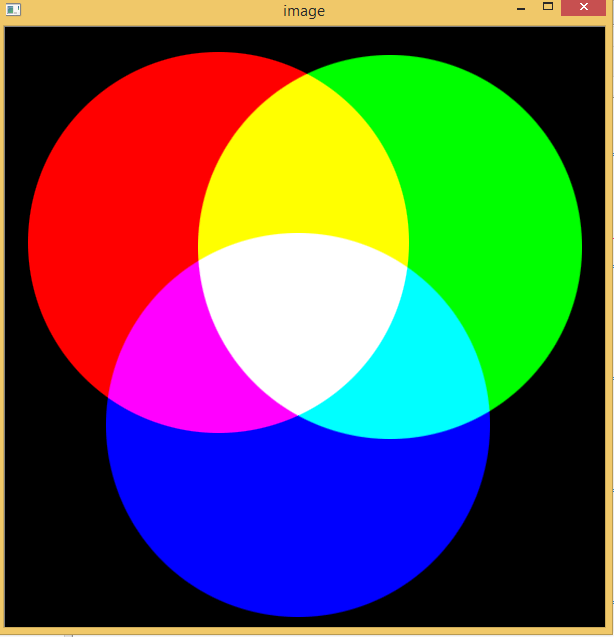
cv2.imshow('image blue ',b)

cv2.imshow('image red ',r)

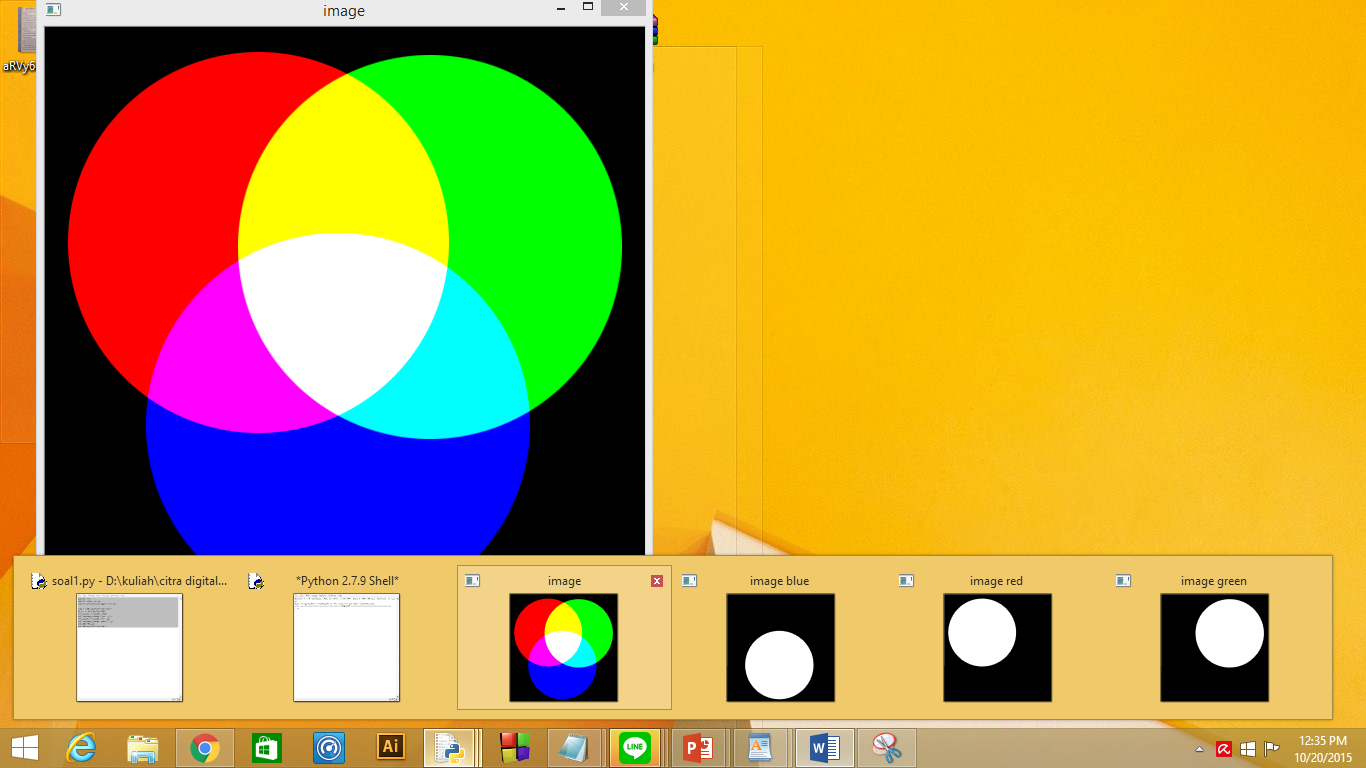
cv2.imshow('image green ',g)

