به نام خدا

درس دید کامپیوتری

گزارش تمرین شماره یک بخش اول

علیرضا بانشی

95101185

سوال اول

(1

بخش الف

ابتدا به کمک دستور imread و به کمک flag های 0 و 1 که به ترتیب تصویر را سیاه سفید و رنگی میخواند تصویر را می خوانیم.
سپس با کمک دستور زیر متن دلخواه را روی عکس می نویسیم:

```
cv2.putText(a,'95101185',(10,1450), font, 1,(255,255,255),2)
```

آرگومان ها به ترتیب عکس متن پوزیشن متن فونت اسکیل فونت رنگ و ضخامت است.
سپس به کمک WAITKEY و تشخیص کلید موردنظر تصویر را با کمک دستور imwrite ذخیره می کنیم.
خروجی در زیر آمده است و همانطور که مشاهده می شود شماره دانشجویی در پایین و سمت چپ تصویر درج شده است.


95101185


95101185

بخش ب

پس از خواندن تصویر مورد نظر ابتدا مختصات محل توپ را استخراج می کنیم و یک مستطیل دور آن می کشیم.
برای این کار از دستور زیر استفاده می کنیم:

```
b = cv2.rectangle(a, (290, 455), (380, 531), (0, 255, 0), 3)
```

آرگومان های آن به ترتیب تصویر مختصات گوشه بالا سمت چپ مختصات گوشه پایین سمت راست رنگ و ضخامت مستطیل است.
حالا یک تصویر ROI درست می کنیم.این تصویر ناحیه دلخواه ما برای کپی کردن است و به سادگی با دادن سطر ها و ستون های مورد نظر از تصویر اصلی آن را می سازیم.

```
roi = a[455:531,290:380]
```

حالا کافی است این تصویر ROI را جایگزین بخش دلخواهی از تصویر اصلی کنیم.

```
a[455:531,590:680] = roi
```

ودر پایان تصویر خروجی را مشاهده می کنیم.


Fly
Emirates
7
Rakuten
10

(2

با استفاده از دستور زیر یک رابط کاربری تهیه می کنیم که به صورت یک اسلاید بار است و کاربر می تواند با حرکت دادن آن میزان چرخش را مشخص کند.

```
cv2.createTrackbar("Degree", "Rotation",0,360,nothing)
```

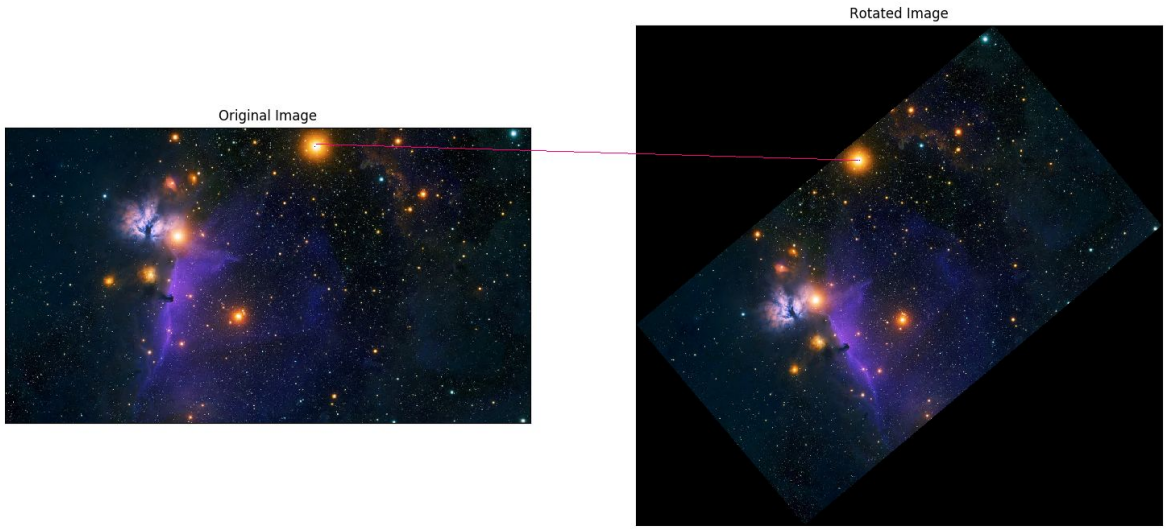
وارگومان های آن نام ترکر نام ترکر فعلی مینیم و ماکزیمم و ارگومان اخر هم چیز خاصی نیست یک تابع تعریف شده که کاری انجام نمی دهد.

