Image Processing dersin kapsamında yaptığım projeyi görebilirsiniz. Soruları ve nasıl cevapladığımın ayrıntılı raporu aşağıdadır. Ayrıca projede kullanılan görüntüler yüklenmiştir.

1. **A large variety of birds make the migration journey, and bird migration is one of the astonishments of the natural world. In this respect, you are given an image acquired during the bird migration in the South Pole of the Earth (birdmigration.jpg).**

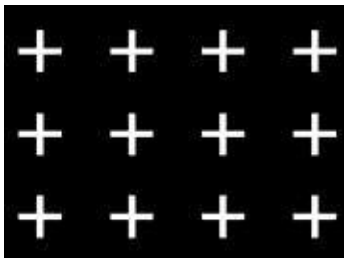
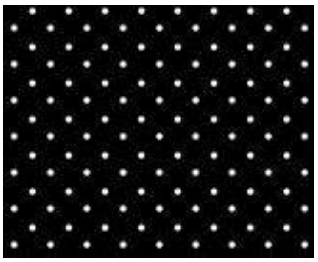
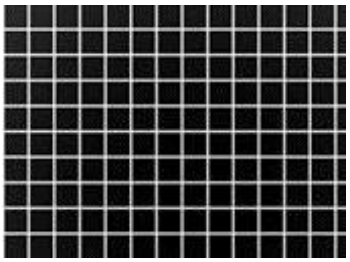
a) **Split the given RGB image into 3 different grayscale images, and save the outputs in "tif" format.**

b) **Create histogram of each grayscale image created in (a).**

c) **Create image negatives of each grayscale image created in (a). (must be done without a python library)**

d) **With the help of one the grayscale images created in (a) or in (c), design and develop an algorithm (using a python code) that automatically counts the number of birds migrating. (must be done without a python library)**

2. **Implement a Python code, which generates synthetic images with a size of 400x600 pixels in grayscale with one of the following patterns or any other pattern of your choice on a black background (must be done without a python library):**

3. **You are given a Portrait of a Young Woman which is a small oil-on-oak panel painting completed in 1470 (Portrait_of_a_Young_Woman.jpg). It marks a major stylistic advance in contemporary portraiture; the girl is set in an airy, three-dimensional, realistic setting, and stares out at the viewer with a complicated expression that is reserved, yet intelligent and alert.**

However, because this picture belongs to a 15th century painting, we clearly see the deformation and cracks within the picture.

Find a way to digitally improve the quality of this picture with a Python code (30 points). (must be done without a python library)

REPORT

In first question, firstly I read and showed the bird migration image by using cv2 library. This library provides showing,reading and writing(saving) images. Also it covers some functions but I have not used them. I have used imread() for reading image, splitted R,G,B formats and used imshow() R,G,B. And then I have saved that forms by using imwrite() function.

```
img=cv2.imread('bird-migration.jpg')

cv2.imshow('bird-migration.jpg',img)

cv2.waitKey(0)

cv2.destroyAllWindows() img_B =

img[:,:,0] img_G = img[:,:,1] img_R =

img[:,:,2]

cv2.imshow('image_B',img_B)

cv2.waitKey(0) cv2.destroyAllWindows()

cv2.imshow('image_G',img_G)

cv2.waitKey(0) cv2.destroyAllWindows()

cv2.imshow('image_R',img_R)

cv2.waitKey(0) cv2.destroyAllWindows()

status1 = cv2.imwrite('bird_blue.tif',img_B)

status2 = cv2.imwrite('bird_green.tif',img_G)

status3 = cv2.imwrite('bird_red.tif',img_R)
```

And then second step of first question, I have created histogram tables by using matplotlib library.

```
hist1 = cv2.calcHist([img_B],[0],None,[256],[0,256])

plt.subplot(121),plt.imshow(img_B,'gray')

plt.subplot(122),plt.plot(hist1) plt.xlim([0,256])

plt.title('blue')

plt.show()
```

```
hist2 = cv2.calcHist([img_G],[0],None,[256],[0,256])
plt.subplot(221),plt.imshow(img_G,'gray')
plt.subplot(222),plt.plot(hist2) plt.xlim([0,256])
plt.title('green')
plt.show()

hist3 = cv2.calcHist([img_R],[0],None,[256],[0,256])
plt.subplot(231),plt.imshow(img_R,'gray')
plt.subplot(232),plt.plot(hist3) plt.xlim([0,256])
plt.title('red')
plt.show()
```