cv2.waitKey(0)

cv2.destroyAllWindows()



gambar gambar asli



gambar gambar per plane rgb

Flipping

import cv2

import numpy as np

img2 = cv2.imread('jbc.jpg')

def flip\_vertical(img):

rows,cols,ch = img.shape

flipVertical = np.zeros((rows,cols,ch),np.uint8)

rows2 = rows - 1

for i in range (0,rows-1):

for j in range(0,cols-1):

flipVertical[rows2,j]=img[i,j]

rows2=rows2-1

return flipVertical

def flip\_horizontal(img):

rows,cols,ch = img.shape

flipHorizontal = np.zeros((rows,cols,ch),np.uint8)

cols2 = cols - 1

for i in range (0,cols-1):

for j in range(0,rows-1):

flipHorizontal[j,cols2]=img[j,i]

cols2=cols2-1

return flipHorizontal

result=flip\_vertical(img2)

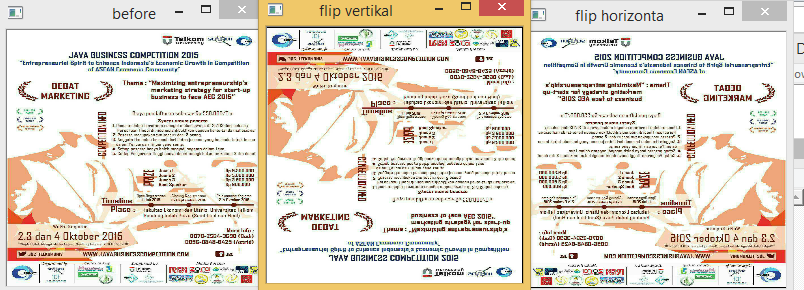
result2=flip\_horizontal(img2)

cv2.imshow('flip vertikal',result)

cv2.imshow('flip horizonta',result2)

cv2.waitKey(0)

cv2.destroyAllWindows()



gambar flip

Kuantisasi

import numpy as np

import cv2

img = cv2.imread('messi.jpg')

Z = img.reshape((-1,3))

Z = np.float32(Z)

criteria = (cv2.TERM\_CRITERIA\_EPS + cv2.TERM\_CRITERIA\_MAX\_ITER, 10, 1.0)

K = 16

ret,label,center=cv2.kmeans(Z,K,None,criteria,10,cv2.KMEANS\_RANDOM\_CENTERS)

center = np.uint8(center)

res = center[label.flatten()]

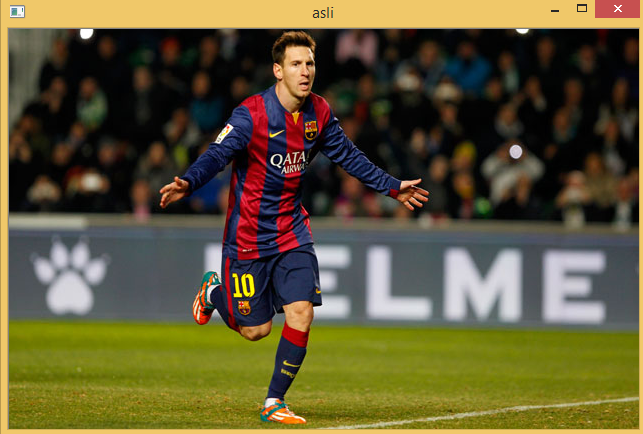
res2 = res.reshape((img.shape))