سپس در یک حلقه به طور دائم مقدار ترکر را می خوانیم و با کمک دستور زیر تصویر را به مقدار وارد شده می چرخانیم.

```
a = cv2.getTrackbarPos("Degree", "Rotation")
rotated = ndimage.rotate(img, a)
```

در انتها نیز به کمک subplot و با کمک کتابخوانه matplotlib هر دو را کنار هم می کشیم..

نمونه ای ازخروجی که به ازای چرخش 45 درجه ای ایجاد شده است در زیر آورده شده است



برای این سوال از دستور آماده پایتون برای چرخش تصویر استفاده کردیم اما میتوان آن را به صورت دستی نیز انجام داد که کد آن در زیر مشخص است.

```
degree = input("give us a number to rotate the image")
rows,cols = img.shape

d = int(math.ceil(math.sqrt((math.pow(cols, 2) + math.pow(rows, 2)))))

rotated_width = int (cols * math.fabs(math.cos(degree * math.pi / 180)) + rows *
math.fabs(math.sin(degree * math.pi / 180)))
rotated_height = int (rows * math.fabs(math.cos(degree * math.pi / 180.0)) + cols *
math.fabs(math.sin(degree * math.pi / 180)))

zimg = np.zeros((d, d))
zimg[(d/2) - (rows/2) : (d/2) + (rows/2), (d/2) - (cols/2) : (d/2) + (cols/2)] = img


M = cv2.getRotationMatrix2D((d/2, d/2), degree, 1)
dst = cv2.warpAffine(zimg,M,(d, d))
r_img = dst[(d/2) - (rotated_height/2) : (d/2) + (rotated_height/2), (d/2) - (rotated_width/2) : (d/2)
+ (rotated_width/2)]
```

به این صورت که ابتدا درجه مورد نظر را ورودی می گیریم سپس عرض و ارتفاع جدید را به کمک فرمول های مشخص شده در بالا بدست می آوریم و به کمک دستورات بعد از آن تصویر چرخیده شده را به دست می آوریم

(5

هنگامی که ما تصویری را erode می کنیم انگار مرزهای اشیا را فشرده می کنیم برای این کار از یک کرنل استفاده می کنیم.

فرض کنیم تمامی پیکسل ها 0 و 1 است هنگام ارود کردن کرنل را روی عکس می چرخانیم به صورت کانولوشن دو بعدی و تنها پیکسال هایی یک می ماند که زیر کرنل آن

ها همه برابر یک باشند برای یک نمونه از erode کردن می توان اشکال زیر را در نظر گرفت.

که پس از اعمال تبدیل به شکل زیر در میاید

اما عمل dilate کردن دقیقا مخالف erode کرن است و تصویر ابجکت را بزرگنمایی می کند به این صورت که حتی اگر یک پیکسل اطراف نیز یک باشد آن پیکسل را یک قلمداد می کنیم.(هنگامی که داریم کرنل را روی تصویر حرکت می دهیم و کانولوشن می کنیم)
نتیجه دایلیت کردن عکس اول در زیر آمده است.

تبدیل opening:

اگر ابتدا dilatwe کنیم و سپس erode کنیم به این کار opening می گویند.
نتیجه در زیر آمده است.

تبدیل closing:
اگر ابتدا erode کنیم سپس dilate به این کار closing می گویند.
نتیجه را مشاهده می کنیم.

