دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

نام و نام خانوادگی: پیمان هاشمی

شماره دانشجویی: 400131032

درس: تصویر پردازی رقمی

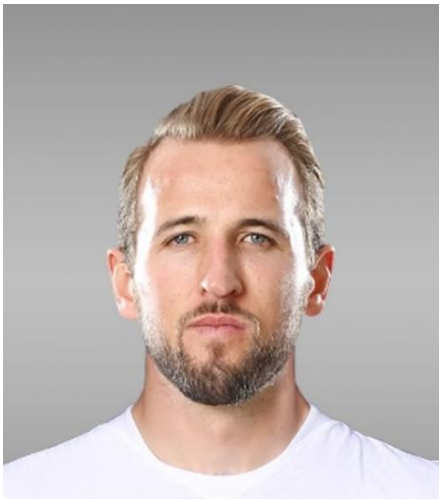
تمرین: تمرین شماره 1

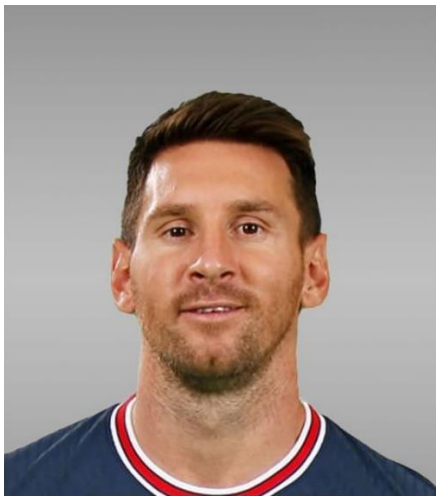
سوال ۱

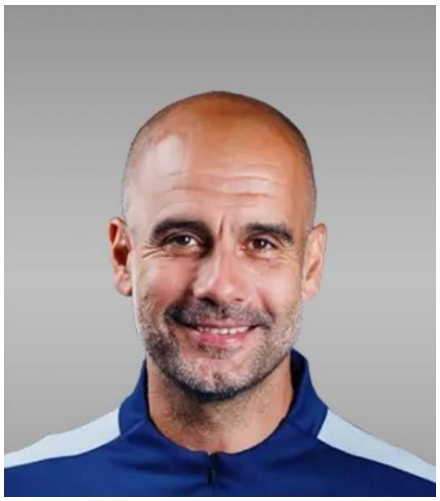
A

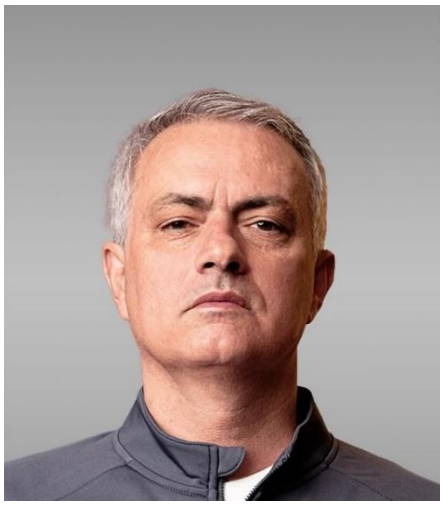
در این قسمت ابتدا عکس ها را خوانده و سپس آن را به ماتریس تبدیل میکنیم و با استفاده از دو حلقه for تو در تو، که اولی تعداد سطر ها را تعیین میکند و دومی ستون ها را بر عکس میکند و میخواند، ماتریس را خوانده و سپس به یک ماتریس جدید اضافه میکنیم.

و در نهایت ماتریس ها را به عکس تبدیل میکینم.









B,C

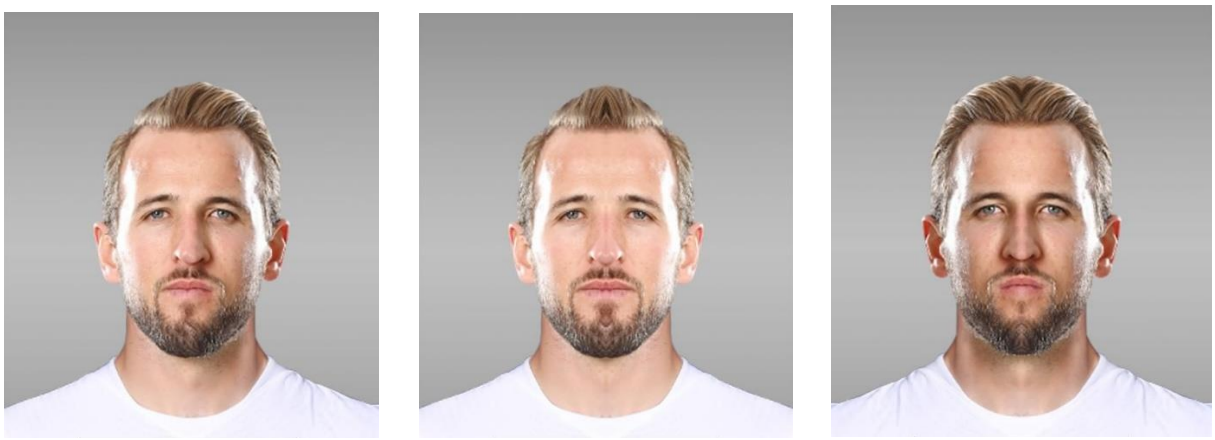
در این قسمت برای درست کردن موارد خواسته شده که شامل که عکس های r-l , l-r, r-r هستند (r-l همان قرینه l , r هستند و..) ابتدا عکس مورد نظر را به ماتریس تبدیل کرده و سپس ماتریس را نصف میکنیم. بر هر نصف ان از تابع پیاده سازی شده در قسمت قبل استفاده کرده تا بتوان قرینه آن را تولید کنیم و در ماتریس های جدید قسمت های قبلی را به جدید میچسبانیم.

همچنین برای تولید psnr و ssim از فرمول آن ها ساتفاده میکنیم که کد آن در فایل موربوط به قسمت اول موجود است.