cv2.imshow('asli',img)

cv2.imshow('kuantisasi',res2)

cv2.waitKey(0)

cv2.destroyAllWindows()



gambar gambar asli



gambar hasil

Masking

import numpy as np

import cv2

import matplotlib.pyplot as plt

im = cv2.imread('messi.jpg')

img2= cv2.imread('wordpress.jpg')

im=cv2.resize(im,(512,512))

img2=cv2.resize(img2,(512,512))

img2gray = cv2.cvtColor(img2,cv2.COLOR\_BGR2GRAY)

ret, mask = cv2.threshold(img2gray, 10, 255, cv2.THRESH\_BINARY)

mask=cv2.bitwise\_not(mask)

masked\_data = cv2.bitwise\_and(im, im, mask=mask)

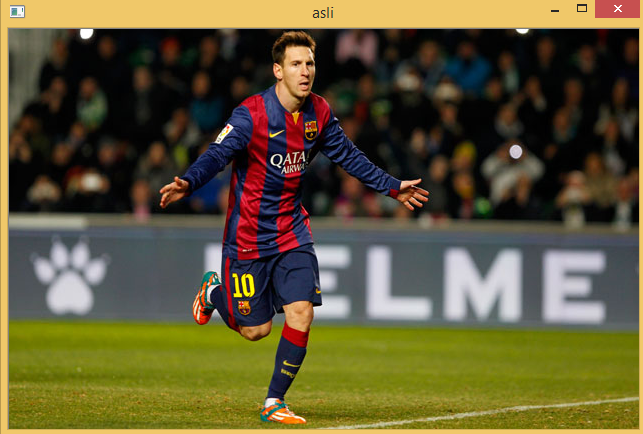
cv2.imshow("messi", im)

cv2.imshow("mask", img2)

cv2.imshow("masked", masked\_data)

cv2.waitKey(0)

cv2.destroyAllWindows()



gambar gambar1



gambar gambar 2



gambar hasil masking

Copy, Paste

import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('jbc.jpg')

rows, cols, ch = img.shape

piece = np.zeros((100,100,ch),np.uint8)

a = 0

b = 0

for i in range(100,200):

for j in range(100,200):

piece[a,b] = img[i,j]

img[i,j] = 0

b += 1

b = 0

a += 1

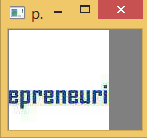
cv2.imshow('img',img)

cv2.imshow('piece',piece)

cv2.waitKey(0)

cv2.destroyAllWindows()





gambar hasil crop

Blending

Blending untuk menggabungkan 2 buah gambar menjadi satu gambar yang bertumpuk

import cv2

import numpy as np

import matplotlib.pyplot as plt

img1 = cv2.imread('ss.png')

img2 = cv2.imread('wordpress.jpg')

resized\_image1 = cv2.resize(img1, (500, 500))

resized\_image2 = cv2.resize(img2, (500, 500))

j=0

i=1

while j<=1 :

result = cv2.addWeighted(resized\_image1,j,resized\_image2,i,0)

cv2.imshow('final',result)

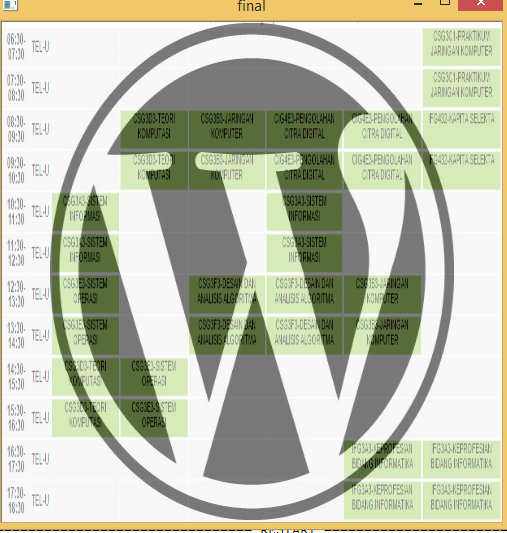
cv2.waitKey(200)

j+=0.10

i-=0.10

cv2.waitKey(0)

cv2.destroyAllWindows()



