



دانشگاه صنعتی امیر کبیر

(پلی تکنیک تهران)

نام و نام خانوادگی:

پیمان هاشمی

شماره دانشجویی:

400131032

شماره تمرین:

تمرین شماره 2

درس:

تصویر پردازش رقمی

سوال 1)

(A

2	1	2	1	2	5	6	6
3	3	2	3	2	4	5	5
2	2	4	2	1	4	5	5
3	2	4	3	4	2	4	4
3	3	5	4	1	1	2	4
2	3	5	1	1	1	1	4
2	2	3	3	2	3	3	4
5	5	4	4	5	6	5	6

(B

تصویر :LSB bit plane

1	0	1	0	1	0	1	1
0	0	1	0	1	1	0	0
1	1	1	1	0	1	0	0
0	1	1	0	1	1	1	1
0	0	0	1	0	0	1	1
1	0	0	0	0	0	0	1
1	1	0	0	1	0	0	1
1	0	1	1	0	1	0	1

تصویر مربوط به بیت وسط

1	1	0	1	0	1	0	0
0	0	0	0	0	1	1	1
0	0	1	0	1	1	1	1
0	0	1	0	1	1	1	1
0	0	1	1	1	1	0	1
0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	1
1	1	1	1	1	0	1	0

تصویر bit_plane مربوط به MSB

0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	0	1	1	0	0	0
1	1	0	1	0	1	0	0
1	1	0	0	1	1	1	0
1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	0
0	0	0	0	0	0	1	0

(C)



(D)

تصویر باینری که از thresholding بدست آمده است:

0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	0	1	1	0	0	0
1	1	0	1	0	1	0	0
1	1	0	0	1	1	1	0
1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	0

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

(E)

تصاویر بیت ها زمانی که از 0 تا 7 بیت به 0 تا 255 بیت تبدیل کرده ایم:

160	192	160	192	160	64	32	32
128	128	160	128	160	96	64	64
160	160	96	160	192	96	64	64
128	160	96	128	96	160	96	96
128	128	64	96	192	192	160	96
160	128	64	192	192	192	192	96
160	160	128	128	160	128	128	96
64	64	96	96	64	32	64	32

سوال 2)

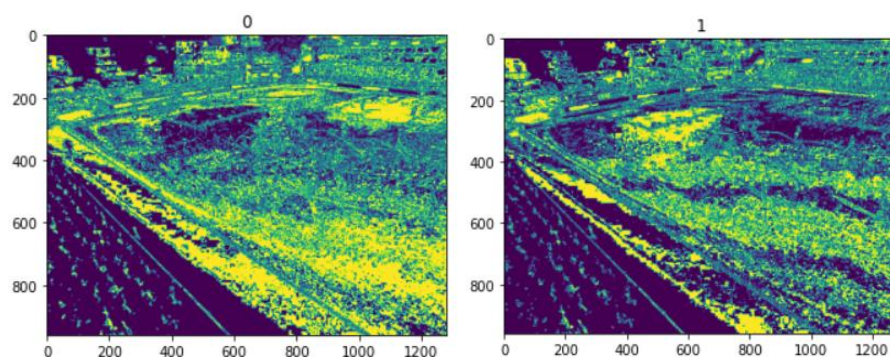
برای هر دو فریم تصویر قسمت های A, B, C باهم آورده شده است.

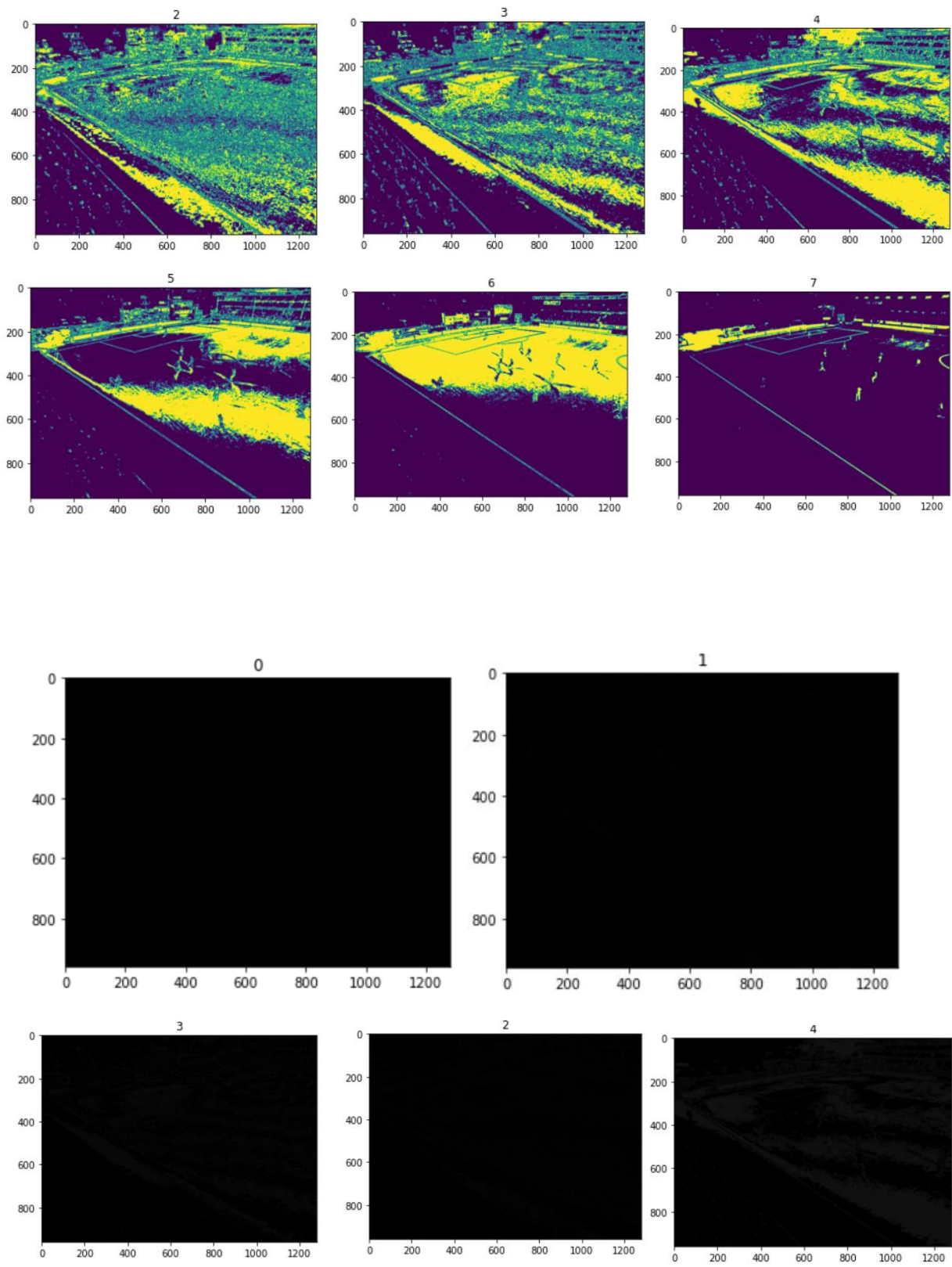
در ابتدا عکس ها را خوانده و به grayscale میبریم سپس برای هر پیکسل یک آرایه به طول 8 در نظر گرفته و آن را به باینری تبدیل میکنیم. سپس 8 بیت شامل (1,2,4,8,16,32,64,128) را به ترتیب نمایش میدهیم

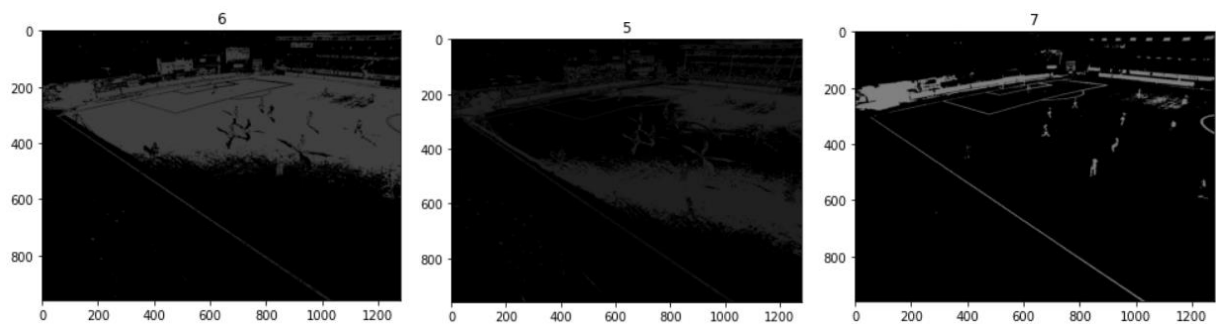
ما لیستی از رشته ها داریم که هر رشته مقدار پیکسل باینری را نشان می دهد. برای استخراج صفحات بیت باید روی رشته ها تکرار کنیم و کاراکترهای مربوط به صفحات بیت را در لیست ها ذخیره کنیم.

برای بازسازی تصاویر از 4 بیت بالا تر نیز، بعد از بدست آوردن XOR تصاویر، تصاویر مربوط به 4 فریم بالا تر را باهم جمع زده و نشان میدهیم.

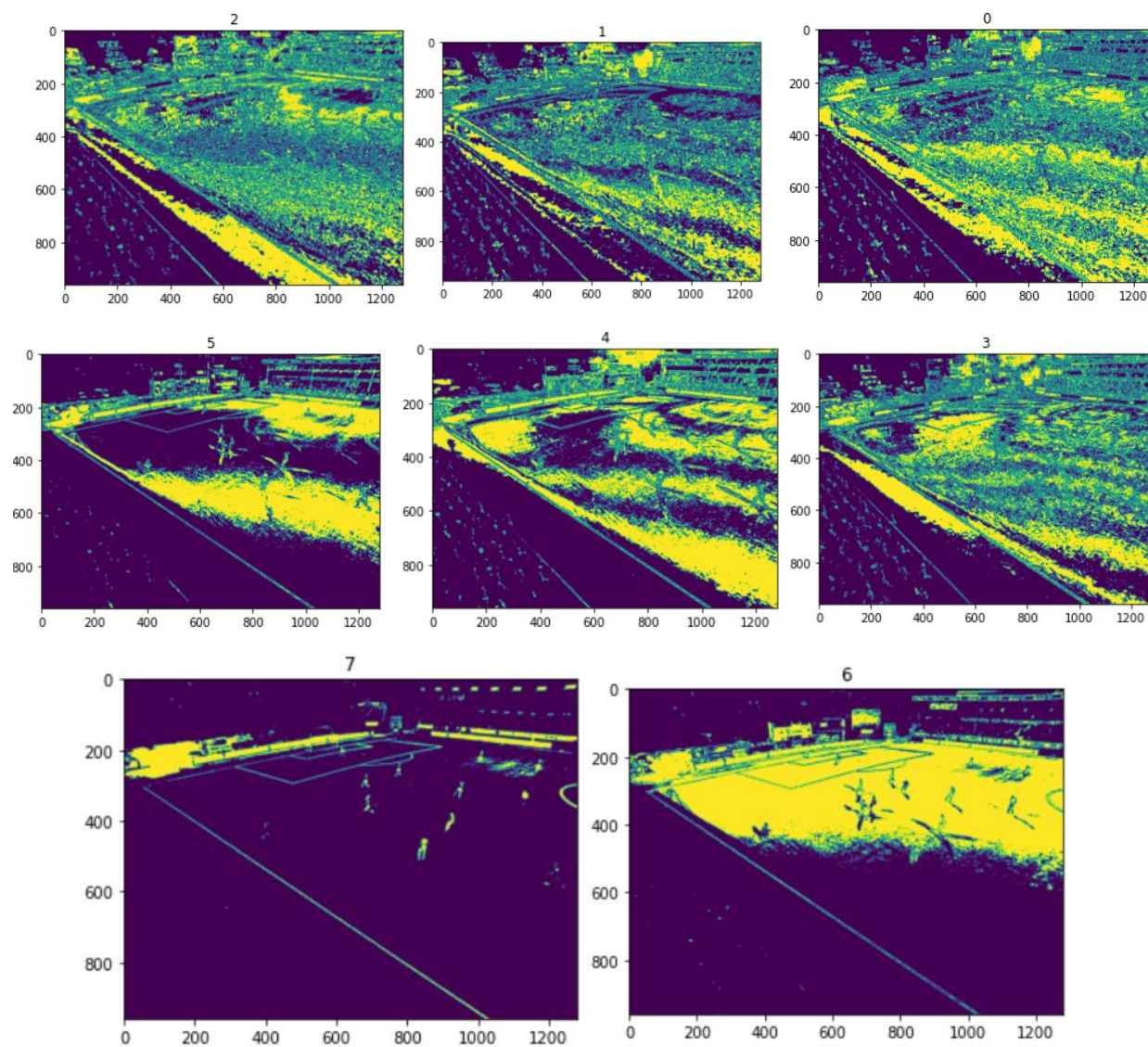
فریم 1 تصویر 1:



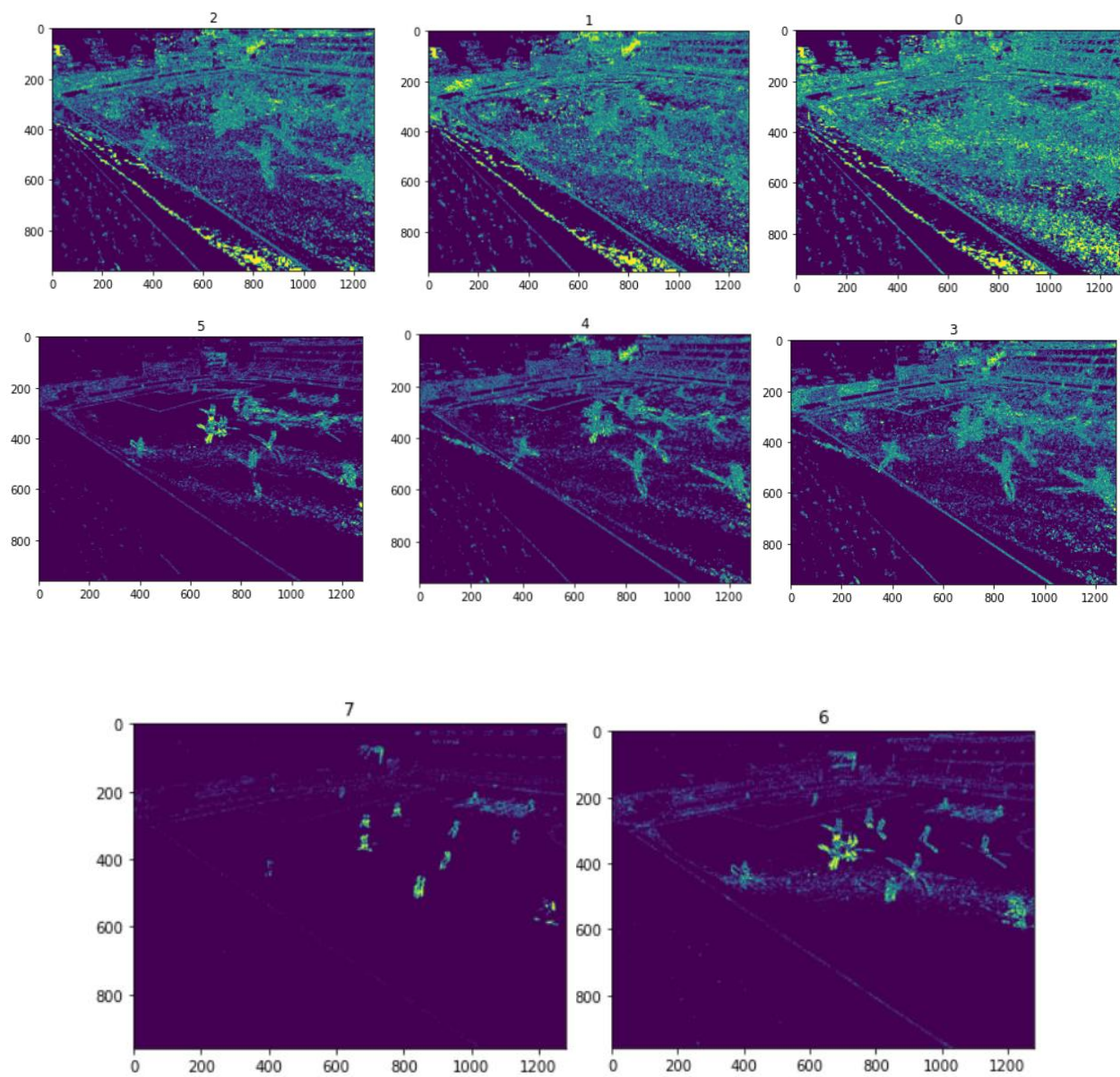




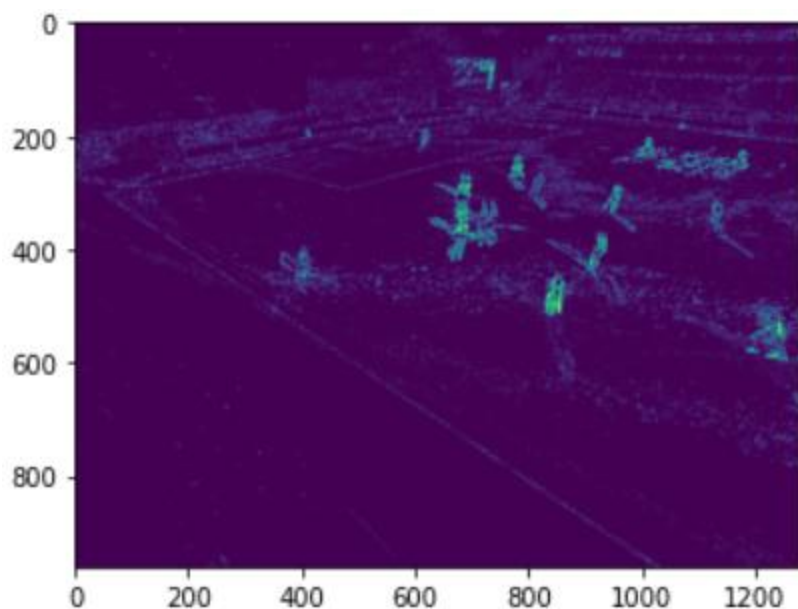
فریم 1 تصویر 2:



Xor فریم 1:

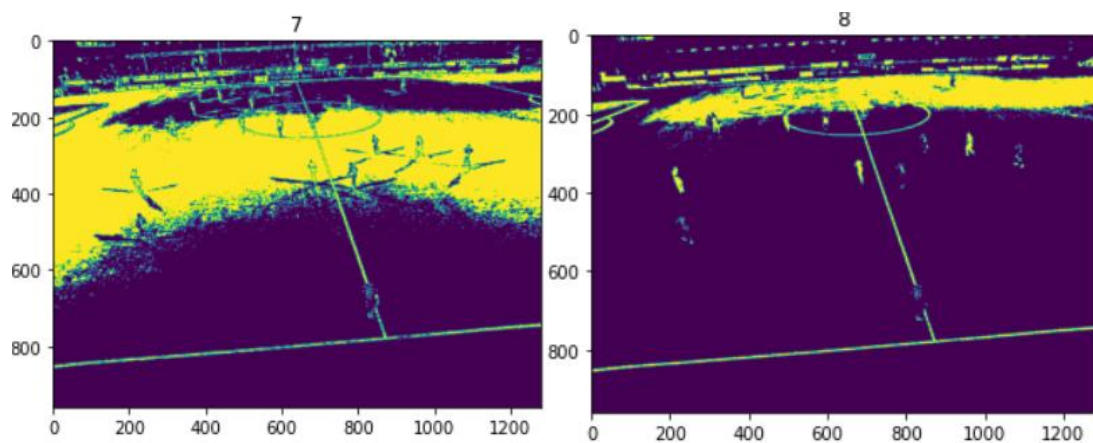


بازسازی تصویر با xor مربوط 4 بیت بالا تر:

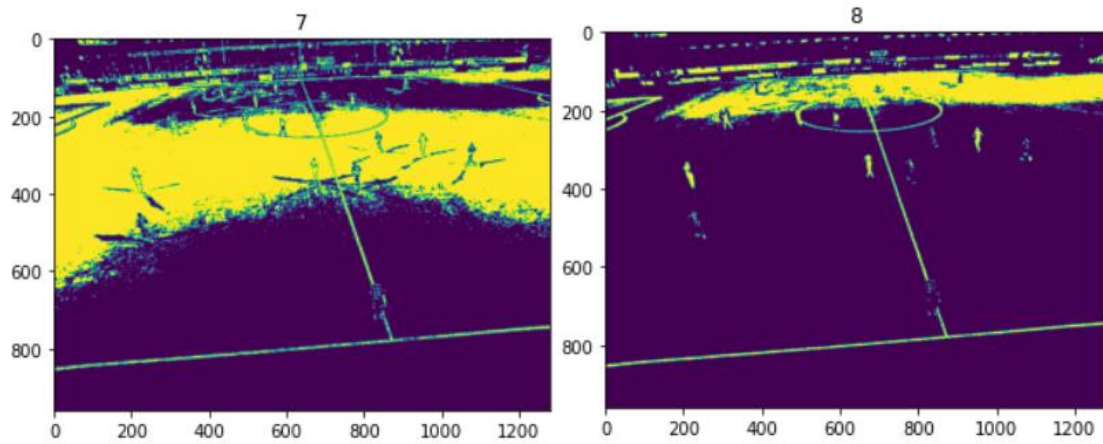


(توجه: به دلیل تعداد زیاد تصاویر در این گزارش تنها تصاویر بیت 8 و 7 ام را قرار میدهم ولی در فایل کد تمام تصاویر به صورت کامل آورده شده است)

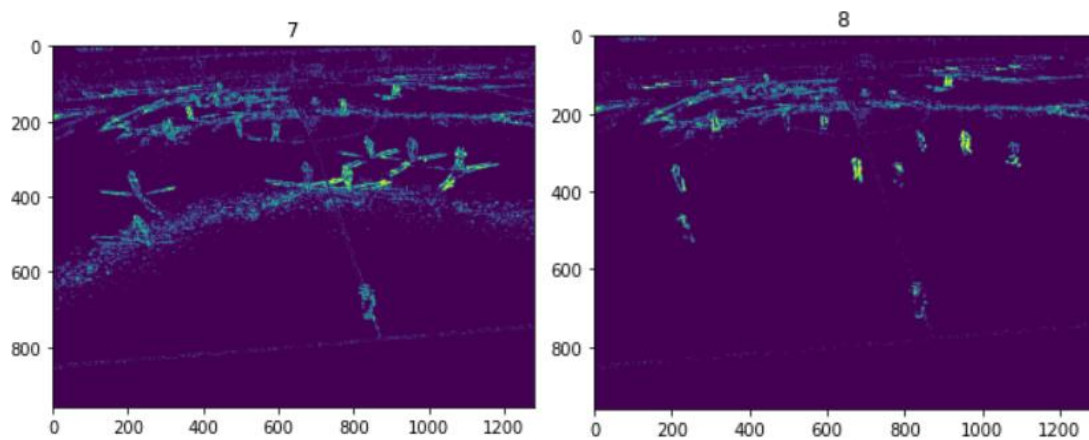
فریم 2 تصویر 1:



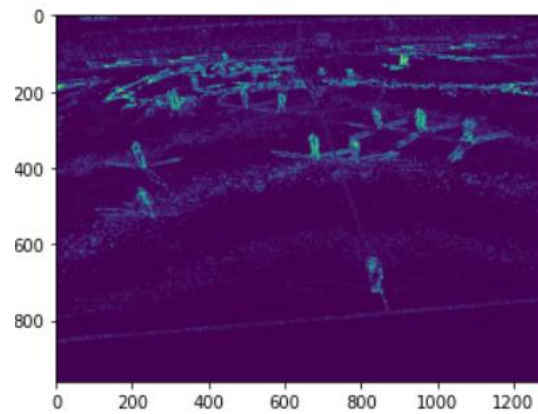
فریم 2 تصویر 2:



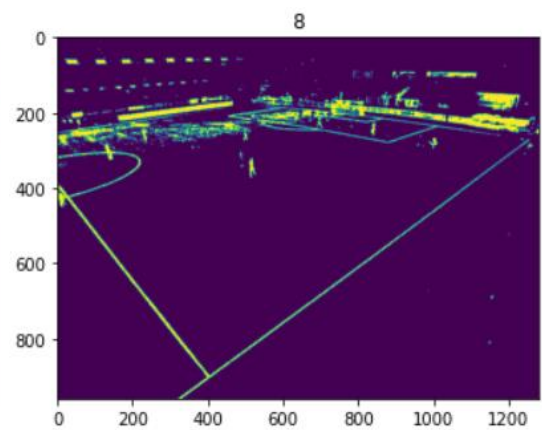
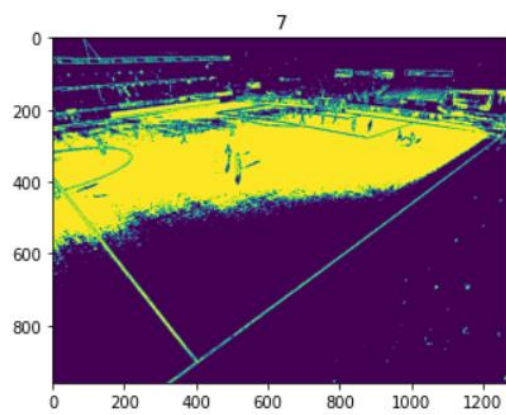
Xor فریم 2:



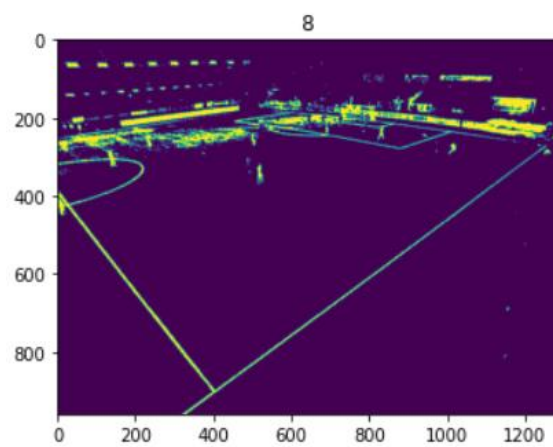
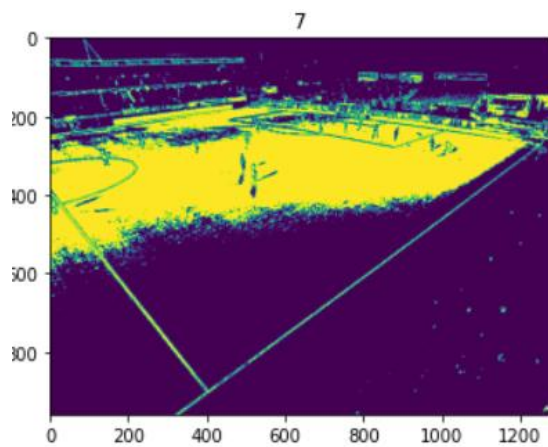
تصویر بازسازی شده:



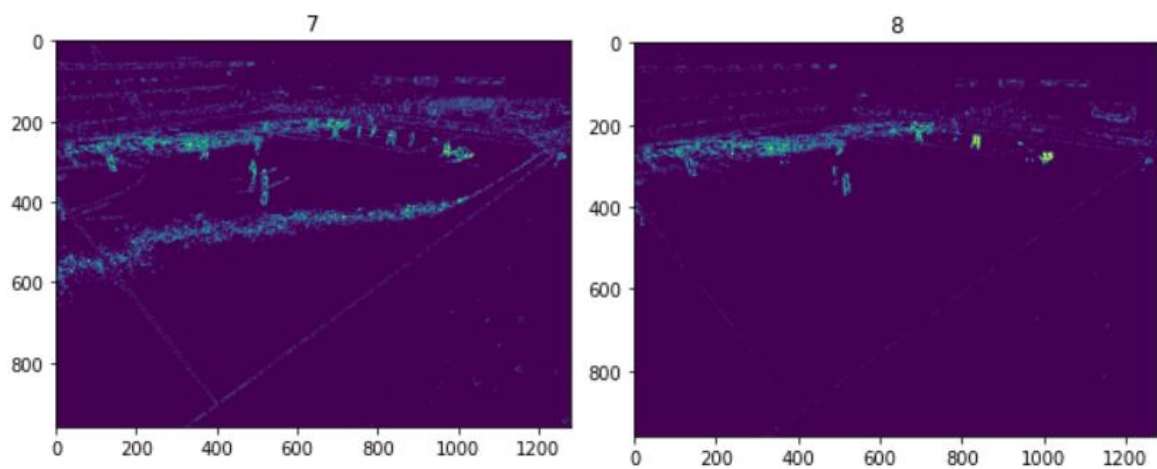
فریم 3 تصویر 1:



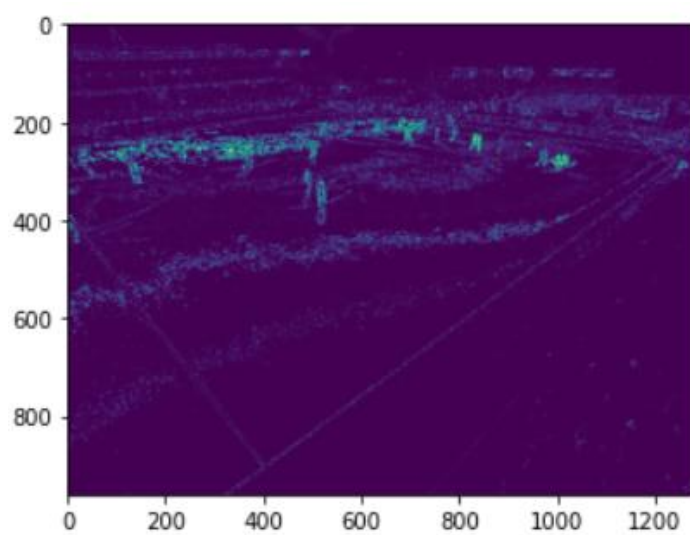
فریم 3 تصویر 2:



Xor فریم 3:



تصویر بازسازی شده:



(D) تصویر 1

از thresholding با بازه 120-255



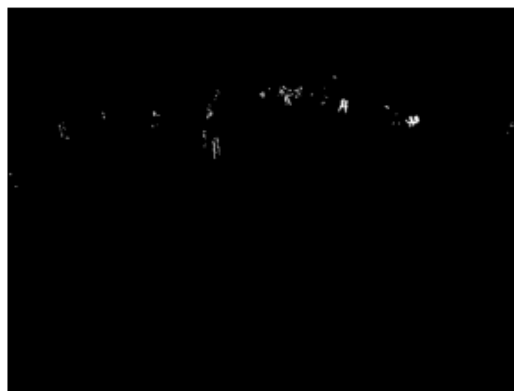
تصویر 2:

بازه 120 تا 230



تصویر 3:

بازه 115 تا 230



سوال 3

(A)

فریم ها باهم جمع زده و با استفاده از `np.floor_divide` میانگین آن ها محاسبه میکنیم.