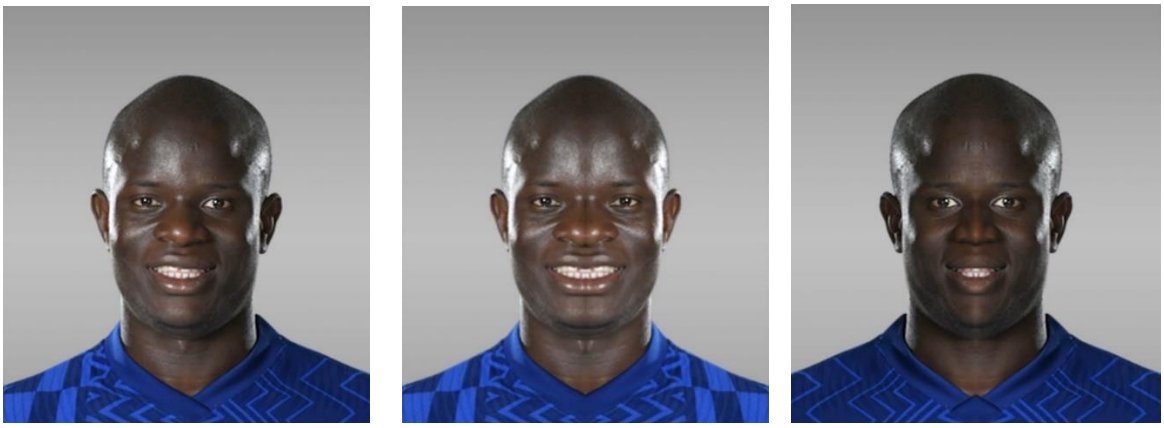
```
def PSNR(original, compress):
  mse = np.mean((original - compress) ** 2)
  if(mse == 0):
    return 100
  max_pixel = len(original)
  psnr= 20 * log10(max_pixel / sqrt(mse))
  return psnr
```

```
def ssim(img1, img2):
    x1 = (0.01 * 255)**2
    x2 = (0.03 * 255)**2
    img1 = img1.astype(np.float64)
    img2 = img2.astype(np.float64)
    kernel = cv2.getGaussianKernel(11, 1.5)
    field = np.outer(kernel, kernel.transpose())
    m1 = cv2.filter2D(img1, -1, field)[5:-5, 5:-5]  # valid
    m2 = cv2.filter2D(img2, -1, field)[5:-5, 5:-5]
    sq1 = m1**2
    sq2 = m2**2
    mu = m1 * m2
    sigma1 = cv2.filter2D(img1**2, -1, field)[5:-5, 5:-5] - sq1
    sigma2 = cv2.filter2D(img2**2, -1, field)[5:-5, 5:-5] - sq2
    sigma = cv2.filter2D(img1 * img2, -1, field)[5:-5, 5:-5] - mu
    ssim_map = ((2 * mu + x1) * (2 * sigma + x2)) / ((sq1 + sq2 + x1) *
                                                     (sigma1 + sigma2 + x2))
    return ssim_map.mean()


def calculate_ssim(img1, img2):
    ssims=[]
    if img1.ndim == 2:
        return ssim(img1, img2)
    elif img1.ndim == 3:
        if img1.shape[2] == 3:
            ssims = []
            for i in range(3):
                ssims.append(ssim(img1, img2))
            return np.array(ssims).mean()
        elif img1.shape[2] == 1:
            return ssim(np.squeeze(img1), np.squeeze(img2))
```

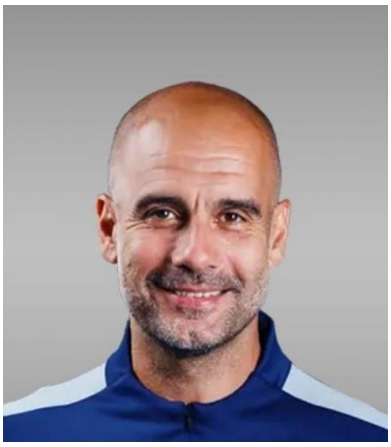
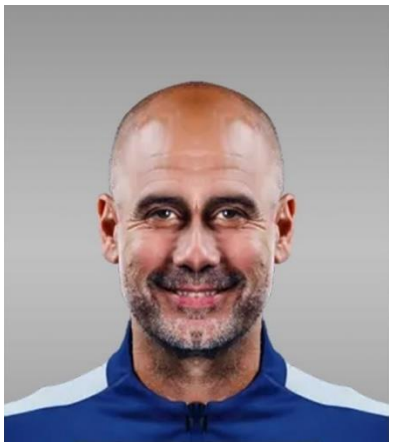
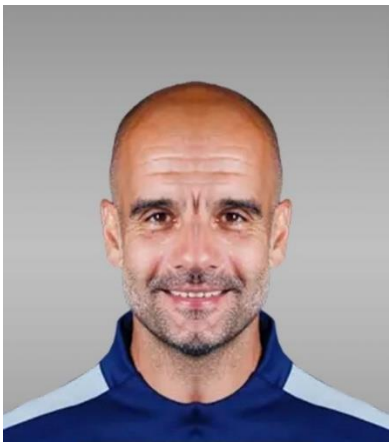
عکس ها به ترتیب : l_l - r_r - l_r میباشند



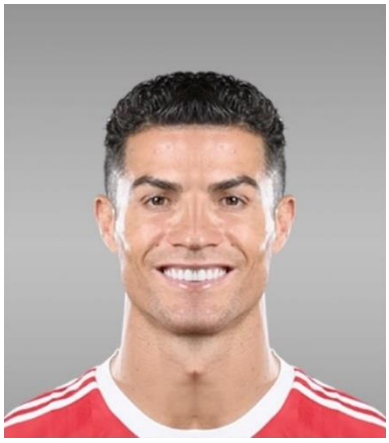
```
PSNR value for l_l picture is 43.049270571119 61 dB
PSNR value for r_r picture is 41.88622350145057 dB
PSNR value for r_l picture is 40.03873944414692 dB
-------------------------------------------
ssim value for l_l picture is 0.9049632795551982 dB
ssim value for r_r picture is 0.8809165807417432 dB
ssim value for r_l picture is 0.8070101854576847 dB
```



```
PSNR value for l_l picture is 42.032964376480564 dB
PSNR value for r_r picture is 41.162952950321795 dB
PSNR value for r_l picture is 39.04398906322247 dB
-------------------------------------------
ssim value for l_l picture is 0.8759983759186648 dB
ssim value for r_r picture is 0.859920409931406 dB
ssim value for r_l picture is 0.751655729001555 dB
```

```
PSNR value for l_l picture is 42.30827899081422 dB
PSNR value for r_r picture is 41.26855687598935 dB
PSNR value for r_l picture is 39.28906572867642 dB
-------------------------------------------
ssim value for l_l picture is 0.8817386150064825 dB
ssim value for r_r picture is 0.8618735758634463 dB
ssim value for r_l picture is 0.7598146574645055 dB
```


```
PSNR value for l_l picture is 42.06112727267036 dB
PSNR value for r_r picture is 41.07431771104635 dB
PSNR value for r_l picture is 39.04186988228274 dB
-------------------------------------------
ssim value for l_l picture is 0.857410841507587 dB
ssim value for r_r picture is 0.8364775697828698 dB
ssim value for r_l picture is 0.7111036400857037 dB
```