Rotasi (45 derajat)

Memutar gambar sebesar 45 derajat

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('jbc.jpg')

img = cv2.resize(img, (256, 256))

rows ,cols ,k = img.shape

M = cv2.getRotationMatrix2D((cols/2,rows/2),45,1)

dst = cv2.warpAffine(img,M,(cols,rows))

cv2.imshow('image before',img)

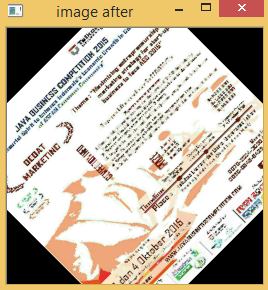
cv2.imshow('image after ',dst)

cv2.waitKey(0)

cv2.destroyAllWindows()



gambar Before Rotate



gambar after rotate

Zoom In

img = cv2.imread('jbc.jpg')

img= cv2.resize(img, (500, 500))

def zoomin(img):

rows,cols,ch = img.shape

zin = np.zeros((rows\*2,cols\*2,ch),np.uint8)

m=0

n=0

for i in range(rows-1):

for j in range(cols-1):

zin[m,n]=img[i,j]

zin[m,n+1]=img[i,j]

zin[m+1,n]=img[i,j]

zin[m+1,n+1]=img[i,j]

n=n+2

m=m+2

n=0

return zin

cv2.imshow('zoomin',zoomin(img))

cv2.waitKey(0)

cv2.destroyAllWindows()



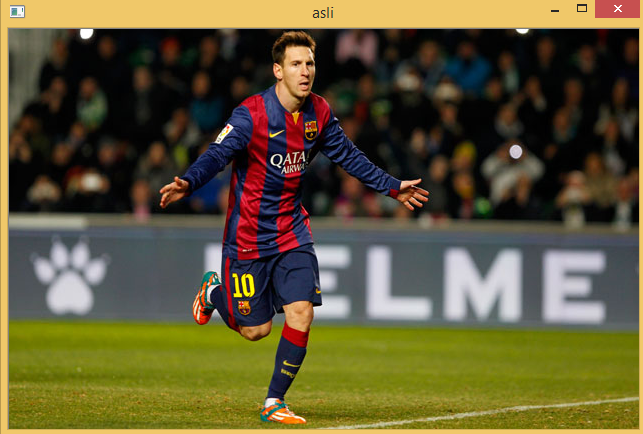
gambar before zoomin



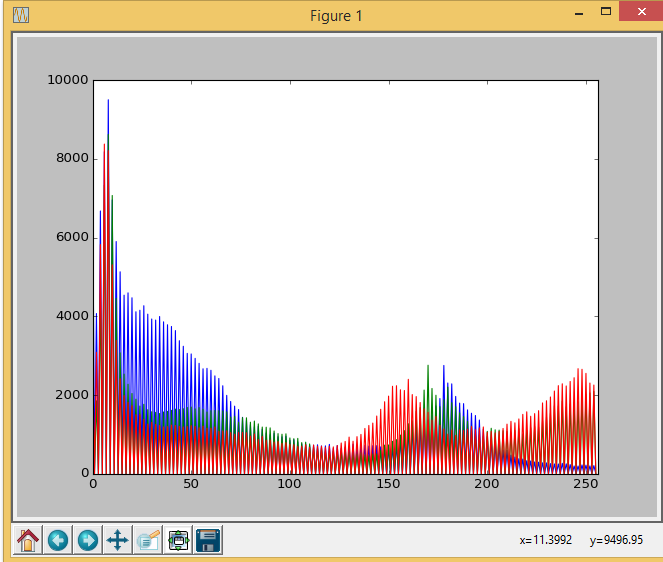
gambar afterzoomin

Plane Histogram

Membagi histogram per plane yang ada



gambar Gambar Asli



gambar histogram per plane

Ekualisasi

import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('messi.jpg')