ما تبدیل های خواسته شده را با سه کرنل مختلف با ابعاد 3در3 و 5 در 5 و 7 در 7 روی تصویر اعمال کرده ایم و تصاویر خروجی را ضمیمه کرده ایم.دستورات به سادگی زیر می باشد.نام هرتصویر نشان دهنده وضعیت آن است.

```
kernel = np.ones((3,3),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)
#dilation = cv2.dilate(img,kernel,iterations = 1)
#opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
#closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```

برای این کار ها به کمک opencv ابتدا یک کرنل ساده که یک ماتریس است می سازیم وسپس با دستورات ساده زیر کار را انجام می دهیم.

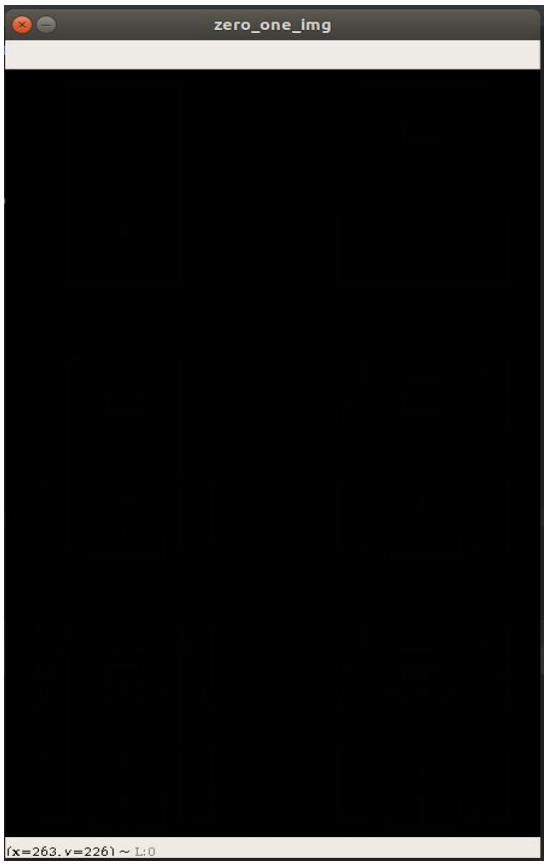
(7

ابتدا تصویر را به صورت grayscale می خوانیم.سپس همه پیکسل های سفید را سیاه می کنیم و پیکسل های سیاه را همه را مقدارشان را 1 قرار می دهیم برای این کار از یک ترشهولدینگ وارونه استفاده می کنیم.