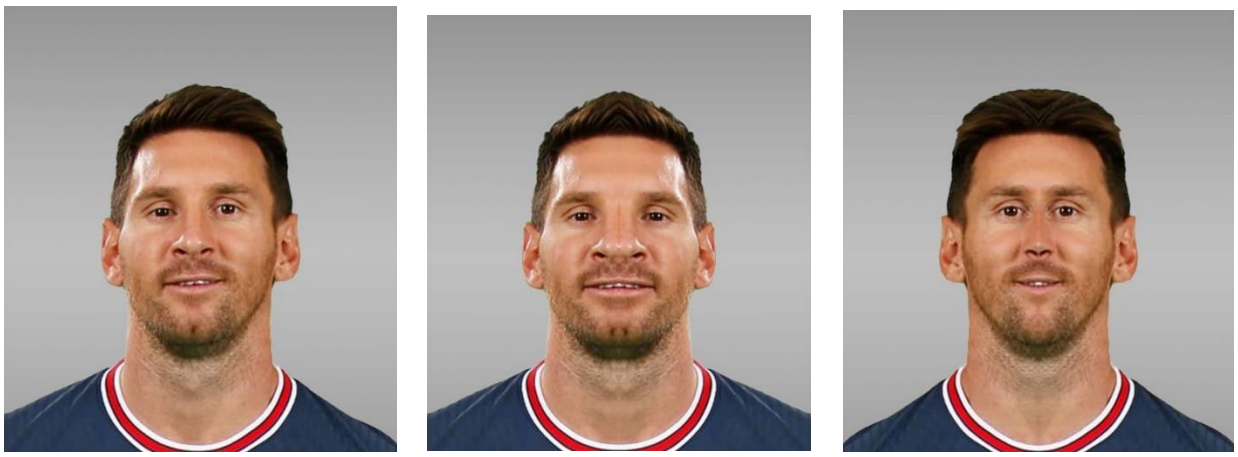


PSNR value for l_l picture is 42.57768362896482 dB
PSNR value for r_r picture is 41.142700213697516 dB
PSNR value for r_l picture is 39.540220725528215 dB
---
ssim value for l_l picture is 0.8519425581274173 dB
ssim value for r_r picture is 0.8232461185220766 dB
ssim value for r_l picture is 0.7015118703067372 dB

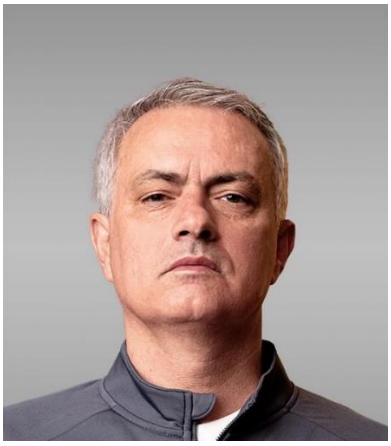
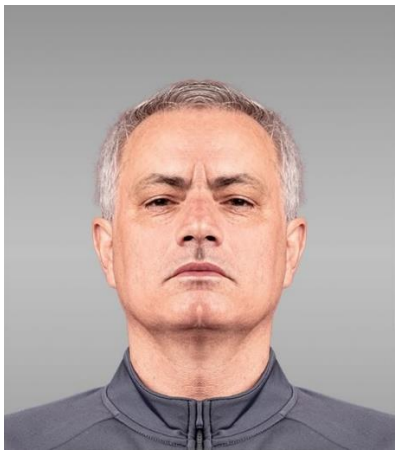
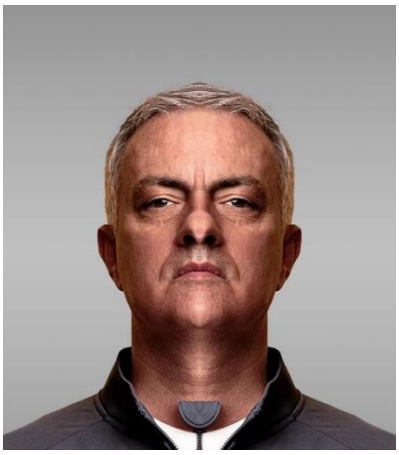


PSNR value for l_l picture is 42.26884663225348 dB
PSNR value for r_r picture is 41.437592093907575 dB
PSNR value for r_l picture is 39.24210853975278 dB
---
ssim value for l_l picture is 0.8840372655559907 dB
ssim value for r_r picture is 0.866892345897463 dB
ssim value for r_l picture is 0.7664969334643742 dB

PSNR value for l_l picture is 42.534596921670726 dB
PSNR value for r_r picture is 41.33948352964172 dB
PSNR value for r_l picture is 39.48813114101218 dB
---
ssim value for l_l picture is 0.8808279830165576 dB
ssim value for r_r picture is 0.8544674275602766 dB
ssim value for r_l picture is 0.7578252381815581 dB



PSNR value for l_l picture is 42.87693474177546 dB
PSNR value for r_r picture is 41.56701562826068 dB
PSNR value for r_l picture is 39.84148360496373 dB
---
ssim value for l_l picture is 0.8726081662620769 dB
ssim value for r_r picture is 0.8467821091553335 dB
ssim value for r_l picture is 0.7434450069381539 dB

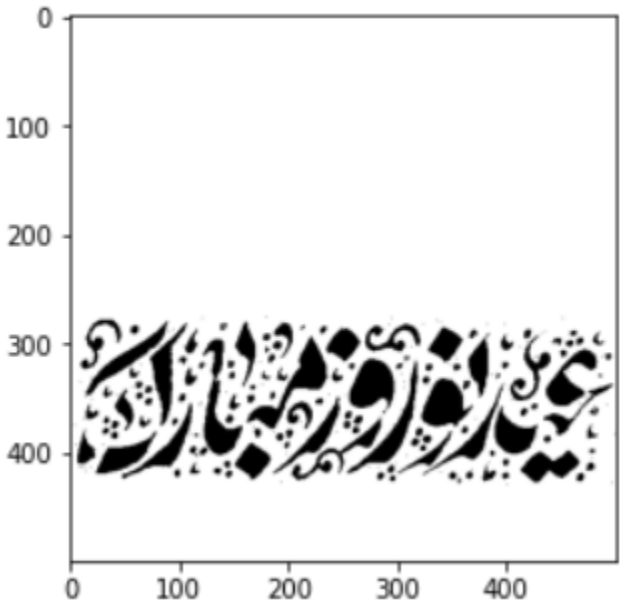




```
PSNR value for l_l picture is 42.066073658711O4 dB
PSNR value for r_r picture is 40.726239880912644 dB
PSNR value for r_l picture is 39.06134562274164 dB
-------------------------------------------
ssim value for l_l picture is 0.8347222071164513 dB
ssim value for r_r picture is 0.7872677587250018 dB
ssim value for r_l picture is 0.6695767470252901 dB
```

سوال 2

A

در این قسمت بعد از وارد کردن دو عکس، با یک حلقه، در یک ماتریس جدید سطر به سطر از اخر که دارای shape برابر با (500, 500) از عکس دو به آن اضافه کرده و حال آن را با استفاده از abs از عکس 1 کم میکنیم و این کار را تا اخر انجام میدهیم تا در نهایت عکس زیر از حاصل اختلاف آن ها ظاهر میشود.