cv2.imshow('asli',img)

hist,bins = np.histogram(img.flatten(),256,[0,256])

cdf = hist.cumsum()

cdf\_m = np.ma.masked\_equal(cdf,0)

cdf\_m = (cdf\_m - cdf\_m.min())\*256/(cdf\_m.max()-cdf\_m.min())

cdf = np.ma.filled(cdf\_m,0).astype('uint8')

img= cdf[img]

cv2.imshow('result',img)

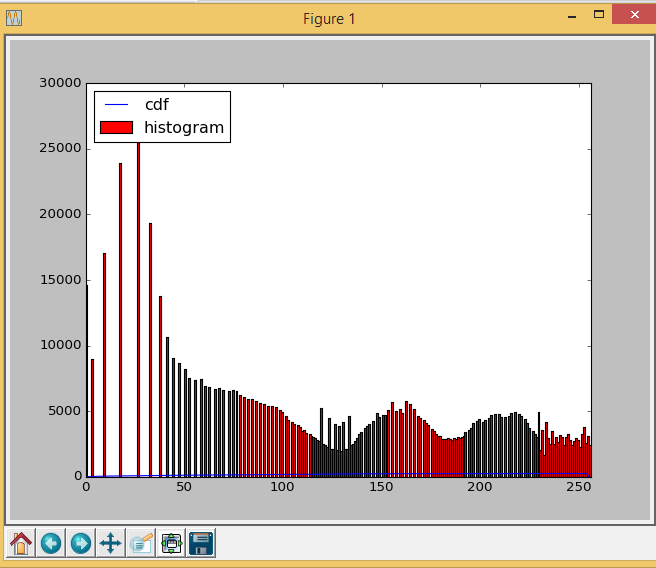
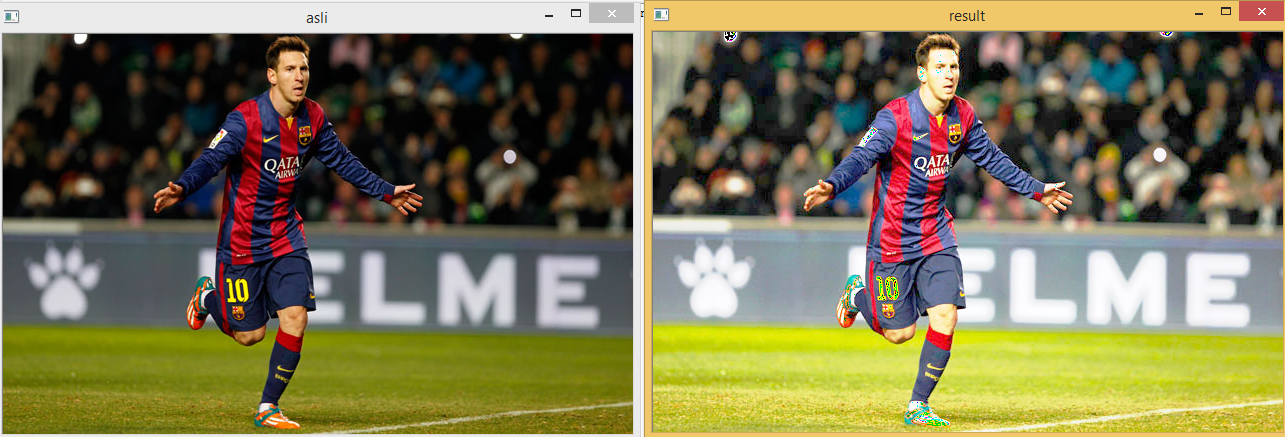
plt.plot(cdf, color = 'b')

plt.hist(img.flatten(),256,[0,256], color = 'r')

plt.xlim([0,256])

plt.legend(('cdf','histogram'), loc = 'upper left')

plt.show()