AVG of 2 frames



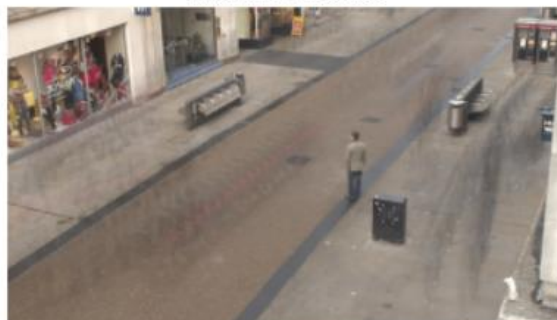
AVG of 5 frames



AVG of 10 frames



AVG of 20 frames



(B)

حاصل تفریق فریم های تست با عکس های مرحله قبل برابر زیر است:

Subtract of AVG_2 and pedestrians_test_01



Subtract of AVG_2 and pedestrians_test_02



Subtract of AVG_5 and pedestrians_test_01



Subtract of AVG_5 and pedestrians_test_02



Subtract of AVG_10 and pedestrians_test_01



Subtract of AVG_10 and pedestrians_test_02



Subtract of AVG_20 and pedestrians_test_01



Subtract of AVG_20 and pedestrians_test_02



(C)

رنج مناسب در بازه 25 تا 250 میباشد و عکس های حاصل طبق سوال و بعد از اعمال فیلتر median با کرنل 5 به صورت زیر هستند:

AVG - 2 frames - pedestrians_test_01



AVG - 2 frames - pedestrians_test_02



AVG - 5 frames - pedestrians_test_01



AVG - 5 frames - pedestrians_test_02



AVG - 10 frames - pedestrians_test_01



AVG - 10 frames - pedestrians_test_02



AVG - 20 frames - pedestrians_test_01



AVG - 20 frames - pedestrians_test_02



(D)

AVG - 2 frames - pedestrians_test_01



AVG - 2 frames - pedestrians_test_02



AVG - 5 frames - pedestrians_test_01



AVG - 5 frames - pedestrians_test_02



AVG - 10 frames - pedestrians_test_01



AVG - 10 frames - pedestrians_test_02



AVG - 20 frames - pedestrians_test_01



AVG - 20 frames - pedestrians_test_02



(E) قسمت 1:

Median of 2 frames



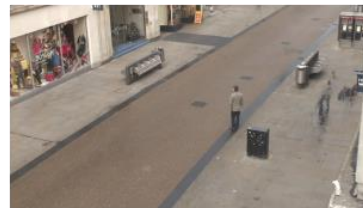
Median of 5 frames



Median of 10 frames



Median of 20 frames



قسمت دو:

Subtract of Median_2 and pedestrians_test_01



Subtract of Median_2 and pedestrians_test_02



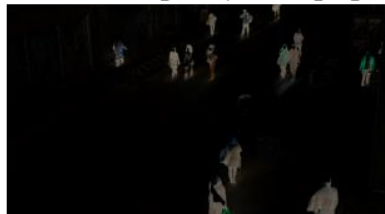
Subtract of Median_5 and pedestrians_test_01



Subtract of Median_5 and pedestrians_test_02



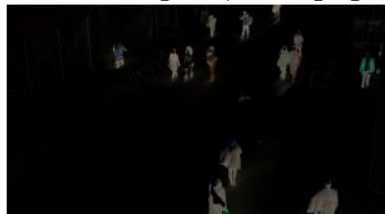
Subtract of Median_10 and pedestrians_test_01



Subtract of Median_10 and pedestrians_test_02



Subtract of Median_20 and pedestrians_test_01



Subtract of Median_20 and pedestrians_test_02



Median - 2 frames - pedestrians_test_01



Median - 2 frames - pedestrians_test_02



Median - 5 frames - pedestrians_test_01



Median - 5 frames - pedestrians_test_02



Median - 10 frames - pedestrians_test_01



Median - 10 frames - pedestrians_test_02



Median - 20 frames - pedestrians_test_01



Median - 20 frames - pedestrians_test_02



Median - 20 frames - pedestrians_test_02



Median - 20 frames - pedestrians_test_02



Median - 20 frames - pedestrians_test_02



Median - 20 frames - pedestrians_test_02



Median - 20 frames - pedestrians_test_02



Median - 20 frames - pedestrians_test_02



Median - 20 frames - pedestrians_test_02



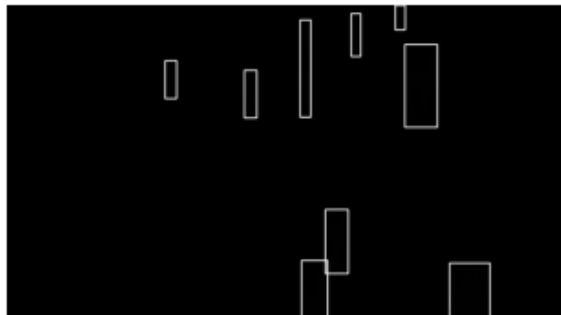
Median - 20 frames - pedestrians_test_02



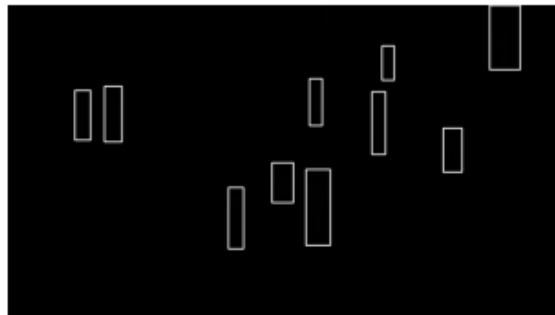
قسمت 5:

برای میانگین:

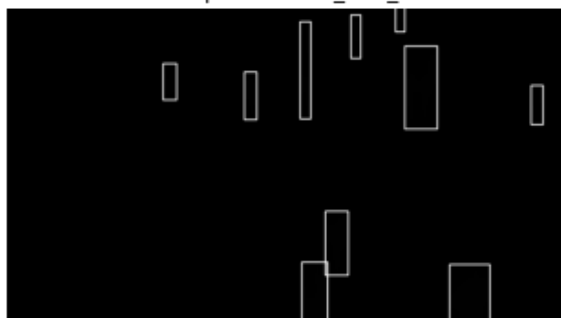
AVG - 2 frames - pedestrians_test_01 - Count: 9



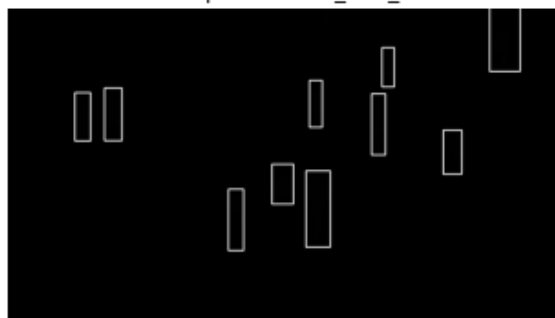
AVG - 2 frames - pedestrians_test_02 - Count: 10



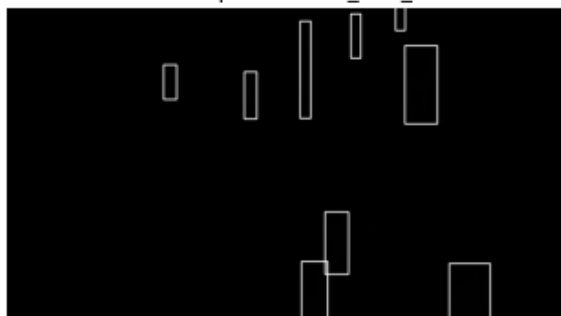
AVG - 5 frames - pedestrians_test_01 - Count: 10



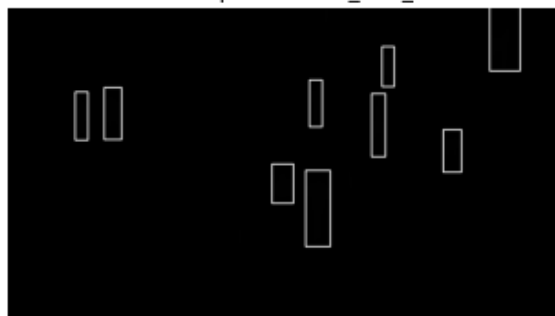
AVG - 5 frames - pedestrians_test_02 - Count: 10



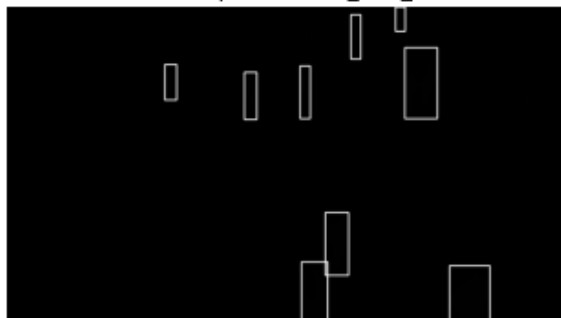
AVG - 10 frames - pedestrians_test_01 - Count: 9



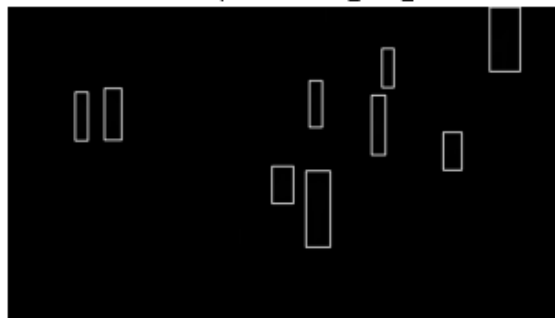
AVG - 10 frames - pedestrians_test_02 - Count: 9



AVG - 20 frames - pedestrians_test_01 - Count: 9

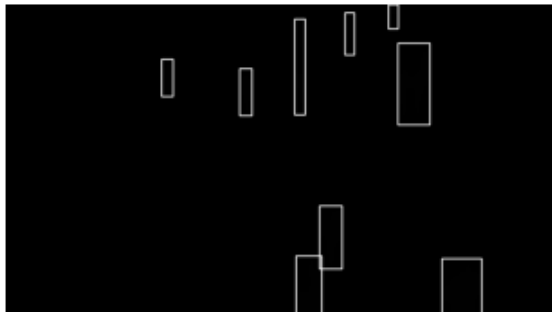


AVG - 20 frames - pedestrians_test_02 - Count: 9

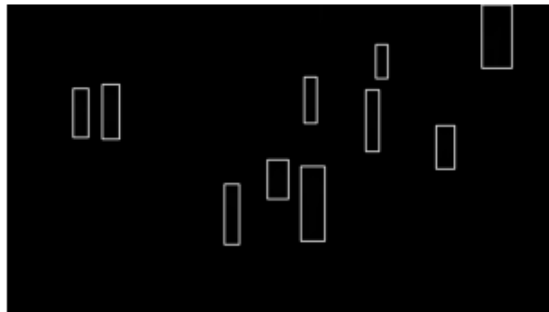


برای median:

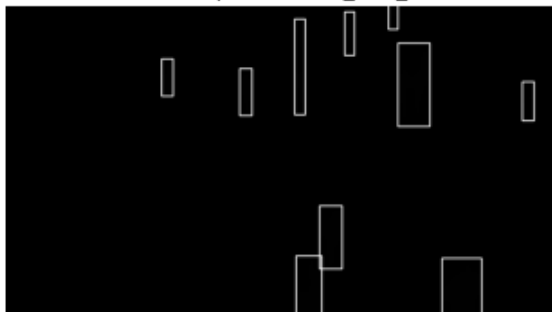
Median - 2 frames - pedestrians_test_01 - Count: 9



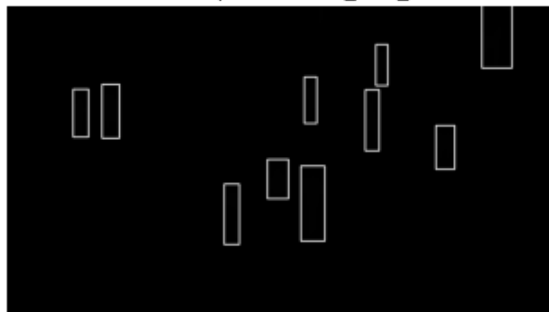
Median - 2 frames - pedestrians_test_02 - Count: 10



Median - 5 frames - pedestrians_test_01 - Count: 10



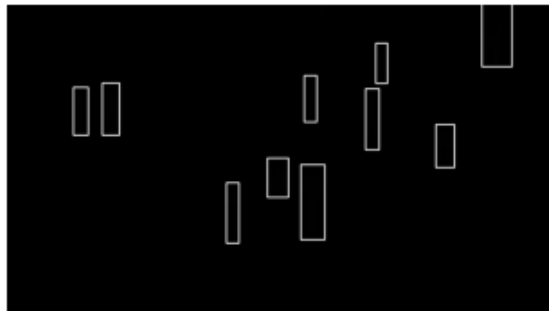
Median - 5 frames - pedestrians_test_02 - Count: 10



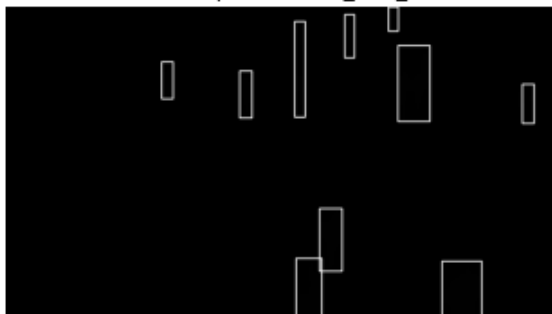
Median - 2 frames - pedestrians_test_01 - Count: 9



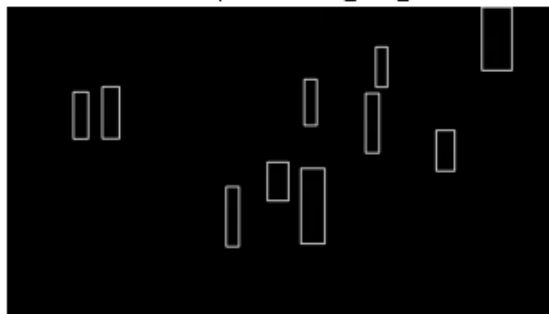
Median - 2 frames - pedestrians_test_02 - Count: 10



Median - 5 frames - pedestrians_test_01 - Count: 10



Median - 5 frames - pedestrians_test_02 - Count: 10



سوال 4)

(A)