```
th, zero_one_img = cv2.threshold(img1, 80, 1,
cv2.THRESH_BINARY_INV)
```

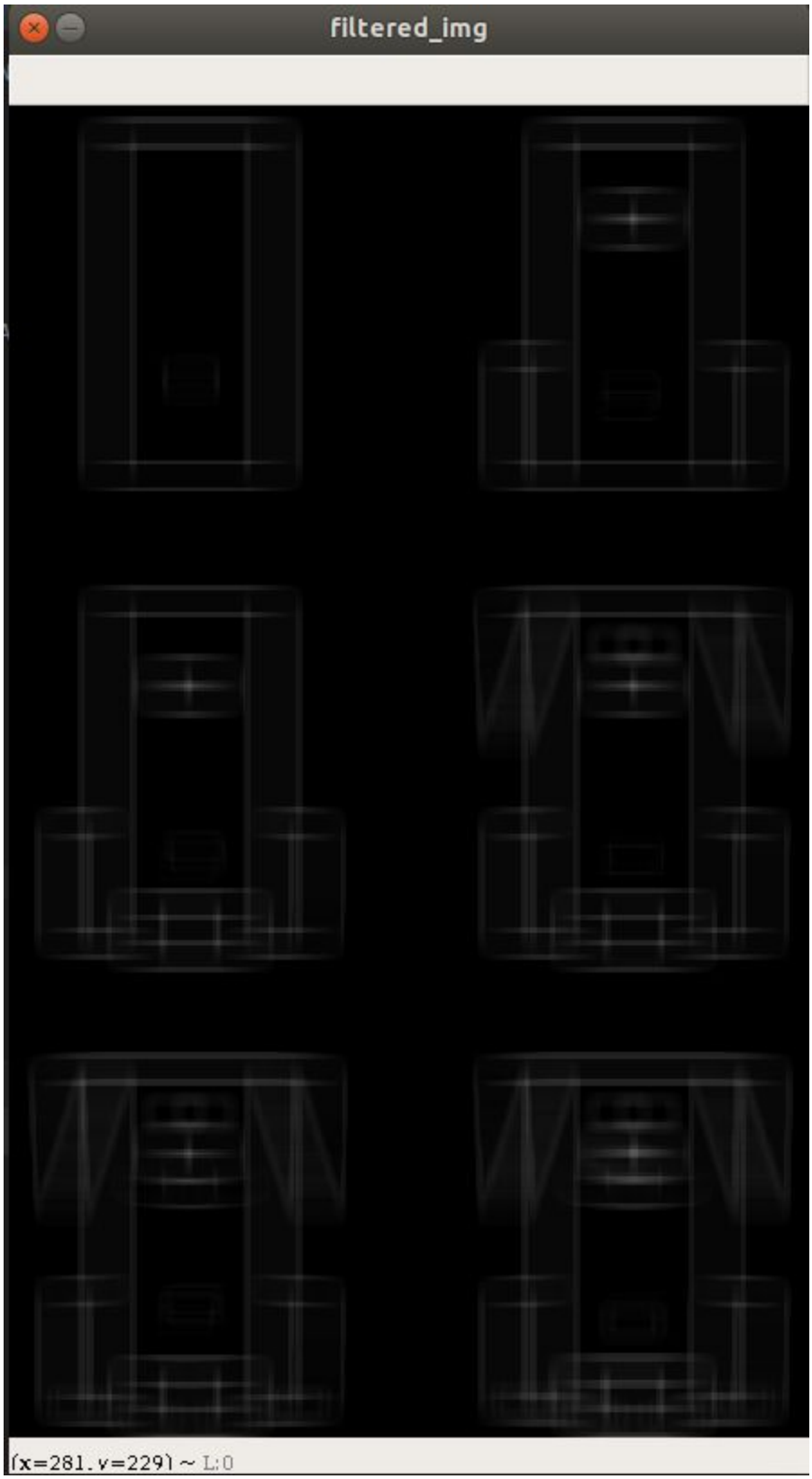
با این کار پیکسل هایی که مقدارشان بالای 80 است 0 می شوند و پیکسل هایی که مقدارشان زیر 80 است مقدار intensity ان ها 1 می شود.
عدد 80 با سعی و خطا به دست آمده است.

خروجی این مرحله تصویر زیر است.
که عملا در آن چیزی قابل رویت نیست زیرا پیکسل ها مقادیر 0 و 1 را دارند که در اسکیل 0 تا 255 سیاه به نظر می رسند.

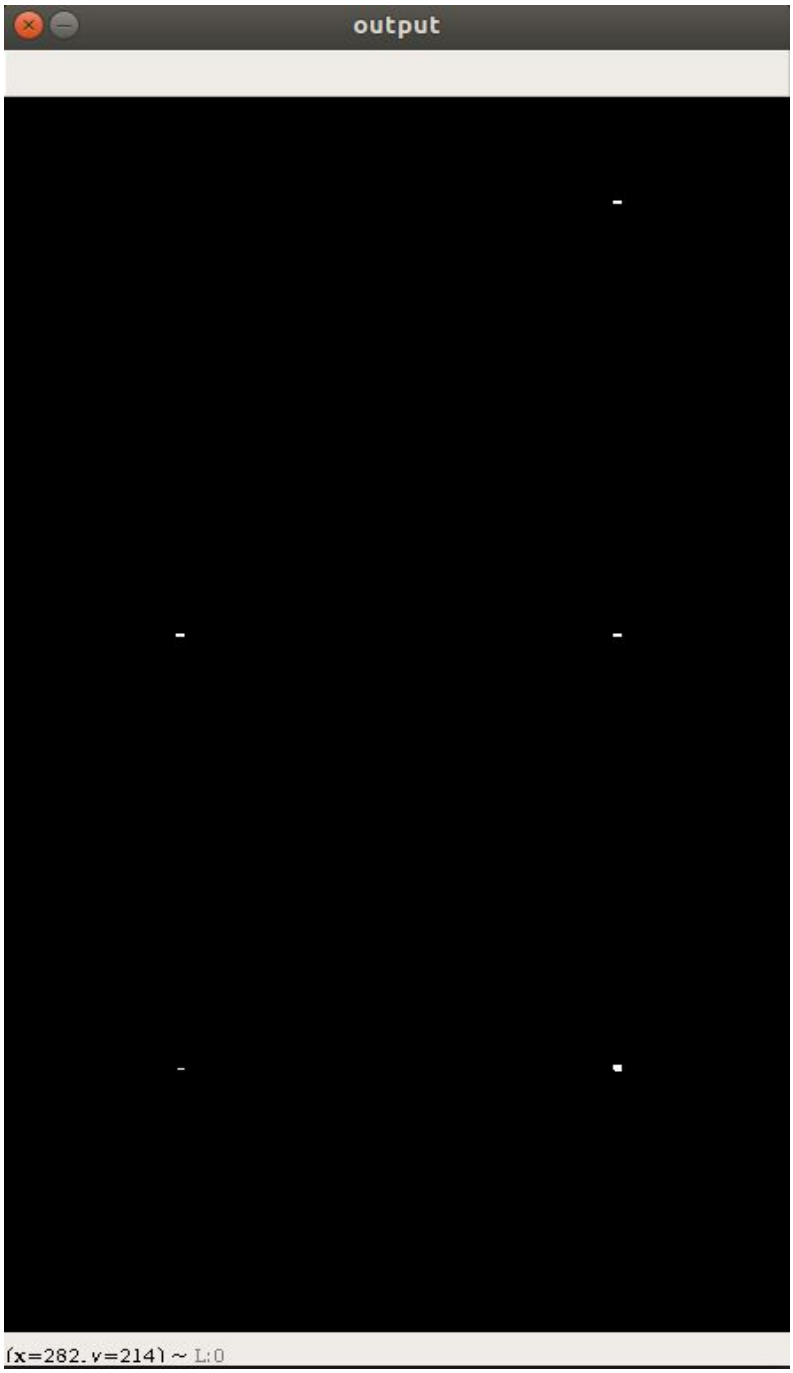


سپس یک ماتریس که اندازه آن برابر با مستطیل مشخص شده است که قرار است آن را تشخیص دهیم تشکیل می دهیم و همه درایه های مرزی آن را یک و درایه های درونی آن را 0 قرار می دهیم.
سپس تصویر حاصل از ترشهولدینگ را با این ماتریس کانوالو می کنیم نتیجه تصویر زیر می شود.