B

در این قسمت ابتدا عکس msg را خوانده و با استفاده از thresh_binary آن را به باینری تبدل میکنیم تا سیاه و سفید شود.

سپس عکس اصلی را خوانده و داریم:

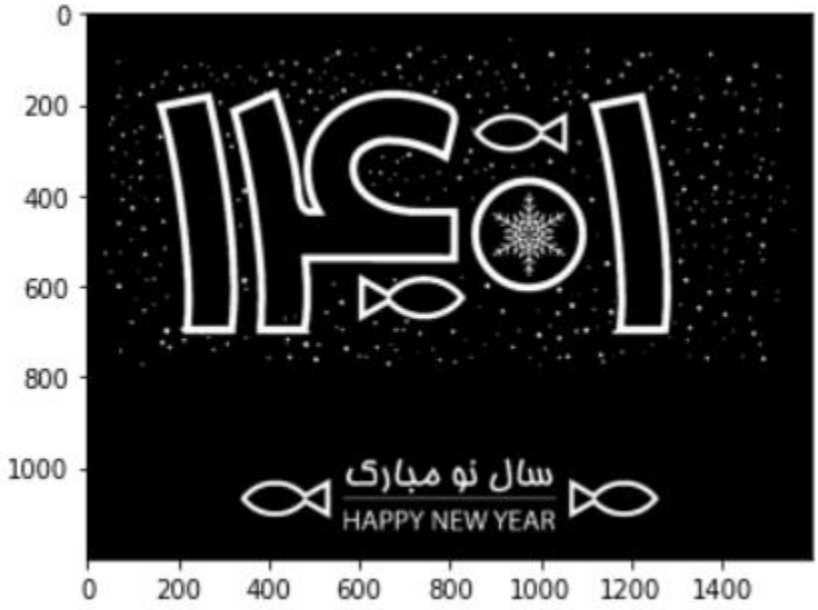
```
for i in range(img_mid.shape[0]):
    for j in range(img_mid.shape[1]):
        if bin[i,j]:
            if not (img_mid[i,j] % 2):
                if encode_img[i,j,0] <255:
                    encode_img[i,j,0] +=1
                else:
                    encode_img[i,j,0] -= 1
        elif (img_mid[i,j] % 2):
            if encode_img[i,j,0] < 255:
                encode_img[i,j,0] += 1
            else:
                encode_img[i,j,0] -= 1
```

C

در این قسمت برای decode کردن پیام عکس اصلی،

```
img_mid = np.absolute(main_img[:,:,0] - encode_img[:,:,2])
```

و سپس در یک آرایه متناظر جدید برای هر یک مقادیر، اگر مقدار value متناظر ان فرد بود برابر 1 و اگر زوج بود برابر 0 قرار داده با استفاده از plotو cmap= gary آن را رسم میکنیم.



سوال 3

A

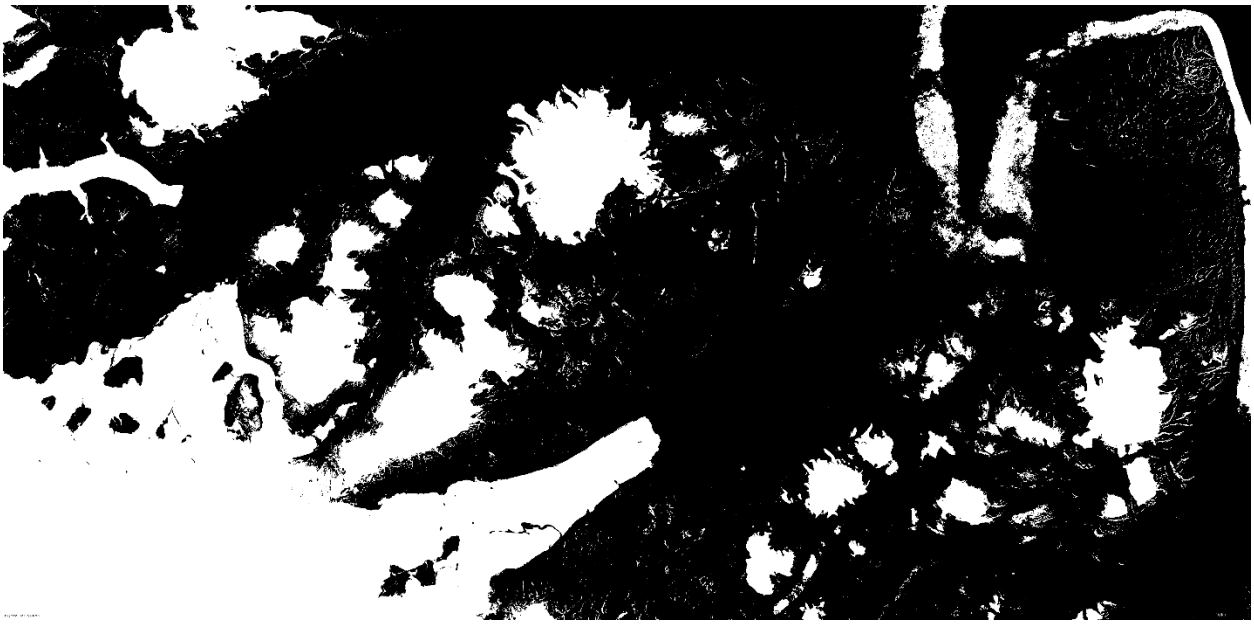
در این قسمت ابتدا عکس ها خوانده و با استفاده از یک تابع تبدیل گر که عکس را به gray تبدل کرده و داریم:

```
def image_show (image_name, rgb):
  col = Image.open(image_name)
  gray = col.convert('L')

  bw = np.asarray(gray).copy()
  bw[bw < rgb] = 0    # Black
  bw[bw >= rgb] = 255 # White

  imfile = Image.fromarray(bw)
  return imfile , bw
```

در این تبدیل برای اینکه بتوان بهترین نتیجه را از تبدیل عکس ها به سیاه و سفید برد، من مقدار rgb در نظر گرفته شده با سعی و خطا پیدا کردم که آن را برای بخش اول برابر 190 قرار دادم و عکس های زیر را تولید کردم.