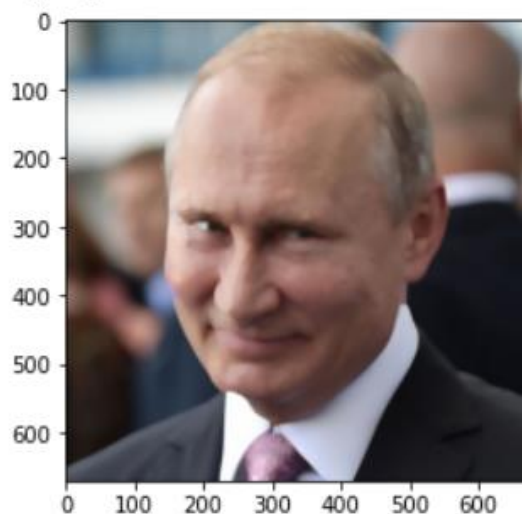
در این سوال برای پیاده سازی bilateral filter در ابتدا ابعاد پنجره مورد نظر را مشخص میکنیم سپس یک ماتریس جدید با ابعاد مشخص شده که حاصل تفاضل ابعاد اصلی تصویر و پنجره است تشکیل میدهیم. سپس در هر تصویر اصلی ناحیه همسایگی را برای هر پیکسل مشخص کرده و سپس به آرایه جدید منتقل میکنیم و با استفاده از فرمول مشخص شده در سوال وزن برای هر خانه در پنجره را مشخص میکنیم. سپس با میانگین گیری وزن های نهایی پنجره را بدست میاوریم

(b)

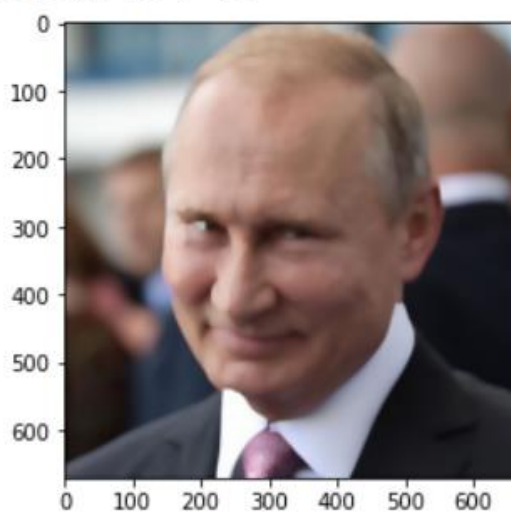
در این قسمت با استفاده از آزمایش های مختلف 4 تا از بهترین نتایج بدست آمده را برای هر عکس نشان میدهیم.

پوتین:

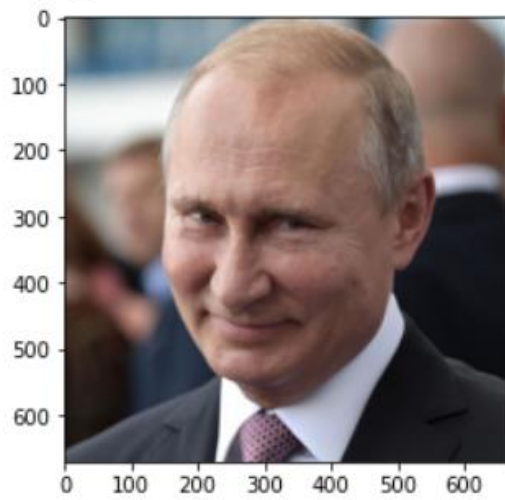
```
window is : 9  
sigma_r is : 40  
sigma_d is : 60
```



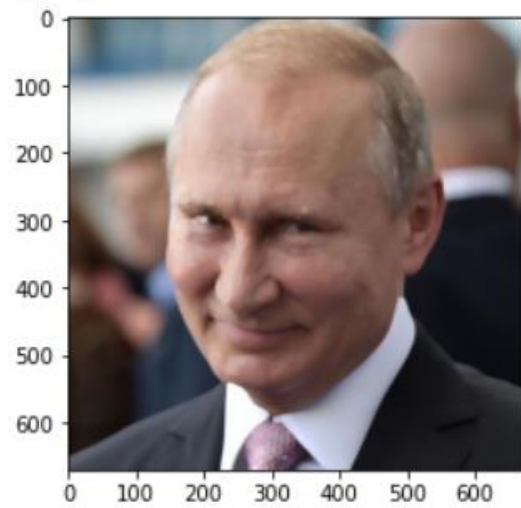
```
window is : 11  
sigma_r is : 50  
sigma_d is : 70
```



window is : 3
sigma_r is : 15
sigma_d is : 20

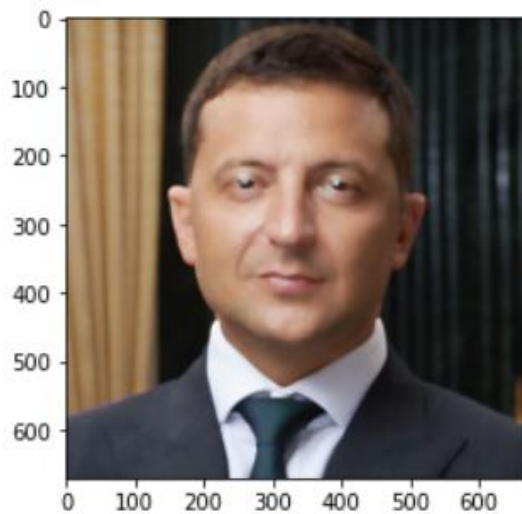


window is : 7
sigma_r is : 30
sigma_d is : 40



زلنيسكى:

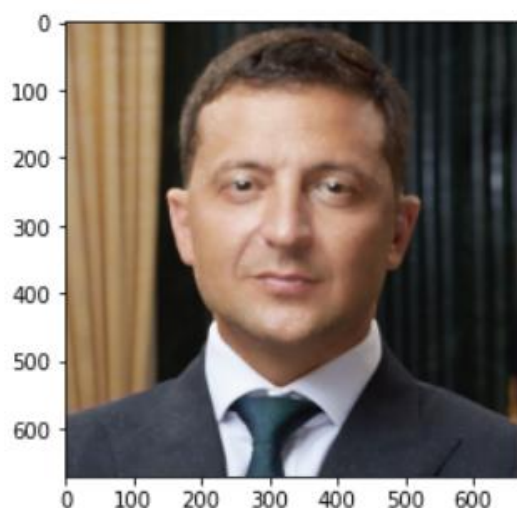
window is : 7
sigma_r is : 40
sigma_d is : 50



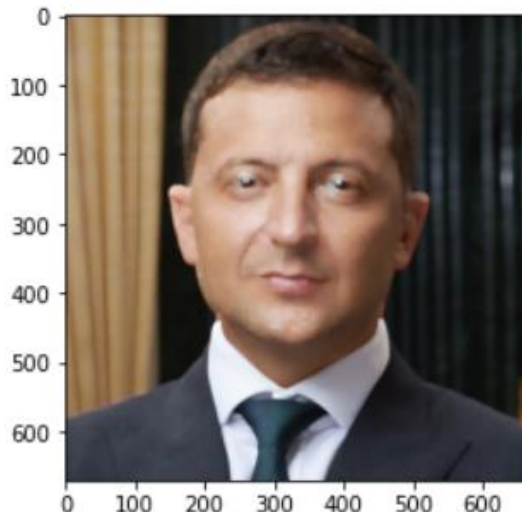
window is : 9
sigma_r is : 50
sigma_d is : 70



```
window is : 5  
sigma_r is : 20  
sigma_d is : 30
```

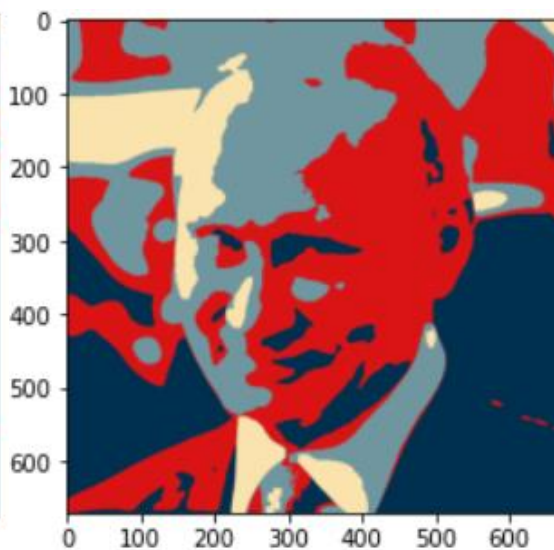
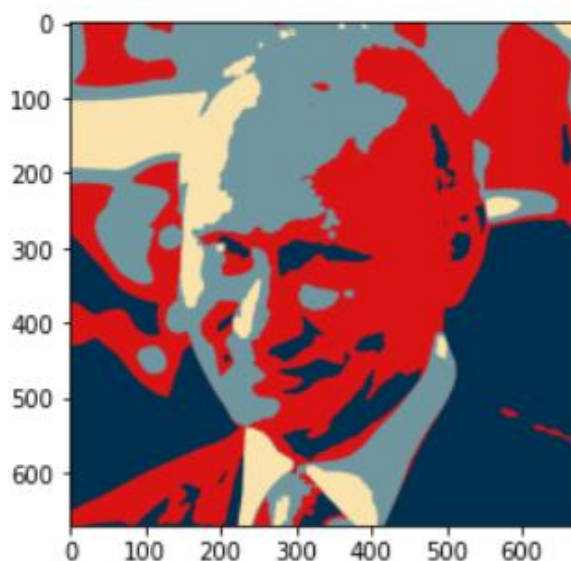


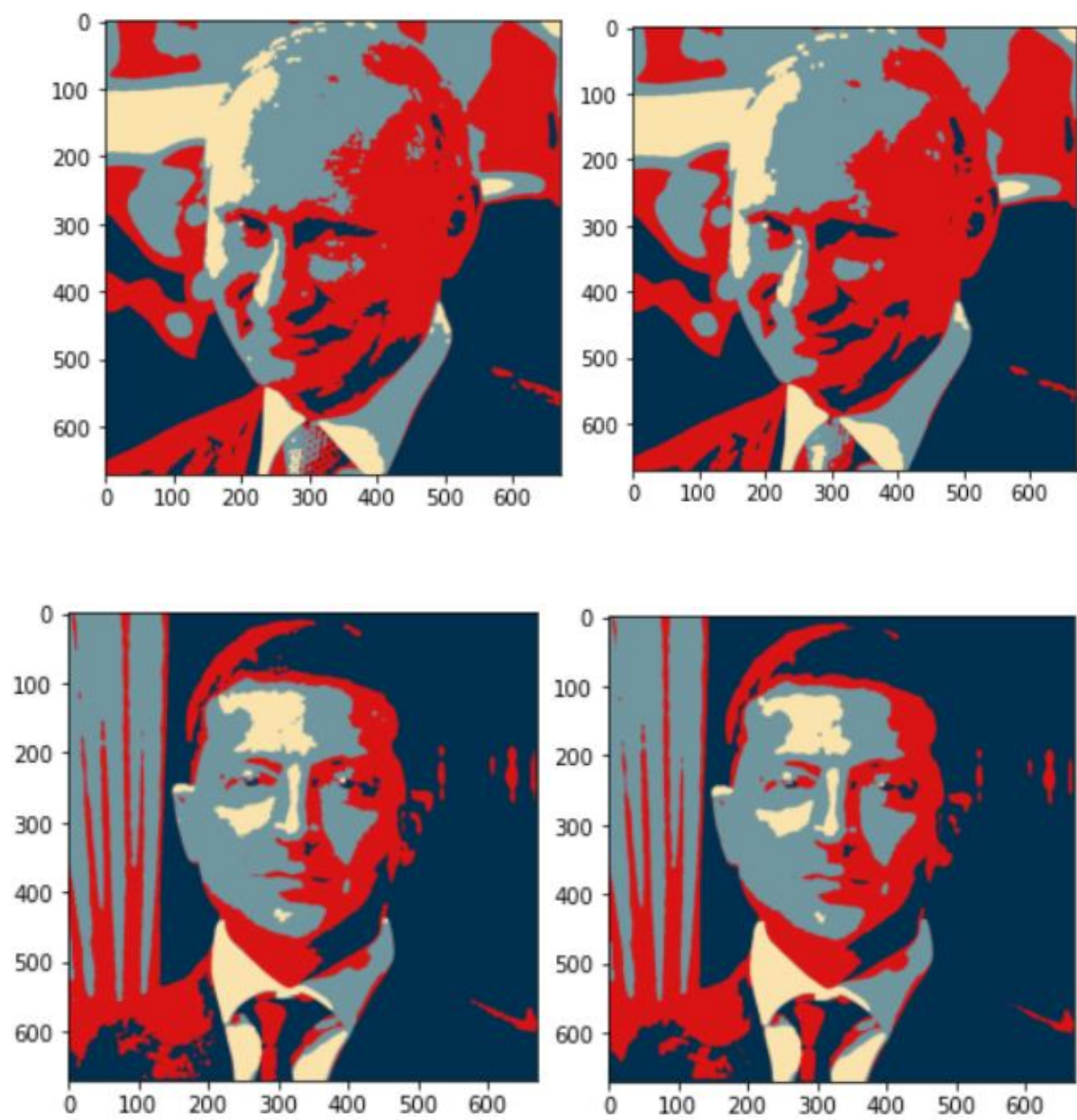
```
window is : 7  
sigma_r is : 30  
sigma_d is : 50
```

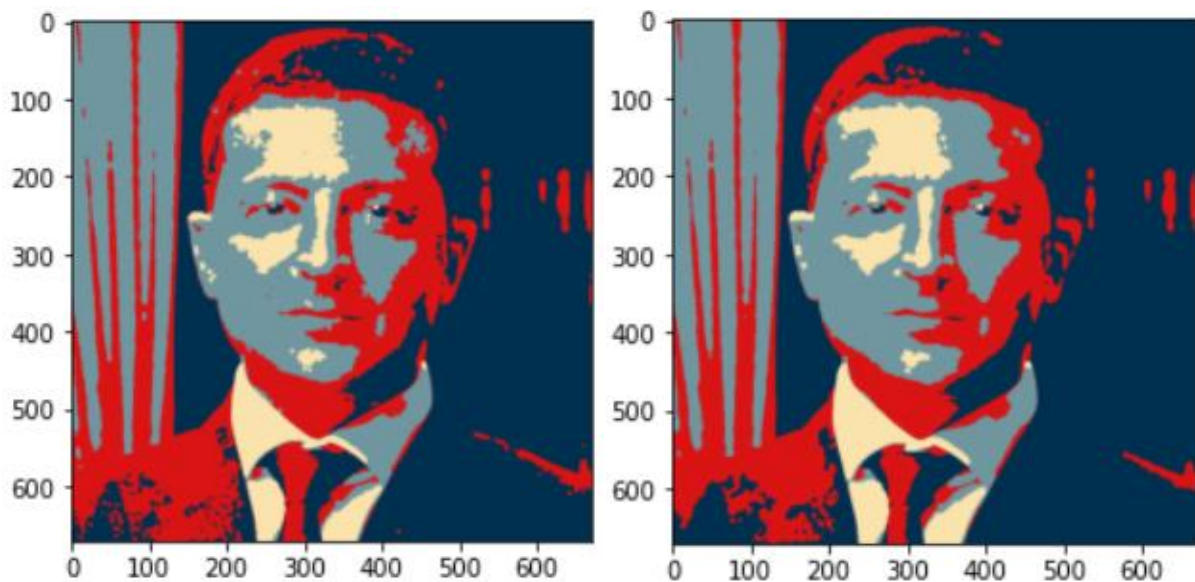


(C) در این قسمت همانطور که در سوال خواسته شده بود رنج ها رنگ معرفی شده (RGB values) برای هر کدام از تصاویر را به صورت زیر تعریف شد.

```
if image[i, j] < 55:  
    color_image[i, j] = [0, 48, 80]  
elif image[i, j] < 125:  
    color_image[i, j] = [218, 20, 21]  
elif image[i, j] < 185:  
    color_image[i, j] = [112, 150, 160]  
elif image[i, j] <= 255:  
    color_image[i, j] = [250, 227, 173]
```