Fourier Transform

import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('messi.jpg',0)

dft = cv2.dft(np.float32(img),flags = cv2.DFT\_COMPLEX\_OUTPUT)

dft\_shift = np.fft.fftshift(dft)

magnitude\_spectrum = 10\*np.log(cv2.magnitude(dft\_shift[:,:,0],dft\_shift[:,:,1]))

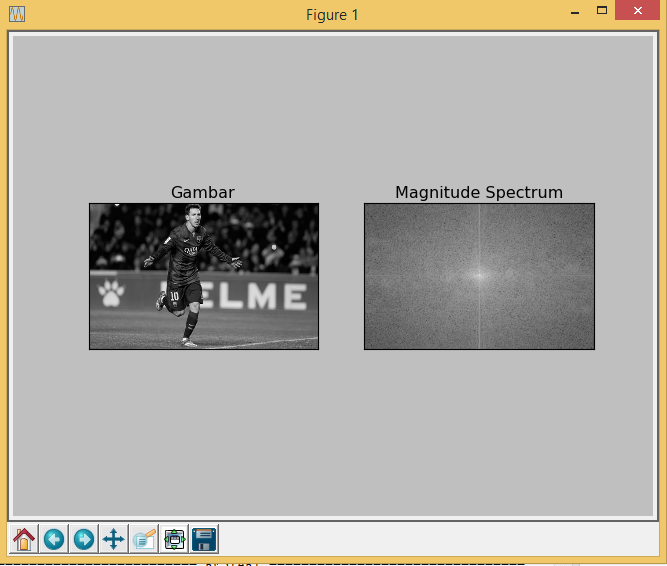
plt.subplot(121),plt.imshow(img, cmap = 'gray')

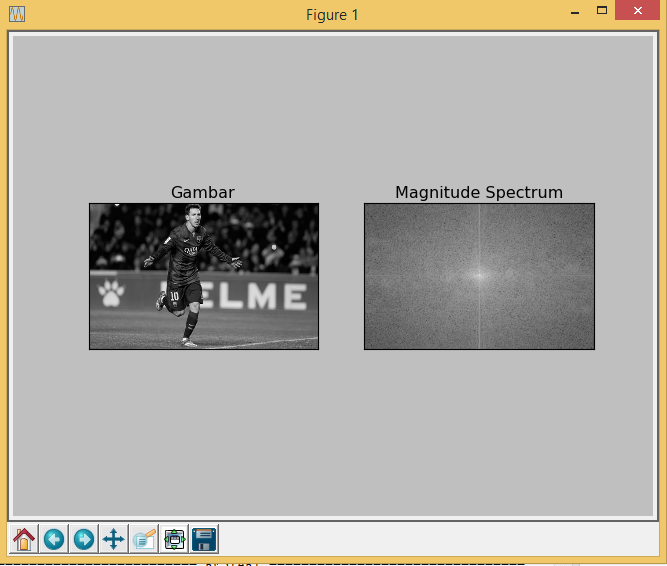
plt.title('Gambar'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(magnitude\_spectrum, cmap = 'gray')

plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])

plt.show()





Sampling

import cv2

import numpy as np

img = cv2.imread('messi.jpg')

img = cv2.resize(img, (512, 512))

r,c,t = img.shape

coba = np.zeros((r,c,3),np.uint8)

x=4

sum=[0,0,0]

for i in range(0,r,x) :

for j in range(0,c,x) :

for m in range(x) :

for n in range(x) :

for k in range(t):

sum[k]=img[j+n,i+m][k]

for k in range(3):

sum[k] =sum[k]/(x\*x)

for m in range(x):

for n in range(x):

coba[j+n,i+m]=sum

sum=[0,0,0]

cv2.imshow('img',img)

cv2.imshow('imgsampling',coba)

cv2.waitKey(0)

cv2.destroyAllWindows()