The third step of the first question, I must create negative of images. I thought the img_B as a matrix because the image has pixels and pixel values are stored matrices. I use transposing property of matrices, then I made a so easy mathematical operation. 1 – img_B. Thanks to this operation, image became negative.

```
negative_of_img_B = 1 - img_B negative_of_img_G
= 1 - img_G negative_of_img_R = 1 - img_R
```

Finally the last step of first question, I must count the birds. I have figured it out that solution. The birds are black in the image that I chose. If I import only black pixels, the birds will appear on the screen. So, I have created a loop and over every pixel. And then I tried some numbers like 127, 0 , 100 , 50 and 60. I chosed 60 pixel. Because when I write greater than 60 pixels the birds appeard, only birds no mountains no clouds.

```
#Creating zero matrix by using rows and columns of the image
r = img_B.shape[0] c = img_B.shape[1]

img_thres = np.zeros((r,c)) n_pix
= 0
```

```
#I created a thresold that counts black pixel because the birds are black in the image.
for y in range(0,r):    for x in range(0,c):        pixel = img_B[y,x]
      if pixel > 60:
n_pix = pixel       else:
         n_pix = 0
      img_thres[y,x] = n_pix

#Here you can see easily birds and count.
cv2.imshow('img',img_thres)
cv2.waitKey(0) cv2.destroyAllWindows()
```

In the second question, I have generated a black background that have size 400,600 and some patterns on it. I used PIL fort his question but only to generate a screen,size and importing pattern. I chosed '£' this pattern. And I wanted to develop my code so I added some spesifications like splitting that font and showing R,G,B. Because I saved my pattern and I wanted to see that spesificly.

```
i = Image.new("RGB", (400, 600), "black") #Variety,size and color of image draw
= ImageDraw.Draw(i,"RGBA")
w, h = i.size

space = 30

#creating pattern on the black font for n in range(space, w, space):
for x in range(space, h - space, space):        draw.text((n,
x),"£",fill="white", font=ImageFont.truetype("arial"))
i.save("synthetic.tif") #Saving pattern
img1=cv2.imread('synthetic.tif') #Reading pattern
```

```
#I splitted the pattern image to RGB
img1_B1 = img1[:,:,0] img1_G1 =
img1[:,:,1] img1_R1 = img1[:,:,2]

#Showing them
cv2.imshow('image_B1',img1_B1)
cv2.waitKey(0) cv2.destroyAllWindows()
cv2.imshow('image_G1',img1_G1)
cv2.waitKey(0) cv2.destroyAllWindows()
cv2.imshow('image_R1',img1_R1)
cv2.waitKey(0) cv2.destroyAllWindows()
```

In the third question, I must improve appearence of a portrait. I thought using Gaussian Filter. Because this filter can improve quality of images and also I like using it. Actually there are some filters that I can use but I chosed one. And then I made some searchings about processing of Gaussian filter because I must use that without spesific python function. So this filter is very useful with cv2 but I could not use that way. And then I generated a function called convolve2d. This function which takes an image and a kernel and returns the convolution of them. I added image padding and loop over every pixel of image. I filtered every pixel I think and clarified them. Also I have saved the filtered image and showed it.

```
image =cv2.imread('portrait_of_a_young_woman.jpg',0)

#This function which takes an image and a kernel and returns the convolution of them. def
convolve2d(image,kernel):

   #Creating zero padd of image     output = np.zeros_like(image)
image_padded = np.zeros((image.shape[0] + 2, image.shape[1] + 2))
image_padded[1:-1,1:-1] = image
```

```
   #Creating loop for every pixel of the image
for x in range(image.shape[1]):        for y in
range(image.shape[0]):
          #element-wise multiplication of the kernel and the image
output[y, x]=(kernel * image_padded[y: y+3, x: x+3]).sum()     return
output

#Kernel to be used Gaussian Blur
KERNEL = np.array([[1, 2, 1], [2, 4, 2], [1, 2, 1]])/16.0 #This values about gaussian blur
image_filtering = convolve2d(image, kernel=KERNEL)

#Saving filtered image
cv2.imwrite('filtered_image.jpg', image_sharpen)
#Showing new filtered image  filtered_image =
cv2.imread('filtered_image.jpg')
cv2.imshow('aaa',filtered_image) cv2.waitKey(0)
cv2.destroyAllWindows()
```