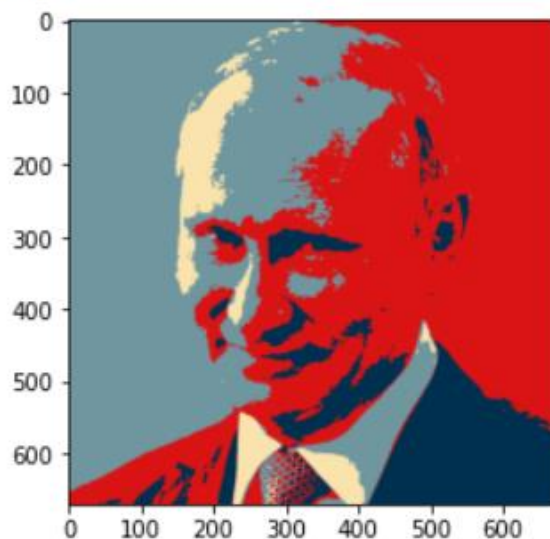




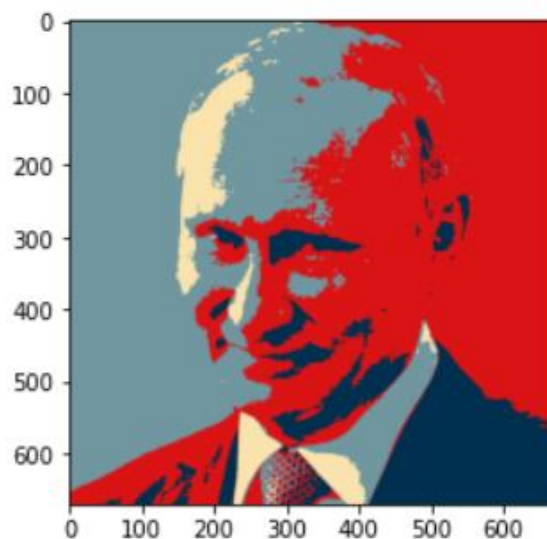
(D)

در این قسمت تمام پیکسل ها به دو قسمت تقسیم کرده و رنگ های معرفی شده را نسبت می دهیم سپس پیکسل هایی که در ماسک سفید هستند را همان مقدار نسبت داده شده در قسمت قبل قرار می دهیم.

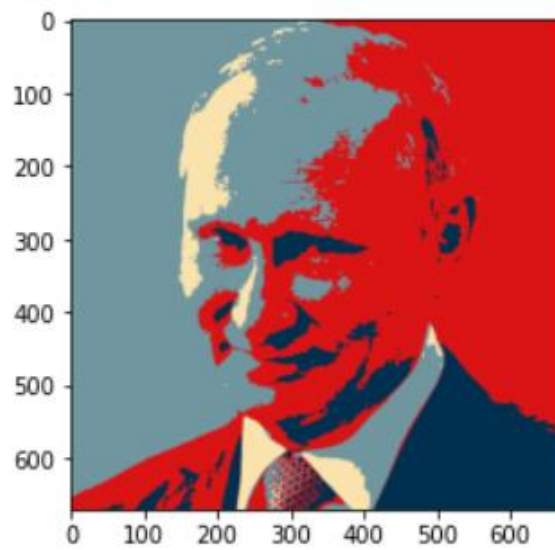
```
window is : 7
sigma_r is : 40
sigma_d is : 60
```



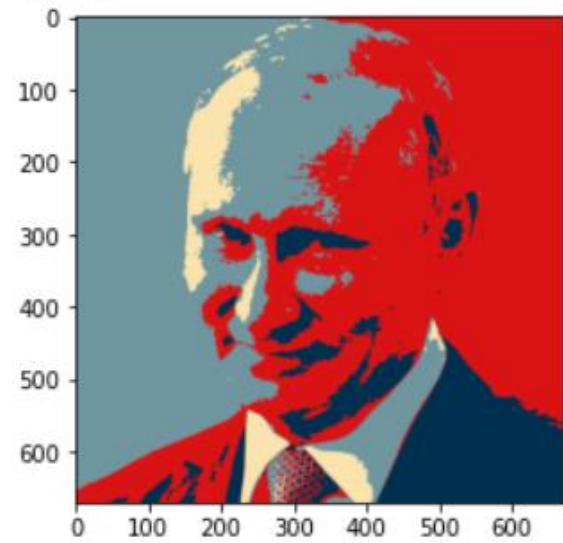
```
window is : 9
sigma_r is : 50
sigma_d is : 70
```



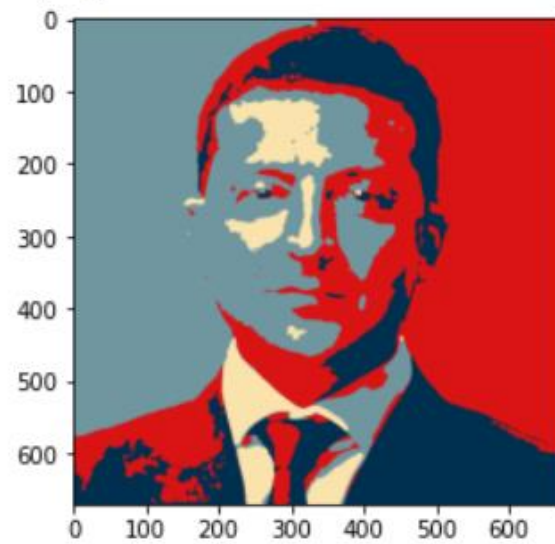
window is : 3
sigma_r is : 20
sigma_d is : 30



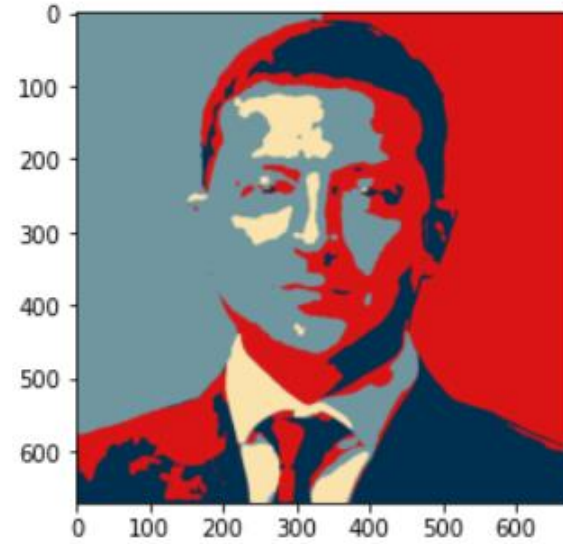
window is : 5
sigma_r is : 30
sigma_d is : 50



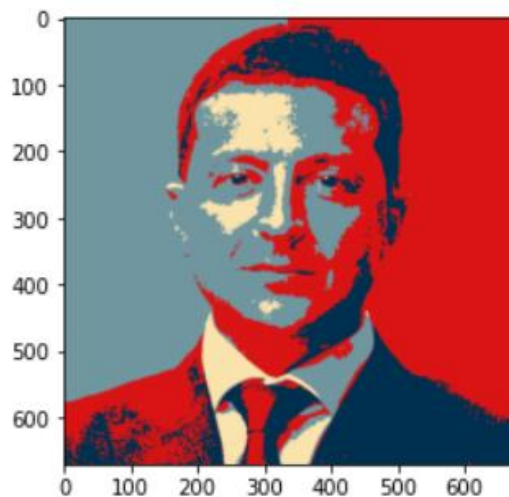
window is : 7
sigma_r is : 40
sigma_d is : 60



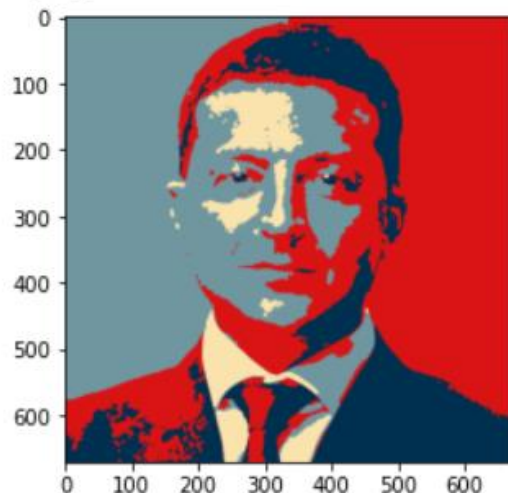
window is : 9
sigma_r is : 50
sigma_d is : 70



```
window is : 3  
sigma_r is : 20  
sigma_d is : 30
```



```
window is : 5  
sigma_r is : 30  
sigma_d is : 50
```



سوال (۵)

در ابتدا توضیحات مختصری از پیاده سازی ارائه میشود.

در بخش اول برای پیاده سازی تصویر را گرفته و تعداد برای عرض و طول، 90 درصد، 75، و 50 درصد آن ها را حساب میکنیم.

در ابتدا 3 آرایه برای محاسبه نقشه انرژی در نظر میگیریم که هر 3 نزدیک در هر ردیف میشود. برای کم کردن Seam در صورتی که بخواهیم طول را کم کنیم آن را 90 درجه میچرخانیم و مانند کم کردن عرض با آن رفتار میکنیم.

(توجه: برای بهتر مشخص کردن انرژی، به نقاطی که انرژی کمی بالاتری داشتند رنگ دیگری اعمال شد تا این اختلاف باعث خطا نشود)

کم کردن ابعاد تصویر:

برای محاسبه نقشه انرژی، انرژی را با مجموع مقدار مطلق گرادیان در هر دو جهت X و Y برای هر سه کانال (R, G, B) محاسبه می کنیم. نقشه انرژی یک تصویر دو بعدی با ابعاد مشابه تصویر ورودی است.

حال برای محاسبه کم ترین انرژی از بالا به پایین، ماتریکسی دو بعدی میسازیم و مقدار هر پیکسل برابر است با مقدار متناظر آن در نقشه انرژی که با حذف یکی از سه همسایه بالای آن (بالا-چپ، بالا-مرکز و بالا-راست) به حداقل انرژی همسایه جدید معرفی شده اضافه میکنیم

سپس از پایین ردیف، کمترین انرژی ها محاسبه میکنیم و با بالا رفتن در هر ردیف، کمترین انرژی برای رسیدن از پیکسل مورد نظر تا پایین ردیف را محاسبه کرده و سپس ردیف های پایین نیز مقادیرشان آپدیت میشود. در واقع در پیکسل باید از طریق پیکسل های همسایه ردیف پایینی به آخرین ردیف برسیم و با هر بار بالا رفتن، مقادیر دیده شده در ردیف های پایینی ممکن است عوض شود که باید به روز شوند و سپس بعد از محاسبه کلی ردیف ها، کم انرژی ترین راه را پیدا کرده و seam مورد نظر در نظر میگیریم.

اضافه کردن ابعاد تصویر :

در این قسمت نیز در ابتدا به مقداری که میخواهیم ابعاد تصویر عوض شود در یک کپی از تصویر عملیات کاهش تصویر را انجام میدهیم. سپس ترتیب seam های کم شده را پیدا میکنیم و حال بر روی تصویر اصلی با همان ترتیبی که پیدا کردیم seam اضافه میکنیم.

ولی با توجه به نداشتن مقدار پیکسل ها از میانگین چپ و راست پیکسل های کناری استفاده میکنیم.

حذف تصویر:

هنگام تولید نقشه انرژی، ناحیه محافظت شده توسط ماسک با مقدار منفی بسیار بالایی وزن دهی می کنیم. این تضمین می کند که حداقل درز از طریق ناحیه ماسک شده هدایت شود. برداشتن درز به طور مکرر انجام می شود تا زمانی که ناحیه ماسک شده به طور کامل همانطور که در بالا گفته شد با یک مرحله دیگر حذف شود. همان حداقل انرژی نیز باید از ماسک برداشته شود تا نقشه انرژی صحیح برای مرحله بعدی بدست آید.

در مرحله بعد به مقدار حذف شده از ابعاد تصویر، seam اضافه میکنیم.

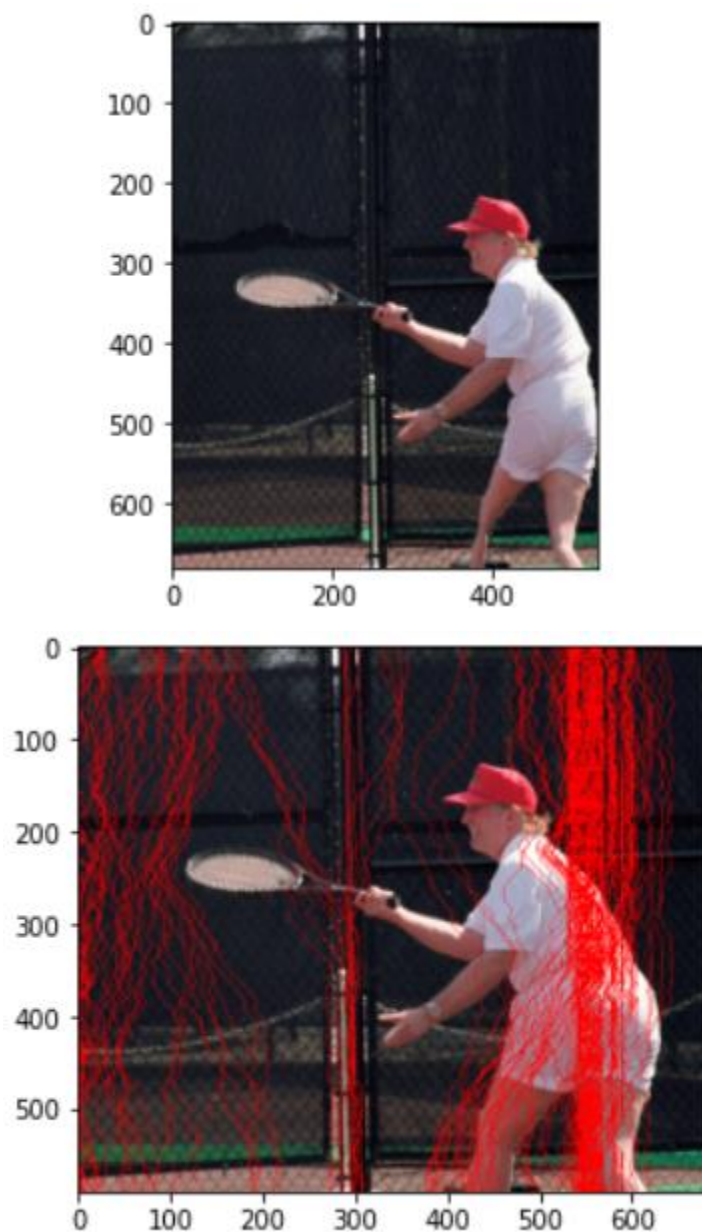
برای نشان دادن seam در هر تصویر نیز، پیکسل پیدا شده در هر ردیف را قرمز میکنیم.