عکس 2000:

عکس 2020:



سپس برای مقدار یخ آب شده در 20 سال ماتریس 2020 را از 2000 کم کرده و سپس درصد گیری را انجام که نتیجه زیر حاصل شد:

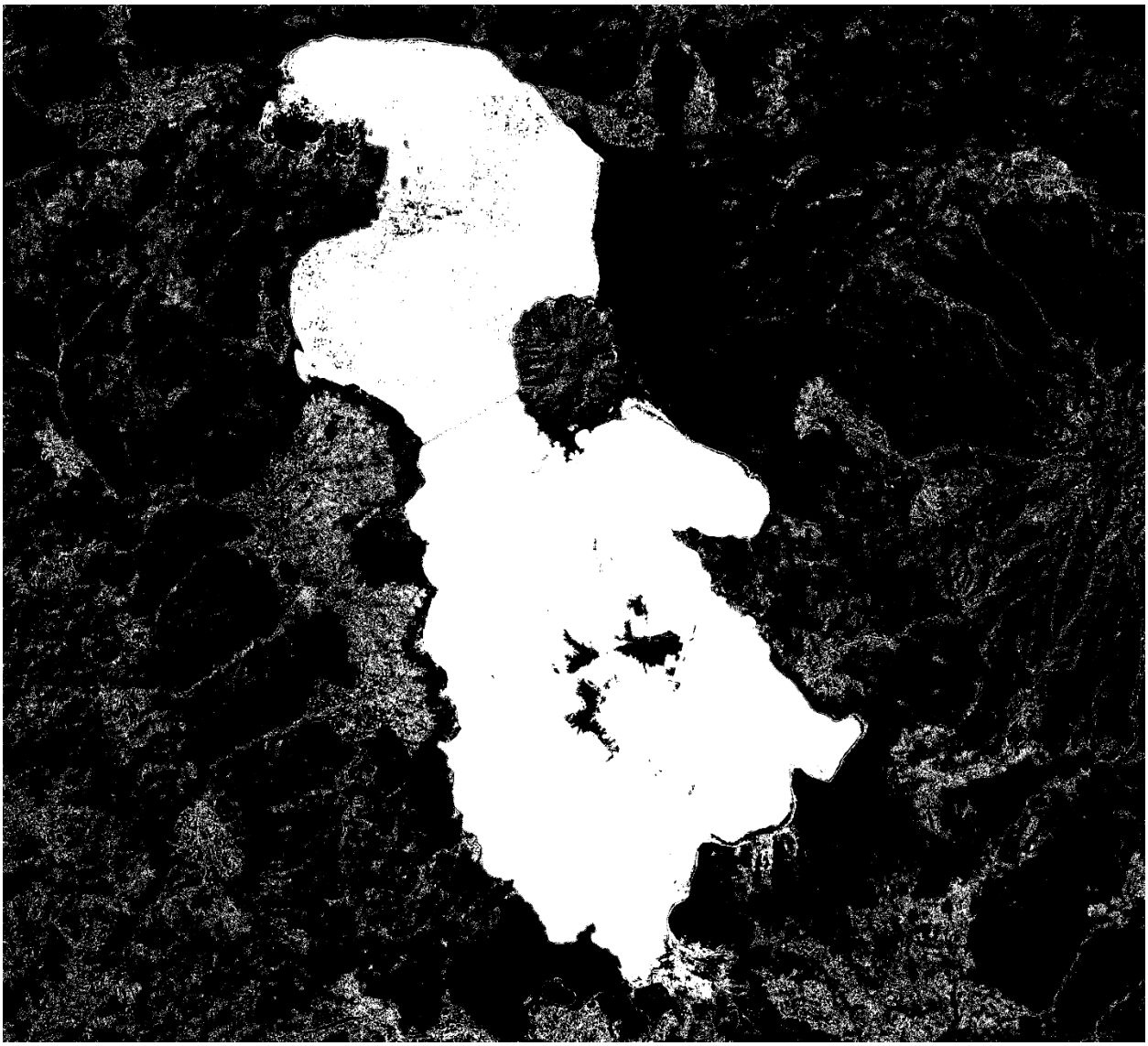
```
61.11157697993615  percent of ice is melted in 20 years
```

B

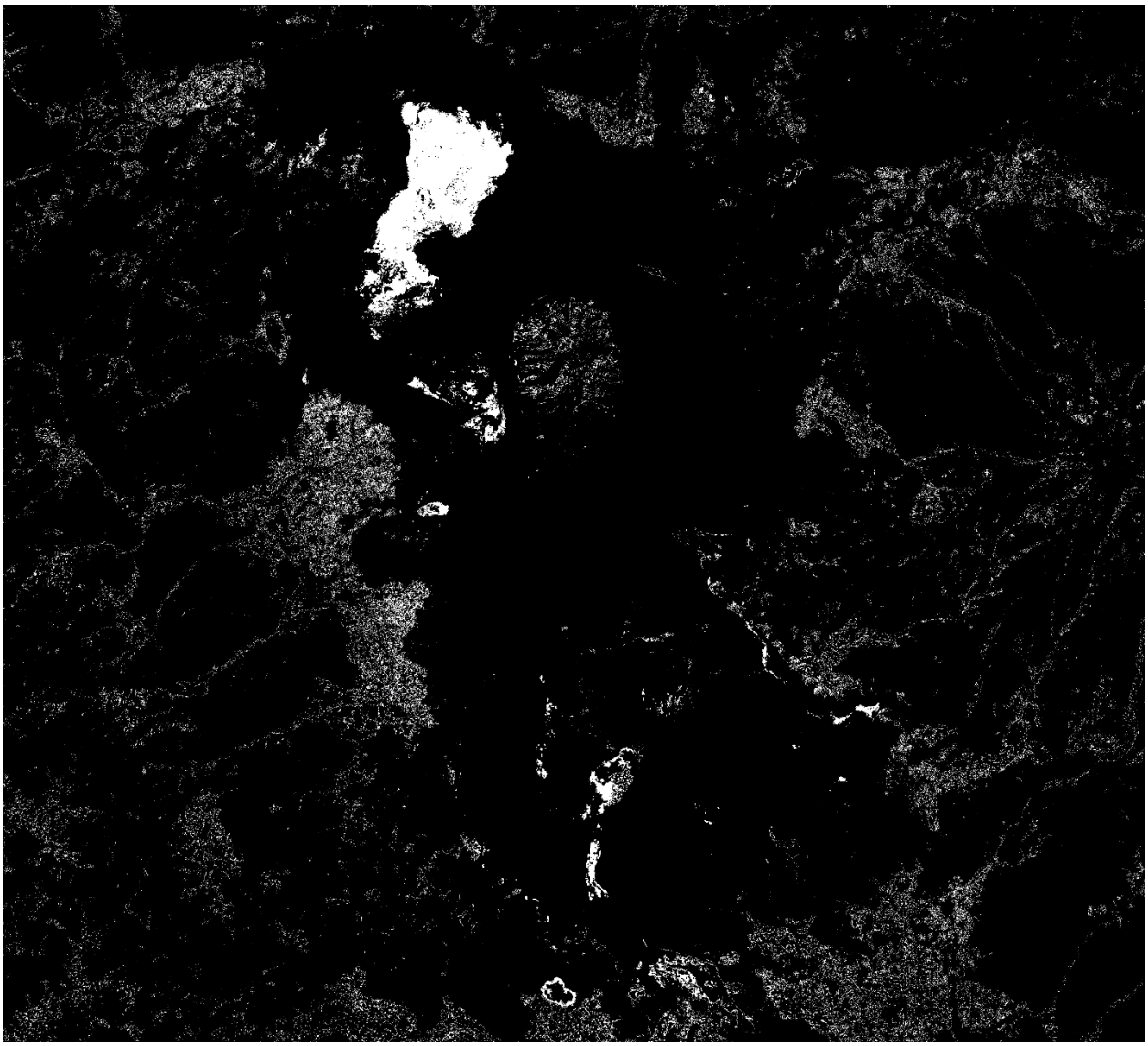
در این قسمت برای اینکه بتوان عکس ها را به سیاه و سفید تبدیل کرد از رنج گیری استفاده کردم ولی متاسفانه نتایج حاصل از آن با استفاده از سعی و خطای زیاد به بهترین شکل ممکن حاصل نشد.