```
kernel = np.zeros((16,30))
kernel[0, :] = 1
kernel[15, :] = 1
kernel[:, 0] = 1
kernel[:, 29] = 1
filtered_img = cv2.filter2D(zero_one_img,-1,kernel)
```

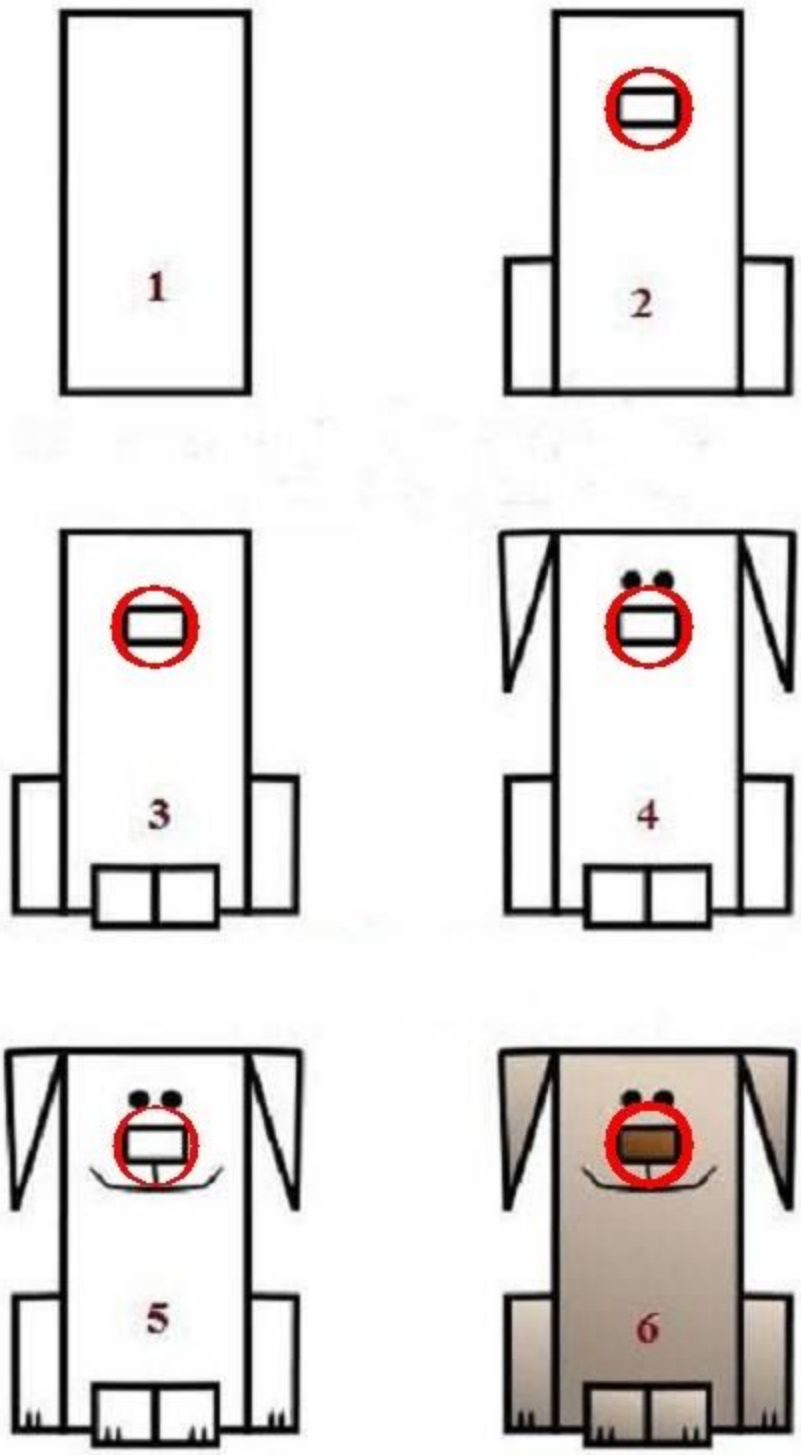

filtered_img
(x=281, y=229) ~ L:0

حالا کافی است در این تصویر یک ترشهولدینگ اعمال کنیم تا جاهای روشن تر را که همان خواسته مسئله ماست پیدا کنیم.پس از این کار خروجی تصویر زیر می شود



حالا صرفا کافیست که نقاط سفید مشخص شده را مکانشان را استخراج کنیم و در تصویر اصلی دور آن ها دایره بکشیم و این کار را به کمک یک تکه کد آماده در اینترنت انجام دادم که کار ساده ای است و از دستور cv2.circle استفاده می کند.
خروجی کل تصویر زیر می شود.