(توجه: با توجه به اینکه در سوال خواسته شده بود از هر قسمت عکس تهیه شود، من Seam های پیدا شده در هر تصویر را بعد از مشخص کردن به گیف تبدیل میکنم ولی با توجه به تعداد زیاد تصاویر، حجم گیف مورد

نظر بسیار بالا می‌رود، به همین دلیل تنها برای کاهش های 10 درصدی این گیف را درست کرده ام و برای بقیه مقادیر تنها نتیجه نهایی را آورده ام)

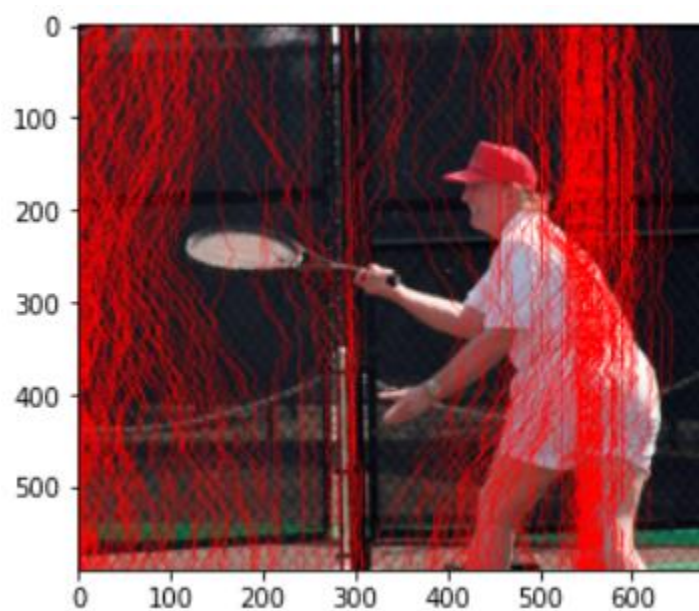
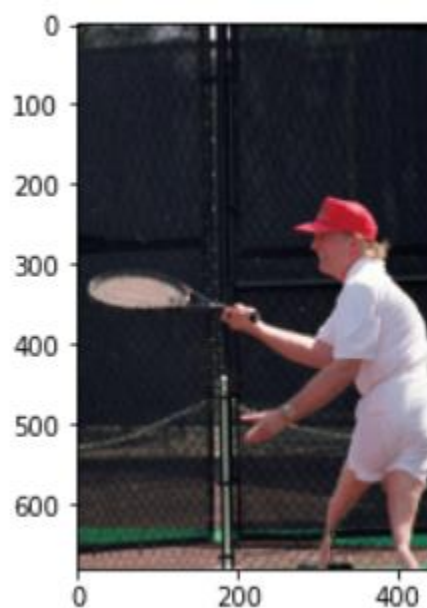
تصویر donald_plays_tennis:

کاهش 10 درصد عرض:

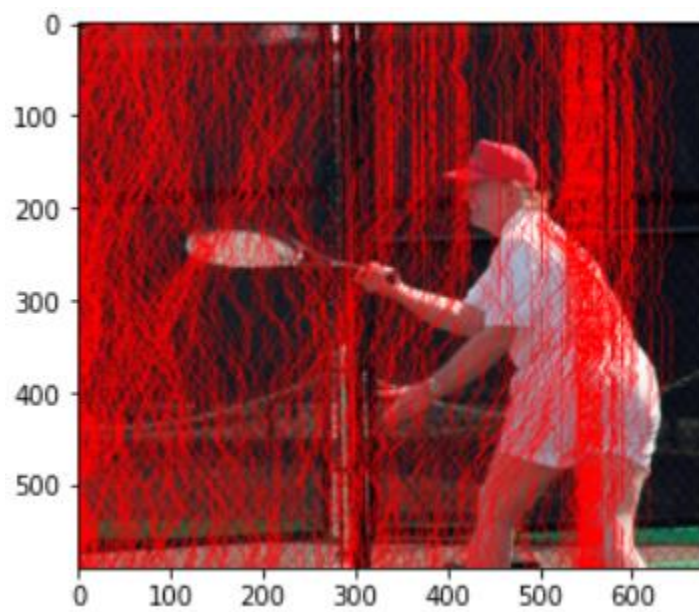
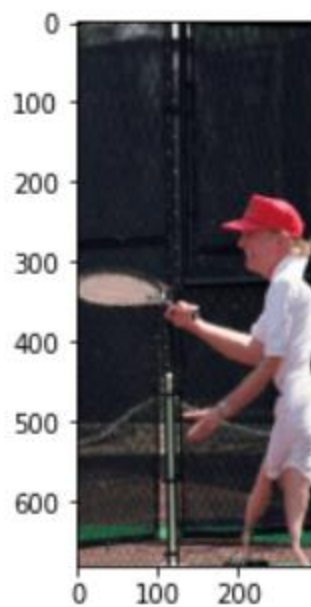


فایل گیف در فولدر موجود است.

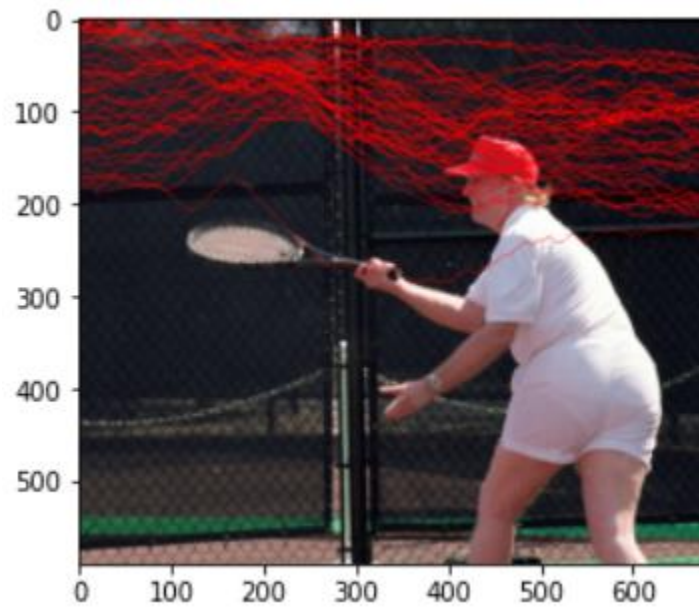
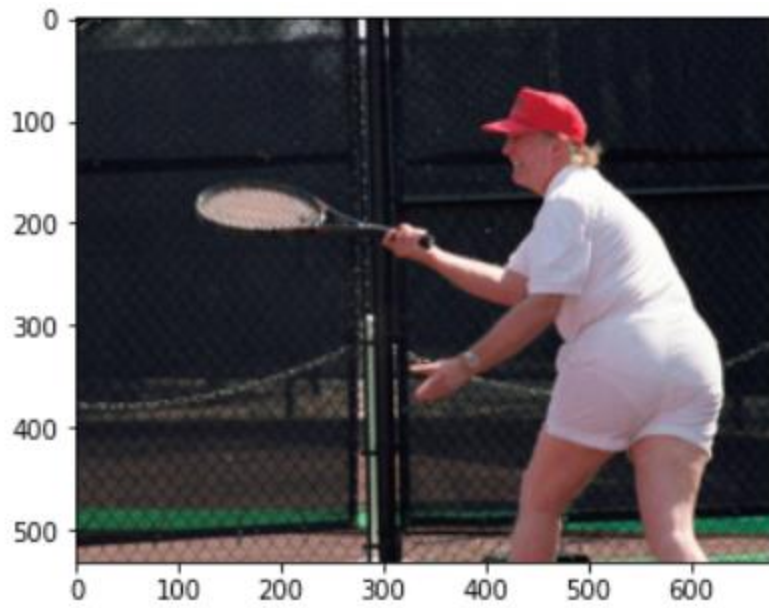
کاهش 25 درصد عرض:



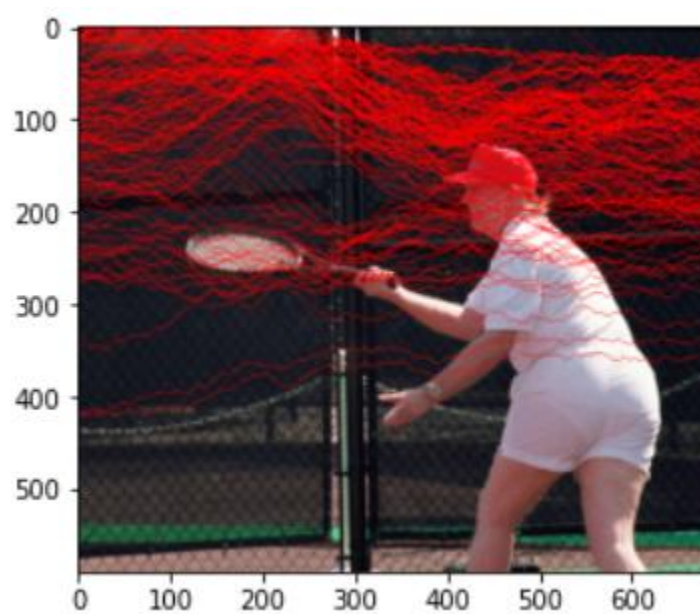
کاهش 50 درصد عرض:



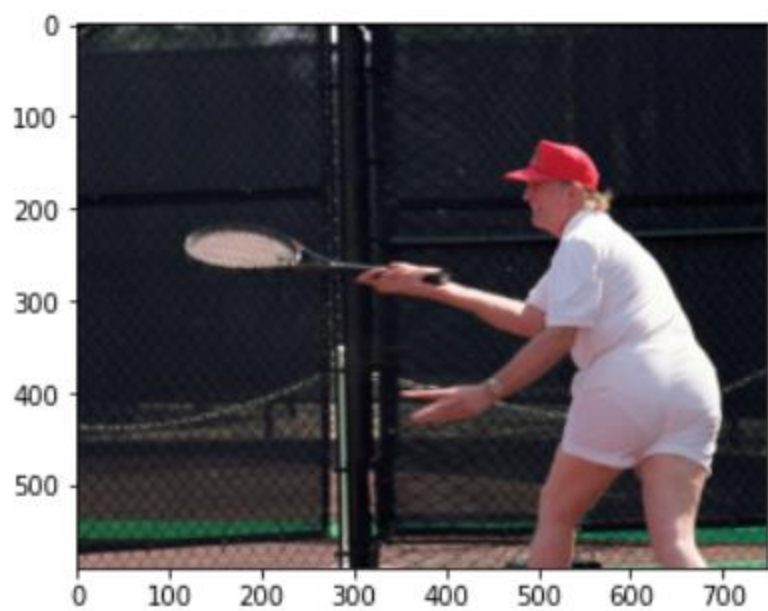
کاهش 10 درصد طول:



کاهش 25 درصد طول:



افزایش 10 درصدی عرض:



افزایش 25 درصدی عرض:



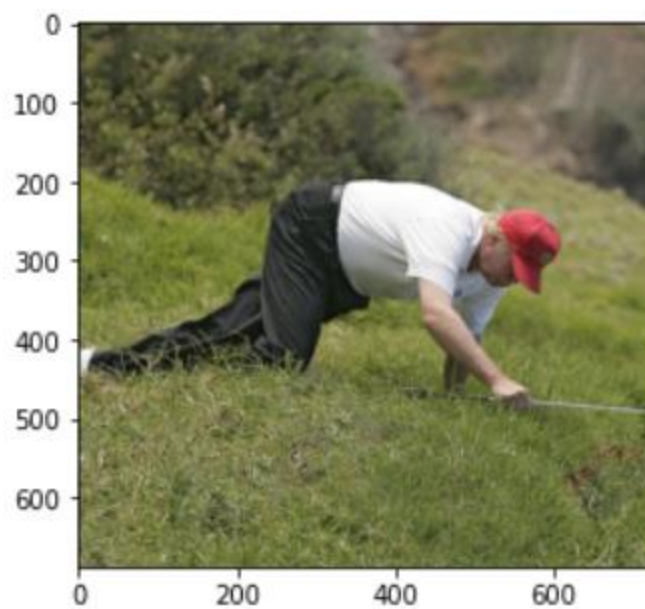
افزایش 50 درصدی عرض:



تصویر donald_plays_golft:

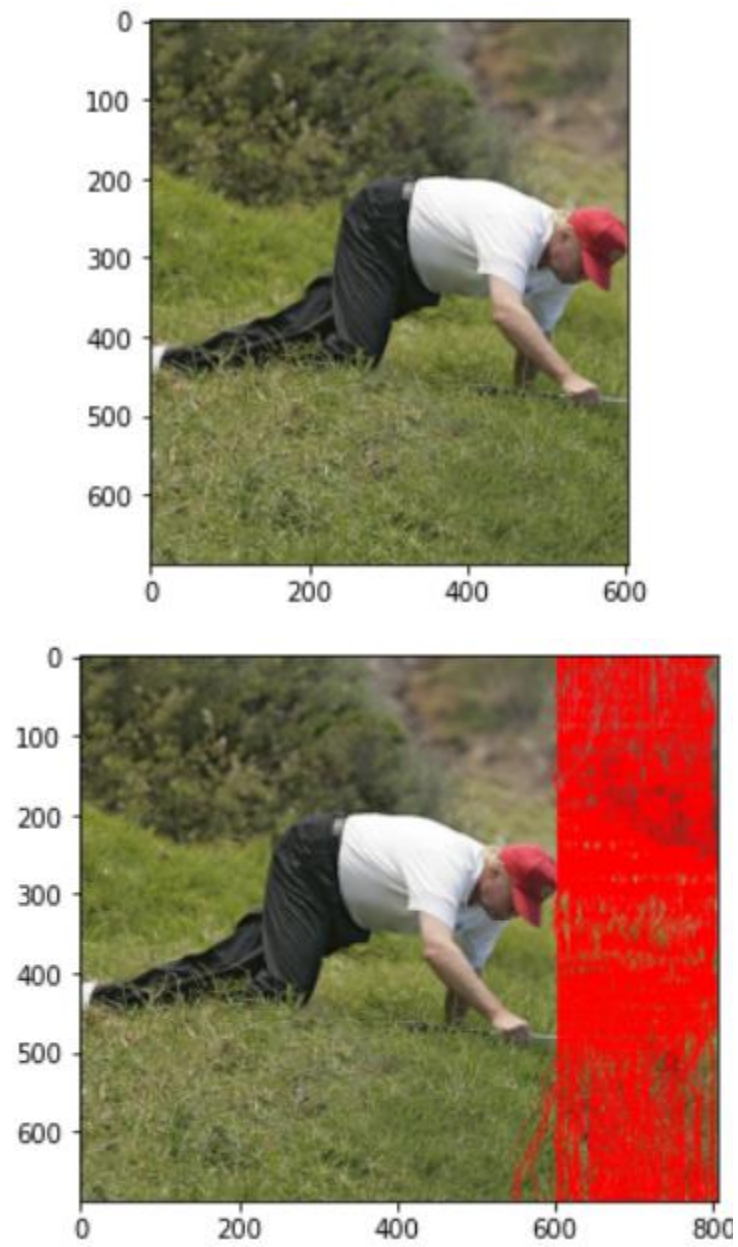
متأسفانه در این تصویر به وضوح تغییرات بین ترامپ و محیط مشخص نیست و به همین دلیل مدل به درستی نمیتواند آن را شناسایی کند. برای حل این مشکل میتوان با استفاده از فوتوشاپ رنگ های لباس را کمی تیره تر کرد یا یک الگوی خطی بر روی ترامپ ایجاد کرد تا انرژی بالا تر رود. به دلیل مسلط نبودن به برنامه فوتوشاپ از این کار صرف نظر کردم