در این قسمت با رنج های مختلف برای rgb برای تشخیص دریاچه، مقداری از اطراف دریاچه نیز با توجه به شدت تشبیه رنگ آن با رنگ دریاچه به سفید تبدیل میشد که برای مینیمایز کردن این خطا تنها مقداری از عکس که دریاچه در آن وجود داشت را در نظر گرفتم که بهترین نتایج حاصل شده به شکل زیر میباشد:

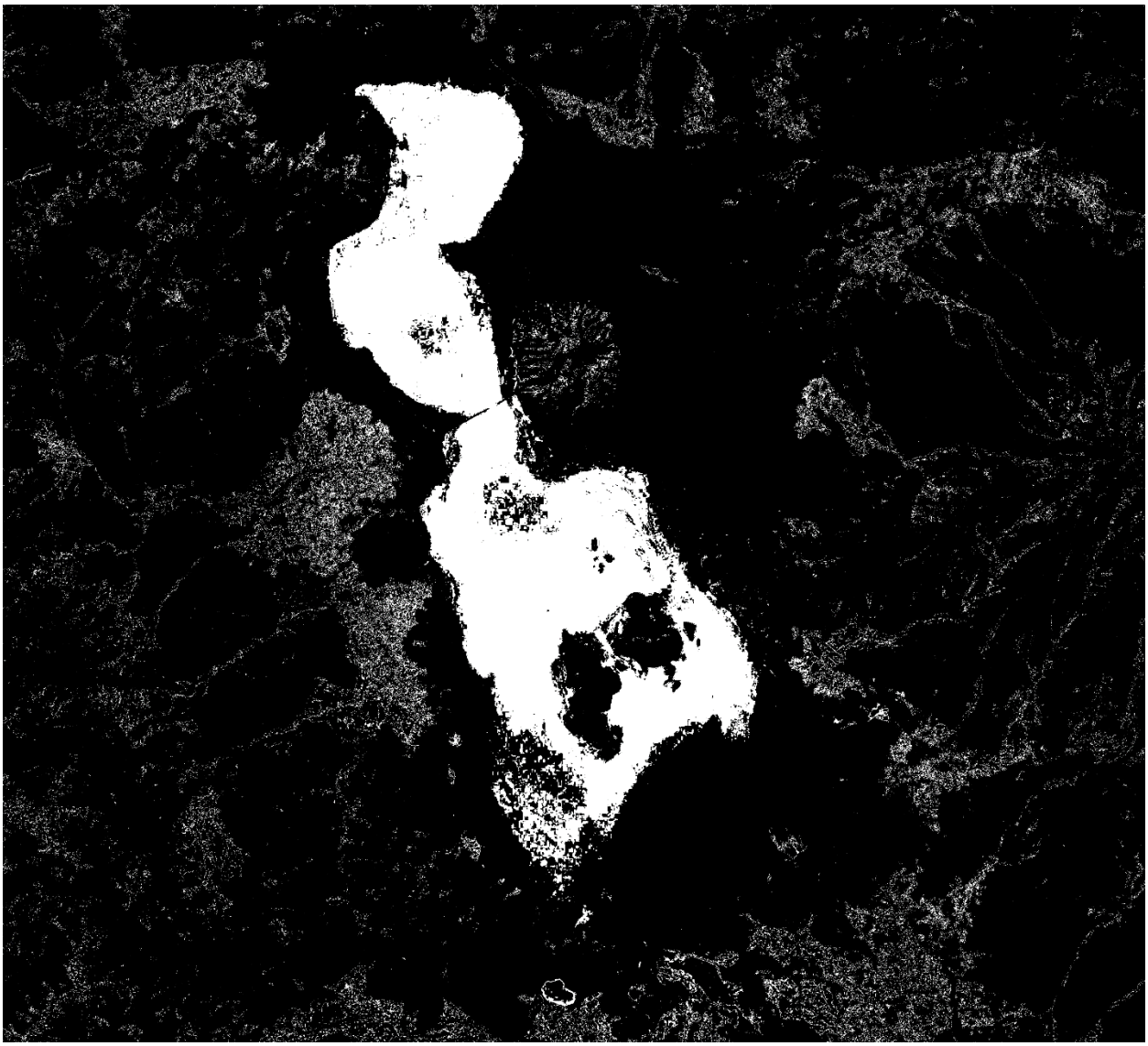
دریچه در 2000:



دریاچه در 2018:



دریاچه در 2020:



سپس با استفاده از روش بخش قبل، برای بدست اوردن پیکسل های متفاوت (پیدا کردن مقدار آب خشک شده)، ماتریس عکس 2018 را از 2000 کم کرده و سپس با استفاده از scale و volume داده شده در سوال حجم آب کم شده در هر کیلومتر مربع حساب کردم و نتایج زیر حاصل گردید:

```
reduced volume of water between 2000 to 2018 is :  13.364618442938863 for km3
```

C

در این قسمت نیز تمام بخش های قسمت قبل را برای عکس های 2018 و 2020 انجام دادم و که درصد آب اضافه شده به نسبت به کل اب موجود سال 2000 به صورت زیر است:

```
10.714285714285717  percent of the lake has been restored since 2018 to 2020
```

D

در این قسمت نیز با استفاده از بخش اول و rgb برابر 190 عکس های موجود را به سیاه و سفید تبدیل کرده تا یخ مشخص شود:

عکس 2003:

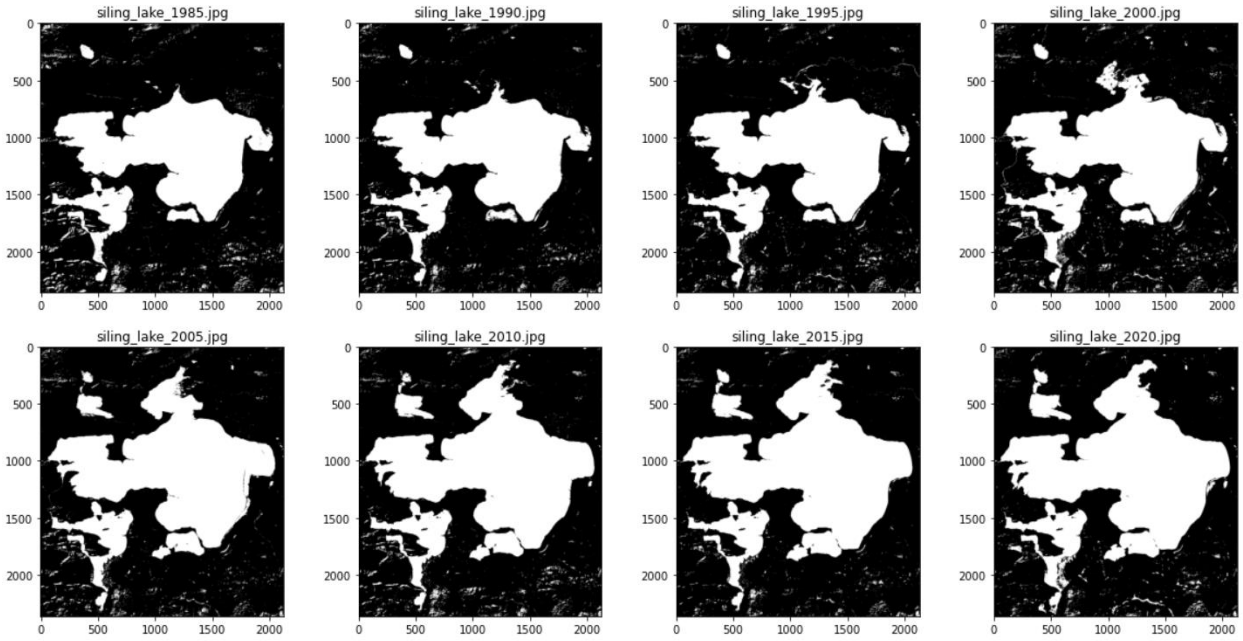
عکس 2017:

همچنین با استفاده از تابع موجود در قسمت اول نتایج زیر حاصل شد:

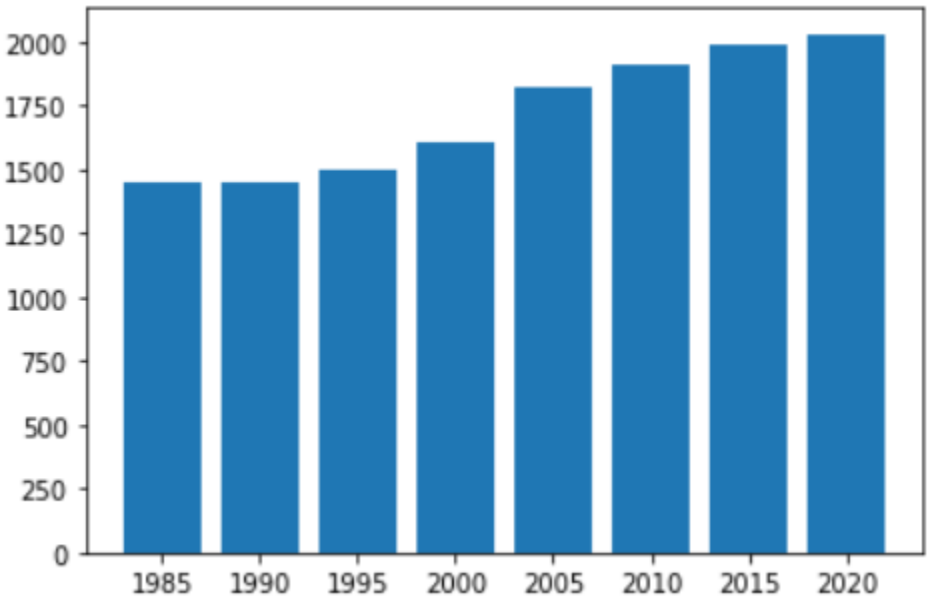
```
74.91318252270989  percent of ice is melted in 14 years
```

E

در این قسمت ابتدا عکس ها را خواندم و و با استفاده از تابع رنج گیری (inRange) و thres_OTSU عکس ها را به سیاه و سفید تبدیل میکنم و سپس در هر عکس تعداد پیکسل های سفید را شمرده و با استفاده از تابعی که در بخش 2 پیاده سازی کردم حجم آب را در هر عکس حساب کرده و سپس عکس ها با استفاده از plt نمایش میدهم:



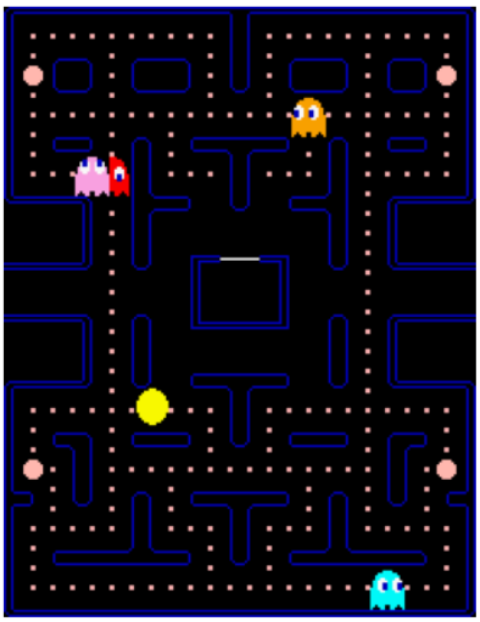
و در اخر نیز با استفاده از plot جدول حجم آب در سال های مختلف به شکل زیر میباشد:



سوال 4

A

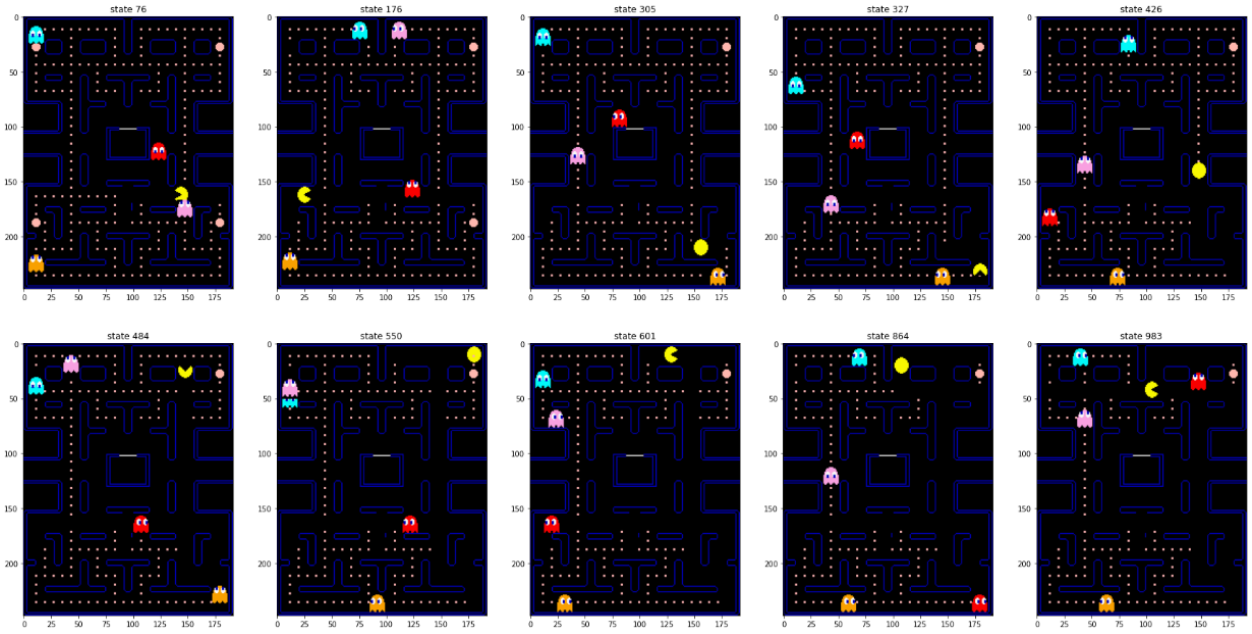
در این قسمت بعد از وارد کردن تمام آیکون ها، و خود Maze، margin های اطراف maze را مشخص کرده و نقطه ها و انرژی ها را جایگزاری میکنیم. بعد از جایگزاری نقاط، برای جایگذاری پک من و هیولا ها ، آیکون ها را گرفته و به صورت رندوم در بازه های نقاط جایگذاری میکنیم.



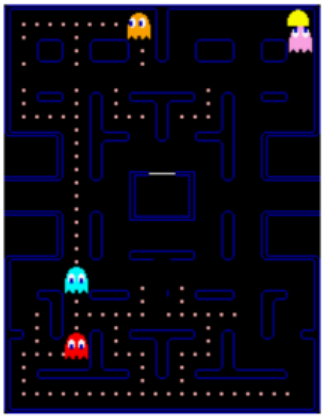
B

در این قسمت بنا به توضیحات مربوطه که در جدول در سوال آمده پیاده سازی انجام شده و حاصل به شکل زیر است.

10 موقعیت در بازی:



موقعیت نهایی بازی (فایل پک من 1 در فولدر)



C

این قسمت در فایل گزارش موجود است.

سوال 5

A)

تشکیل تصویر - عناصر نوری چشم، تصویری از یک جسم را روی یک "فیلم" حساس به نور - شبکیه چشم - متمرکز می کنند و در عین حال از مقدار صحیح نور برای ایجاد "نور قرار گرفتن" مناسب اطمینان می دهند. سه فرآیند تشکیل تصویر 1. شکست یا خمش نور توسط عدسی و قرنیه. 2. محل اقامت، تغییر در شکل لنز. 3. انقباض یا باریک شدن مردمک. انکسار - خم شدن پرتوهای نور. - خمش پرتوهای نور در محل اتصال دو ماده شفاف با چگالی متفاوت. - با ورود پرتوهای نور به چشم، در سطوح

قدامی و خلفی قرنیه شکسته می شود. هر دو سطح عدسی چشم، پرتوهای نور را بیشتر می‌شکنند تا به طور دقیق روی شبکیه متمرکز شوند.

دید دوچشمی - در انسان، هر دو چشم به جای تمرکز بر روی یک مجموعه سمت راست و یک مجموعه چپ، تنها بر روی یک مجموعه از اشیاء (دید بیونولی) تمرکز می کنند. - هنگامی رخ می دهد که پرتوهای نور از یک جسم به نقاط مربوطه به دو شبکیه برخورد کند

همچنین در مواقعی که تاری دید وجود دارد با کوچک کردن مردمک چشم سعی در بالا بردن وضوح تصویر میکنیم.

فوکوس خودکار بر روی اجسام دور یا نزدیک

در شب، با گذشت زمان، دید در تاریکی بهتر میشود

و..

B

رزولوشن: دوربین سامسونگ دارای رزولوشن 5800 در 8700 است در حالی که آیفون 3000 در 4000

حجم عکس: به واسطه رزولوشن بالا و f برابر 1.8 عکس های دوربین سامسونگ از کیفیت بالا تری برخوردار است.

کیفیت: بر روی کاغذ و تئوری باید کیفیت عکس های سامسونگ بسیار بالا تر از آیفون باشد ولی به واسطه هوش مصنوعی و پردازش تصویر مشاهده میکنیم که معمولا عکس های گرفته شده با دوربین ایفون کیفیت بهتری دارند.(تجربه شخصی من و تحلیل های افراد حرفه ای در یوتیوب هست)

C

در هر عکس به واسطه اینکه هر پیکس 3 مقدار RGB دارد میتواند مقادیر مختلفی را شامل شود، بعد از تبدیل به gray ان مقادیر تغییر میکند و نسبت موجود در هر پیکسل از بین میرود و برای انجام عکس آن راهی وجود ندارد که این ترکیبات را به همان اندازه اولیه بفهمیم و با تبدیل عکس نامناسبی تحویل میگیریم.

D

Sub_sampling

از آن برای کاهش ابعاد و حجم استفاده میشود. به این صورت است که در هر یک سری مشخص از داده، مقداری از آن را انتخاب میکنیم و سپس تصویر را میسازیم.

Resampling

از آن برای دستکاری در جهت، وضوح، شارپ بودن، چرخش و... استفاده میشود که در نتیجه برای دستکاری تصویر مورد ساتفاده قرار میگیرد.

E

بله، در color space های مختبف اطلاعات مختلفی را میتوان دریافت کرد. البته بستگی به نوع مقایسه دارد. در صورتی که مقایسه نویز و ... باشد میتواند دو عکس را به color space ساده تری مانند gray برد و انجا مقایسه کرد ولی در صورتی که مقایسه پیکسل به پیکسل باشد که رنگ را باهم مقایسه کنیم باید اینکار در rgb رخ دهد چرا که ممکن است با تبدیل آن به color space دیگر، مقداری از اطلاعات از دست برود.