```
th2, output = cv2.threshold(filtered_img, 72, 255, cv2.THRESH_BINARY)
pixels = np.argwhere(output == 255)
p = pixels
m, n = pixels.shape
for i in range(m):
    c = pixels[i][0]
    p[i][0] =pixels[i][1]
    p[i][1] = c
for x in range(m):
    final_output = cv2.circle(img, tuple(p[x]), 20, (0, 0, 255))
```


1
2
3
4
5
6

سوال دوم

1)

برای خواندن ویدیو به کمک opencv یک حلقه object تعریف می کنیم و در یک حلقه while یک به یک فریم ها را می خوانیم.

ابتدا به کمک دستور زیر به webcam دسترسی پیدا می کنیم:

```
cap = cv2.VideoCapture(0)
```

سپس با دستور زیر فریم ها را یک یک می خوانیم

```
ret, frame = cap.read()
```

برای ذخیره کردن نیز از دستورات زیر استفاده می کنیم:

```
cap2 = cv2.VideoCapture(0)
t = cv2.VideoWriter_fourcc(*'mp4v')
output = cv2.VideoWriter("cam.mp4", t, 20.0, (640, 480))
```

و در انتها نیز :

```
cap2.release()
output.release()
```

(2

برای این قسمت کافی است ویدیو را فریم به فریم بخوانیم به صورت grayscale و از هر فریم تصویر زمینه را کم کنیم حالا کافی است یک ترشهولد ساده بزنیم تا بتوانیم ویدیو مورد نظر یعنی ویدویی که صرفا شامل اجسام متحرک است را داشته باشیم.

کد این قسمت در دو بخش زده شده است با اجرای کد

2_2_init.py

تصویر زمینه از روی ویدیو ذخیره می شود که بدست اوردن ان به این صورت است که با استفاده از یک مدین فیلتر ساده تصویر background را بدست می آوریم.

برای این کار ابتدا به صورت رندوم 25 فریم را انتخاب می کنیم.

```
# Randomly select 25 frames
frameIds = cap.get(cv2.CAP_PROP_FRAME_COUNT) * np.random.uniform(size=25)
```

و این فریم ها را در یک آرایه با نام frames ذخیره می کنیم.

```
# Store selected frames in an array
frames = []
for fid in frameIds:
    cap.set(cv2.CAP_PROP_POS_FRAMES, fid)
    ret, frame = cap.read()
    frames.append(frame)
```

سپس به کمک دستور ساده زیر میانه را بدست می آوریم

```
# Calculate the median along the time axis
medianFrame = np.median(frames, axis=0).astype(dtype=np.uint8)
```

سپس در کد 2-2 کار مورد نظر را انجام و ویدیو خواسته شده ساخته و با نام grayvideo.mp4 ذخیره و به دلیل حجم بالا لینک گوگل درایو ویدیو داده شده است

https://drive.google.com/open?id=1Rkzyq04f4ZiGGafmH9qmJt_0r1f3FJYf

البته راه های دیگری نیز برای استخراج بک گراند وجود دارد که می توان از یکی از آن ها نام برد.

به این صورت که در ویدیو های خلوت فریم اول را به عنوان بک گراند انتخاب کرده و سعی می کنیم object های متحرک را درون آن شناسایی و حذف کنیم.