کاهش 10 درصد عرض:

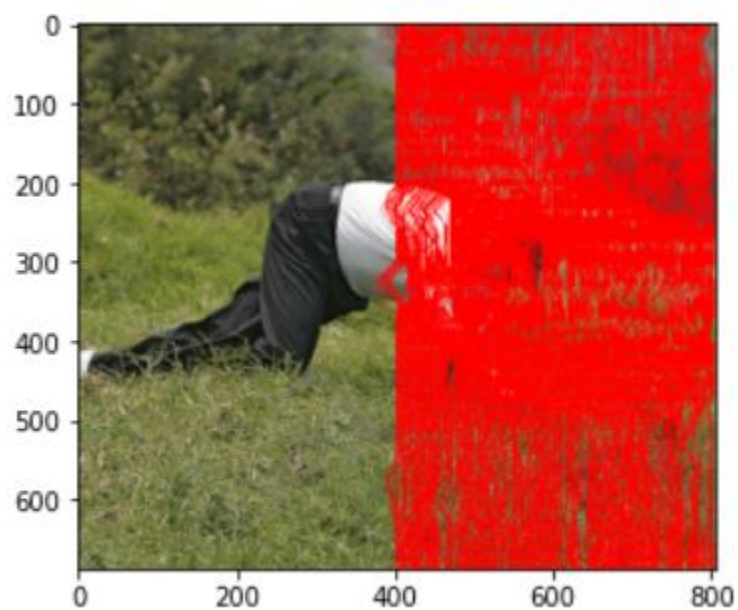
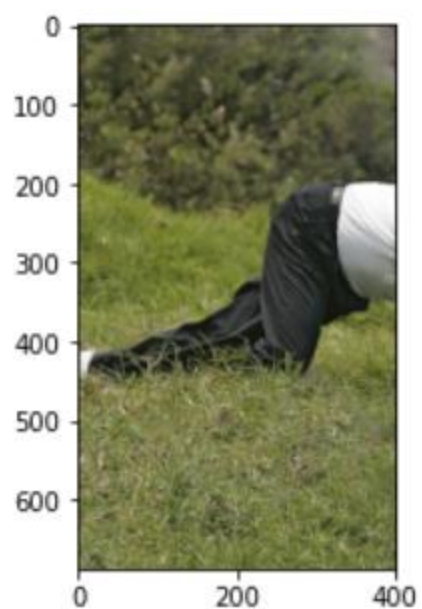


فایل گیف در فولدر موجود است.

کاهش 25 درصد عرض:



کاهش 50 درصد عرض:



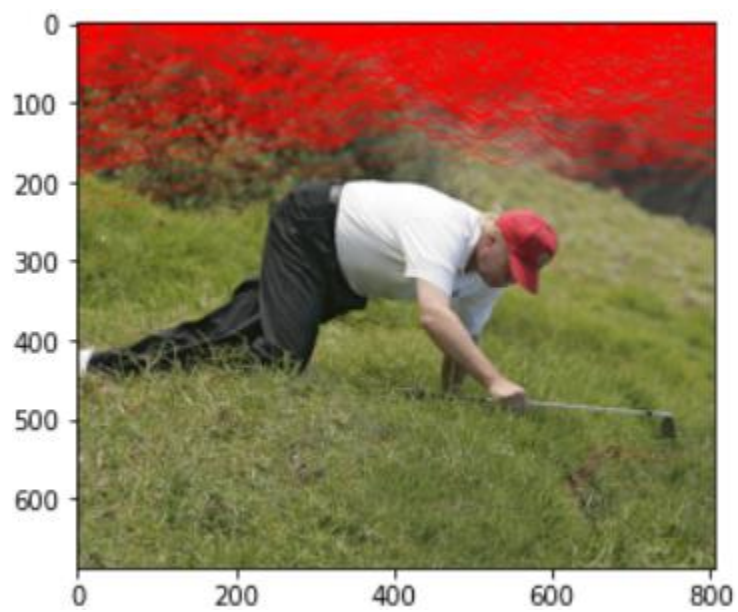
کاهش 10 درصد طول:



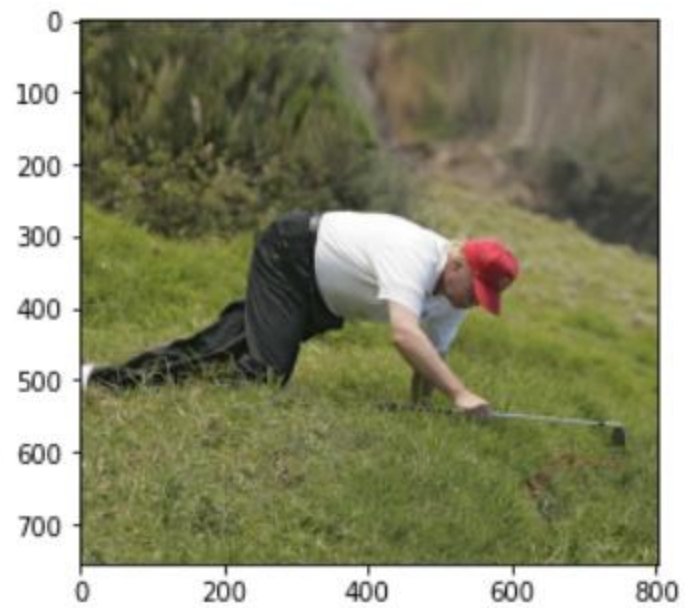
کاهش 25 درصد طول:



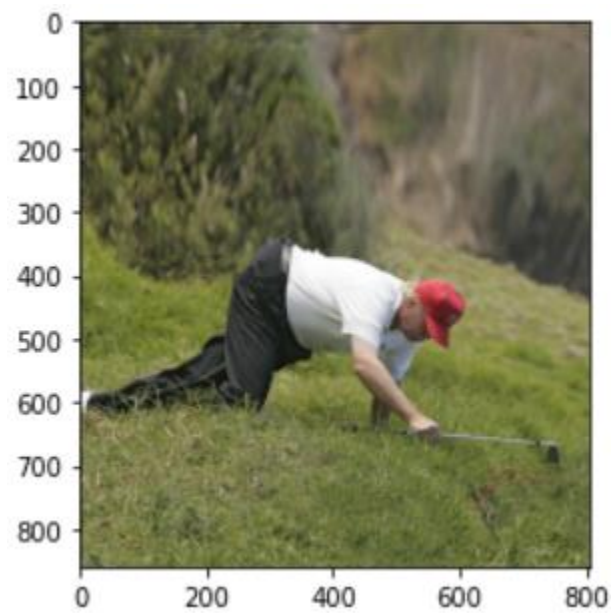
کاهش 50 درصد طول:



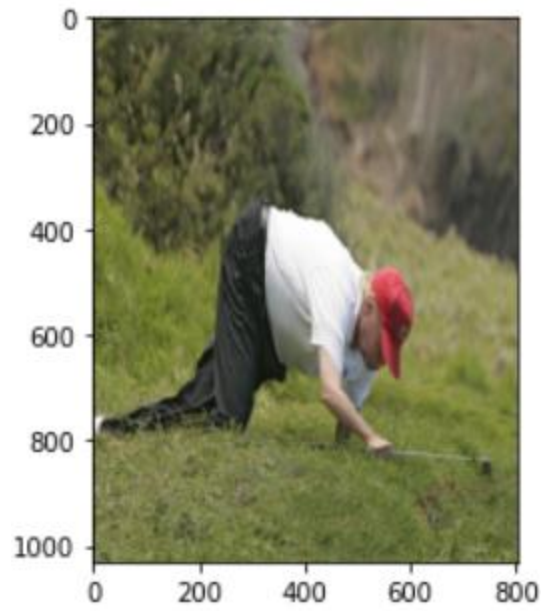
افزایش 10 درصدی ابعاد:



افزایش 25 درصدی ابعاد:



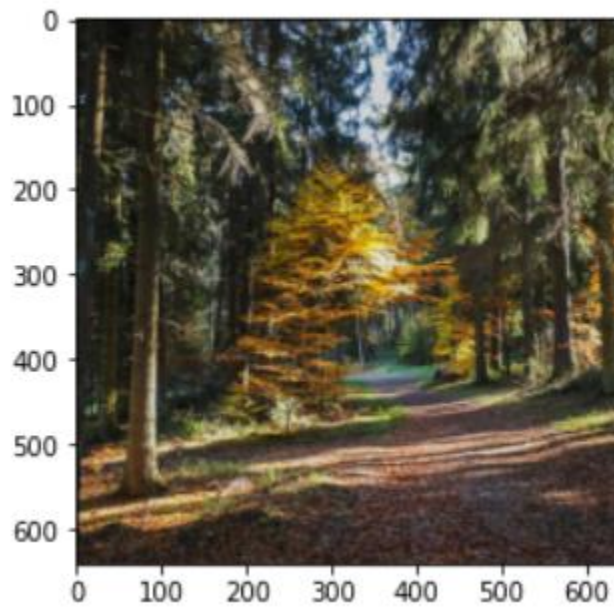
افزایش 50 درصدی ابعاد:



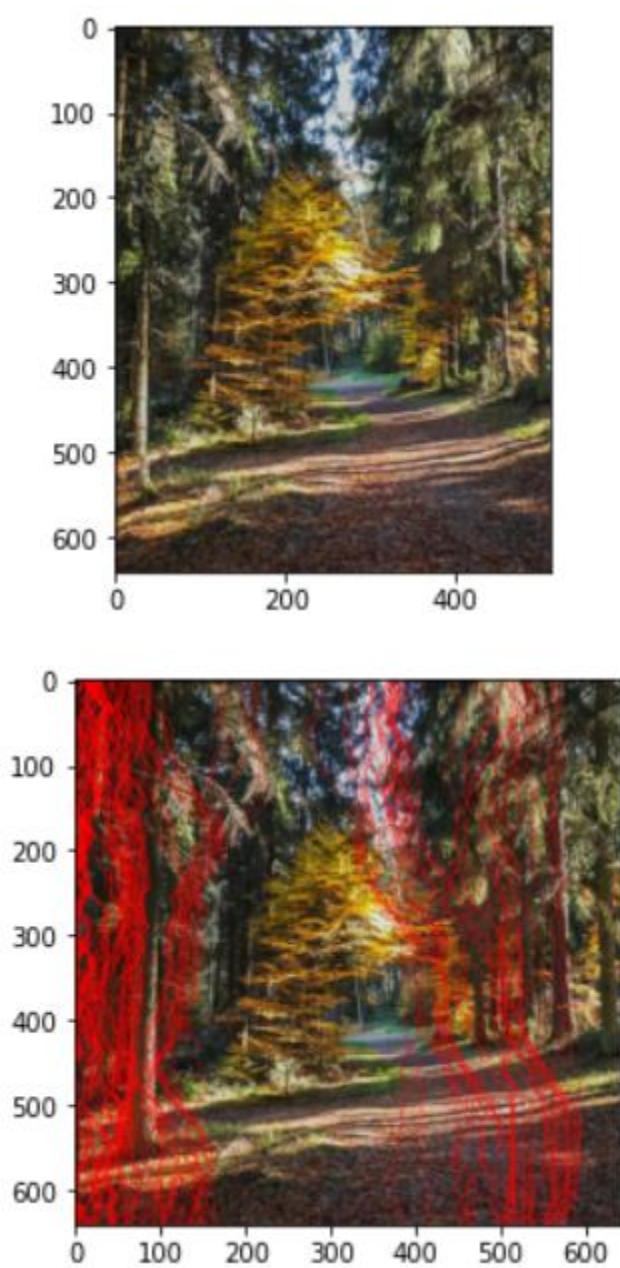
(C

تصویر اصلی:

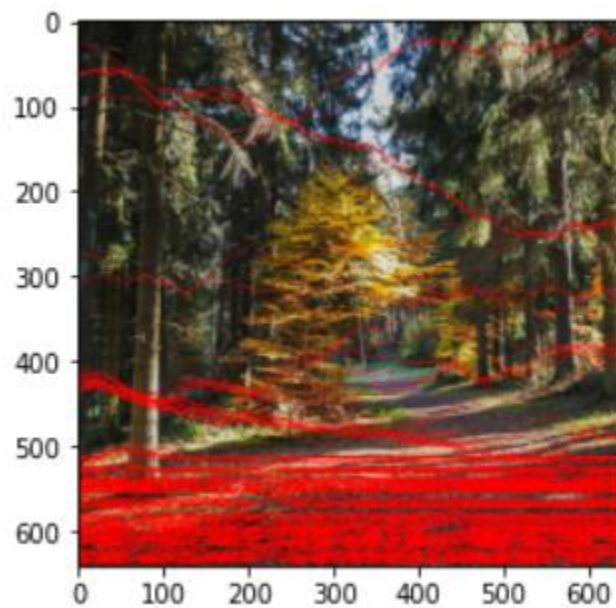
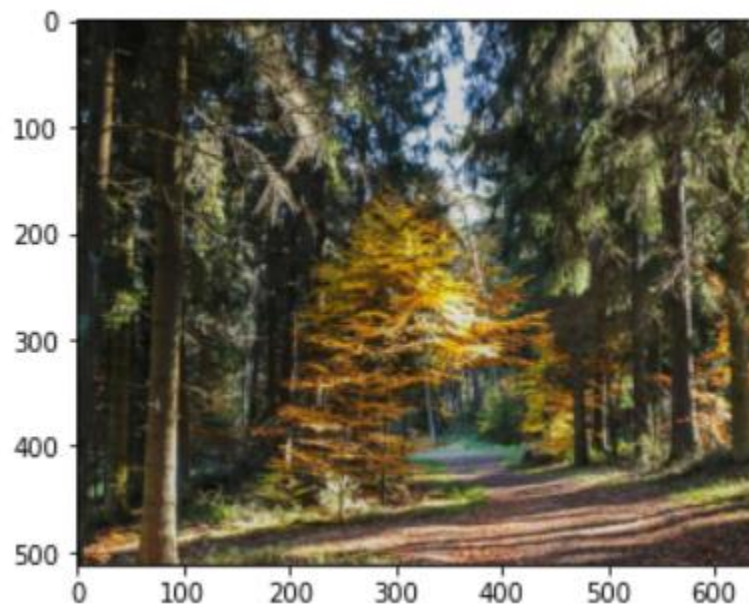
width : 640 height : 640



تصویر با کاهش 20 درصدی عرض:



تصویر با کاهش 20 درصدی طول:



(D)

در این قسمت میبینیم به دلیل اینکه صفحه بسیار اطلاعاتی زیادی را در بر میگیرد با کاهش تصویر اطلاعات زیادی پاک میشود.

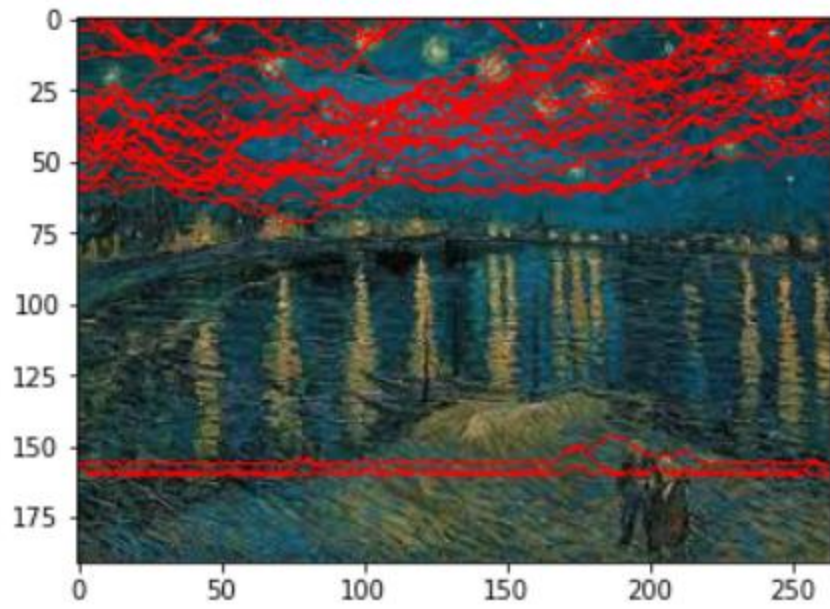
تصویر اصلی

width : 264 height : 191

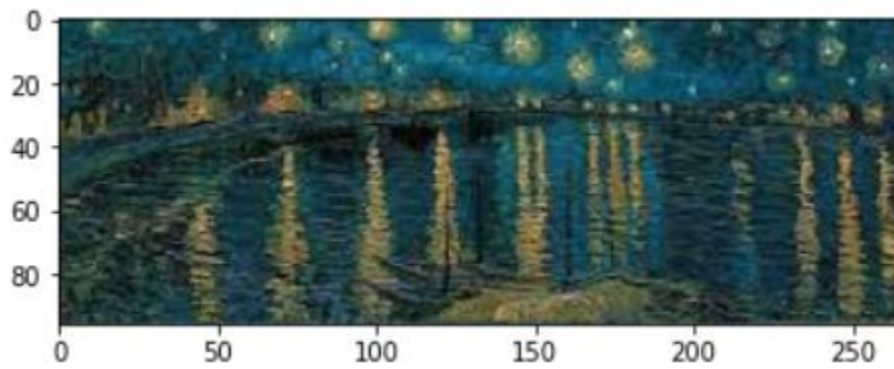


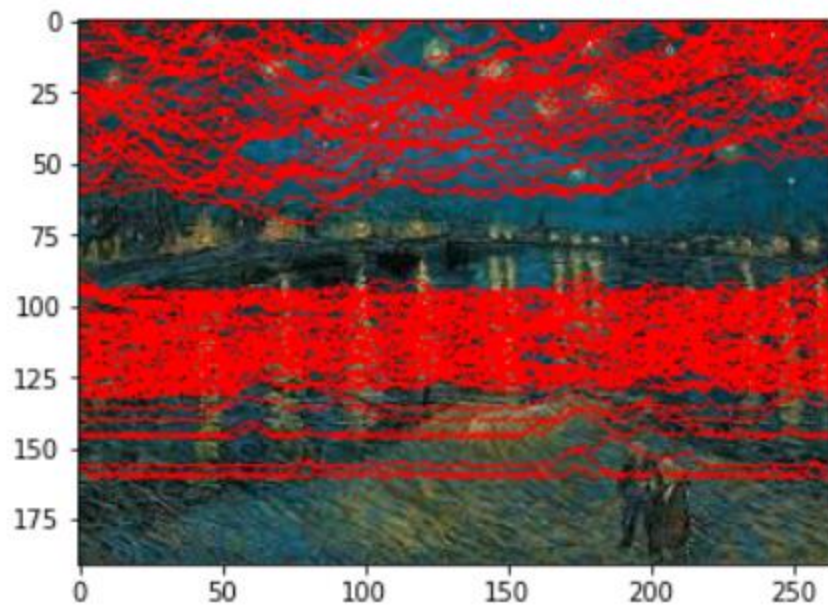
کاهش 20 درصدی عرض:



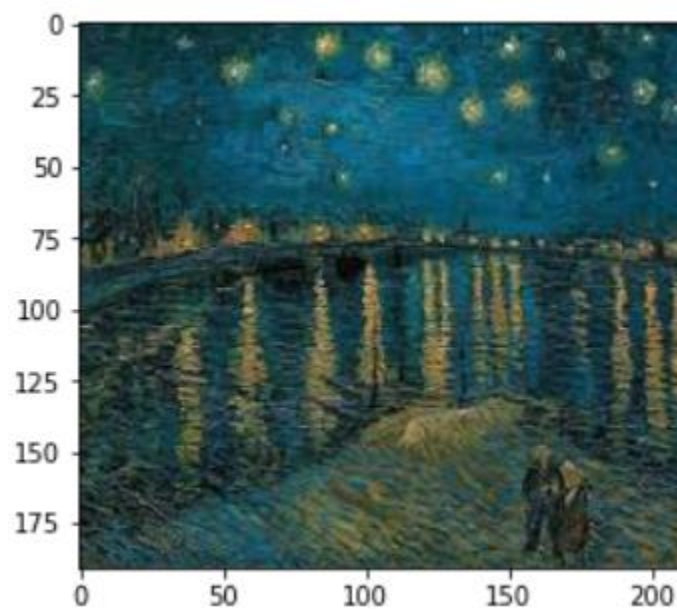


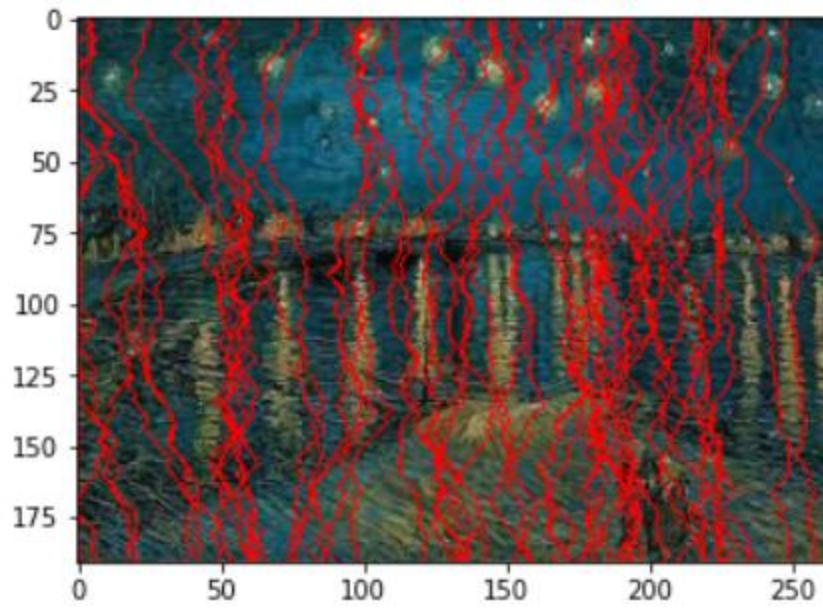
کاهش 50 درصدی عرض:





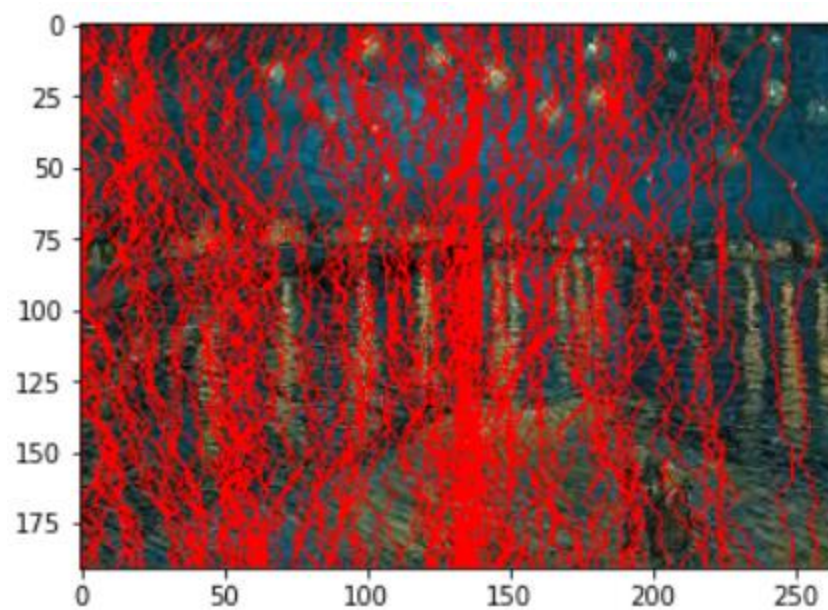
کاهش 20 درصدی طول:





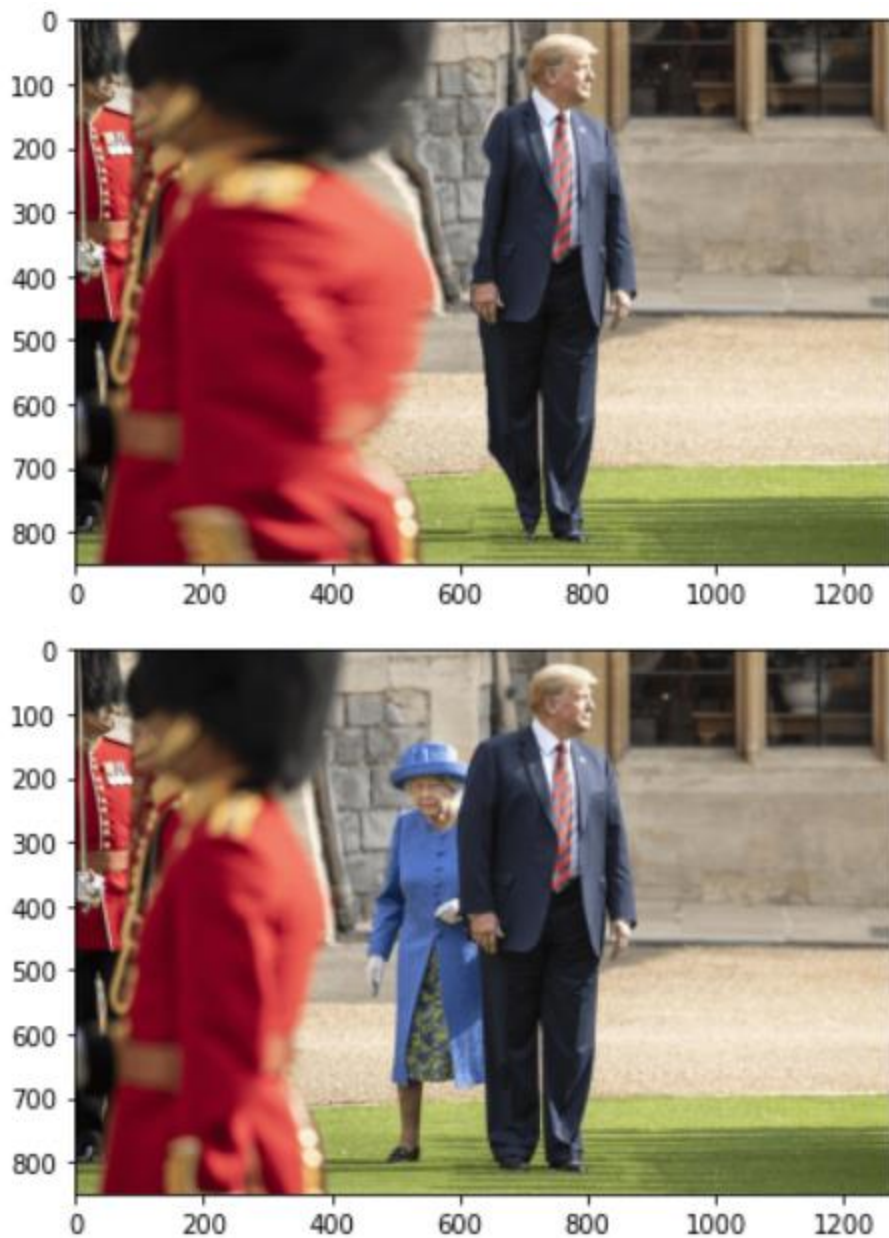
کاهش 50 درصدی طول:





(F)

تصویر ملکہ



سوال 6)

(A)

بله، گاهی با توجه به اینکه در لبه های تصاویر اطلاعات ارزنده وجود دارد نیاز داریم تا گوشه تصاویر را نیز در convolve قرار دهیم. برای اینکار به تعداد نیاز به گوشه های تصاویر پیکسل اضافه میکنیم که مقادیر مربوط به آن ها نیز معمولاً صفر در نظر میگیریم. به همین دلیل ممکن است که سایز فیلتر ممکن است بزرگتر باشد.

(B)

بله، برای اینکه عمل لاپلاسیون در واقع عمل مشتق گیری را انجام می‌دهد که باتوجه اینکه تابع ما، یک تابع با مقادیر ثابت است، در نهایت به سمت صفر و به یک حالت خاص همگرا میشود.

(C)

امکان پذیر نیست، چرا که سائز فیلتر همیشه ثابت است و نمیتواند الگوهای مختلف را شناسایی کند.

(D)

دو تا فیلتر 1 بعدی بهتر است، زیرا در این فیلتر میتوان برای مثال مجموعه ای از تصاویر دوبعدی به 1 بعد کاهش داد و سپس با استفاده از فیلتر دوم، سائز آن را باز کوچک تر کرد. با اینکه در فیلتر دو بعدی نیز این حرکت ممکن است ولی در استفاده از دو فیلتر تک بعدی، بار محاسباتی کم شده و سرعت بالا میرود.

(E)

این فقط برای نویز تصادفی با میانگین صفر کار می کند. اگر نویز تصادفی (به اندازه کافی) باشد، هرگز کاملاً یکسان نیست، در حالی که صحنه ای که از آن عکس می گیرید باید باشد. از آنجایی که اطلاعات مربوط به صحنه چندین بار ضبط می شود، هر بار با نویز کمی متفاوت، می توان میانگین پیکسل ها را به دست آورد و نسبت سیگنال به نویز بالاتری نسبت به یک ضبط دریافت کرد. این بدان معنی است که صحنه باید ثابت باشد.

به عنوان مثال داده زیر را در نظر بگیرید:

104, 99, 98, 100, 101, 105, 99, 102, 94, 105

میانگین این مقادیر (با جمع آنها و تقسیم بر 10) مقدار پیکسل زیر را به دست می دهد: 100.7، که به 101 گرد می شود، که بسیار نزدیک تر از مقدار واقعی است که در صورت انتخاب تنها یکی از موارد انتظار می رود. 10 عکس به صورت تصادفی