

معیار کارکنه
آرستم با اعداد

$$\frac{8167 + 71575}{2} = 81271 \rightarrow 55$$

چون که این عدد خیلی بزرگه و این حلقه هم که است و در محاسبات را انحصارند هم و پاسخ
ممنوع است

آرستم باندی نباشد با اعداد 1 تا 10 هم کارکنه :

$$\frac{71575 + 2}{2} = 35788.5$$

$$\frac{0 \times 1 + 1 \times 2 + 2 \times 3 + 3 \times 4 + 4 \times 5}{1 + 2 + 3 + 4 + 5} = 2.375$$

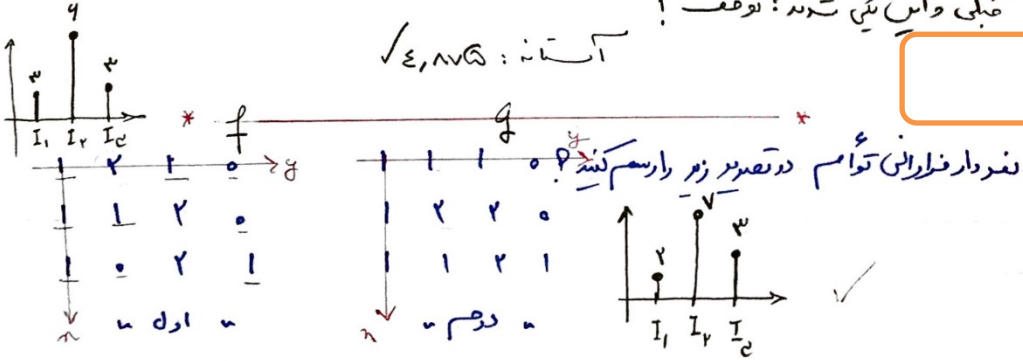
$$\frac{5 \times 2 + 6 \times 1 + 8 \times 2 + 9 \times 2}{2 + 1 + 2 + 2} = 71575$$

$$\frac{71575 + 2.375}{2} = 35788.9375 \rightarrow \text{cluster iteration}$$

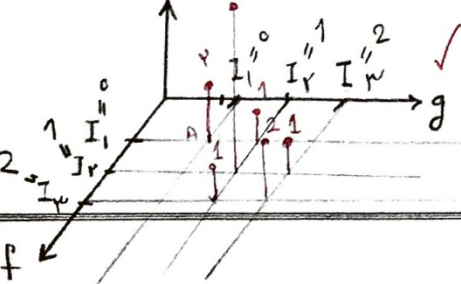
مجبوری و این یکی شدند : توقف !

آستانه : $\sqrt{35788.9375}$

سوال 2-



در تصویر که انطباق ندارند می توان همسگرام مشترک تشکیل داد و محفید در تصویر حقیقه با هم



انطباق دارند. (12)

11	21	11	00	f
11	12	22	00	f
11	01	22	11	g

g=0, f=0	x2	g=2, f=1	x1
g=1, f=0	x1	g=0, f=2	x0
g=2, f=0	x0	g=1, f=2	x1
g=0, f=1	x0	g=2, f=2	x2
g=1, f=1	x0		

صفحه

سوال 3-

تبدیل دایره ای هاف برای این کار لازم است.

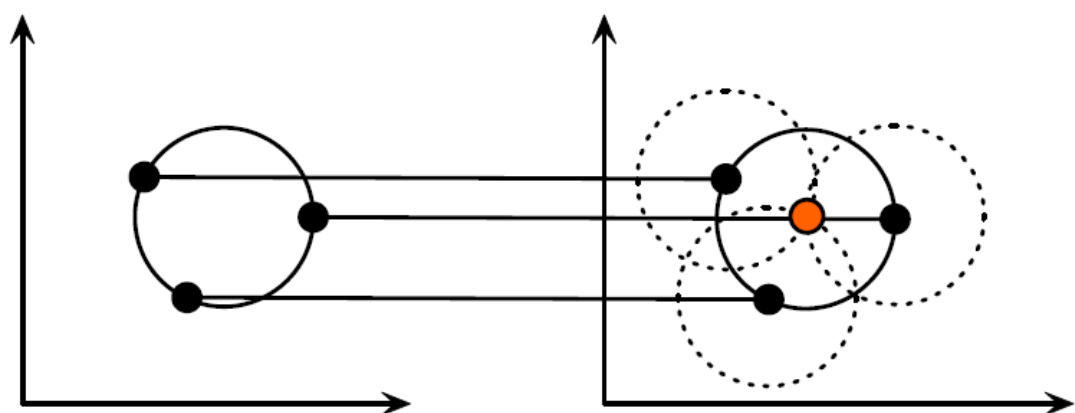
در صورت مشخص بودن تعداد نقاطی که روی محیط دایره قرار می گیرند ، می توان از تبدیل Hough برای تعیین پارامترهای یک دایره استفاده کرد. یک دایره با شعاع R و مرکز (a, b) را می توان با معادلات پارامتری توصیف کرد:

$$x = a + R \cos(\theta)$$

$$y = b + R \sin(\theta)$$

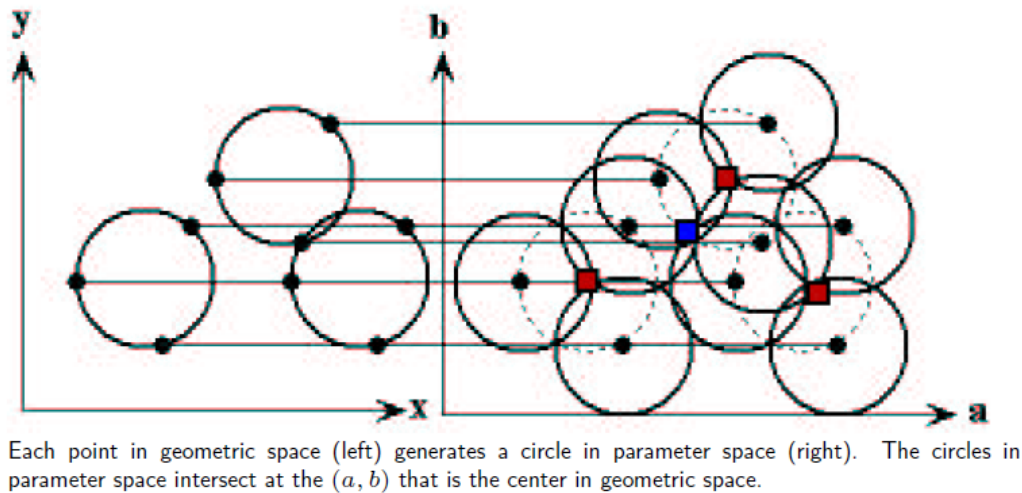
وقتی زاویه در محدوده 360 درجه جابجا می شود، نقاط (x, y) محیط یک دایره را ترسیم می کنند. اگر تصویری حاوی نقاط زیادی باشد که بعضی از آنها در محیط دایره ها قرار گیرند، وظیفه برنامه پیدا کردن سه گانه های پارامتر (R, b, a) برای توصیف هر دایره است. این موضوع که فضای پارامتر سه بعدی است، اجرای مستقیم روش Hough را در حافظه و زمان رایانه، گران تر می کند.

اگر دایره های تصویر شعاع R مشخص داشته باشند ، می توان جستجو را به 2 بعد کاهش داد. هدف یافتن مختصات (b, a) مراکز است.



Each point in geometric space (left) generates a circle in parameter space (right). The circles in parameter space intersect at the (a, b) that is the center in geometric space.

هر نقطه در فضای هندسی (چپ) یک دایره در فضای پارامتر (راست) ایجاد می کند. دایره ها در فضای پارامتر در (a , b) که مرکز آن در فضای هندسی است ، تلاقی می کنند.



هر نقطه در فضای هندسی (چپ) یک دایره در فضای پارامتر (راست) ایجاد می کند. دایره ها در فضای پارامتر در (a , b) که مرکز آن در فضای هندسی است ، تلاقی می کنند.

معادله یک دایره به ترتیب زیر است:

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

مراحل زیر برای شناسایی حلقه های موجود در تصویر دنبال می شود:

1. با کمک ردیاب های لبه (Canny) لبه ها را در تصویر داده شده پیدا میکنیم.
2. برای تشخیص دایره ها در یک تصویر ، یک آستانه برای حداکثر و حداقل مقدار شعاع تعیین می کنیم.
3. شواهد در یک آرایه جمع کننده سه بعدی برای حضور دایره هایی با مراکز و شعاع های مختلف جمع آوری می شود.

از HoughCircles در OpenCV برای تشخیص دایره های موجود در تصویر استفاده می شود. این تابع پارامترهای زیر را می گیرد:

- image: تصویر ورودی.
 - method: روش تشخیص.
 - dp: نسبت معکوس وضوح accumulator و وضوح تصویر.
 - mindst: حداقل فاصله بین مراکز حلقه های شناسایی شده.
 - param_1 و param_2: این پارامترهای خاص روش هستند.
 - min_Radius: حداقل شعاع دایره ای که باید شناسایی شود.
 - max_Radius: حداکثر شعاع قابل شناسایی است.
- تابع HoughCircles قابلیت شناسایی لبه Canny را دارد، بنابراین تشخیص لبه ها به طور جداگانه قبل از آن لازم نیست.

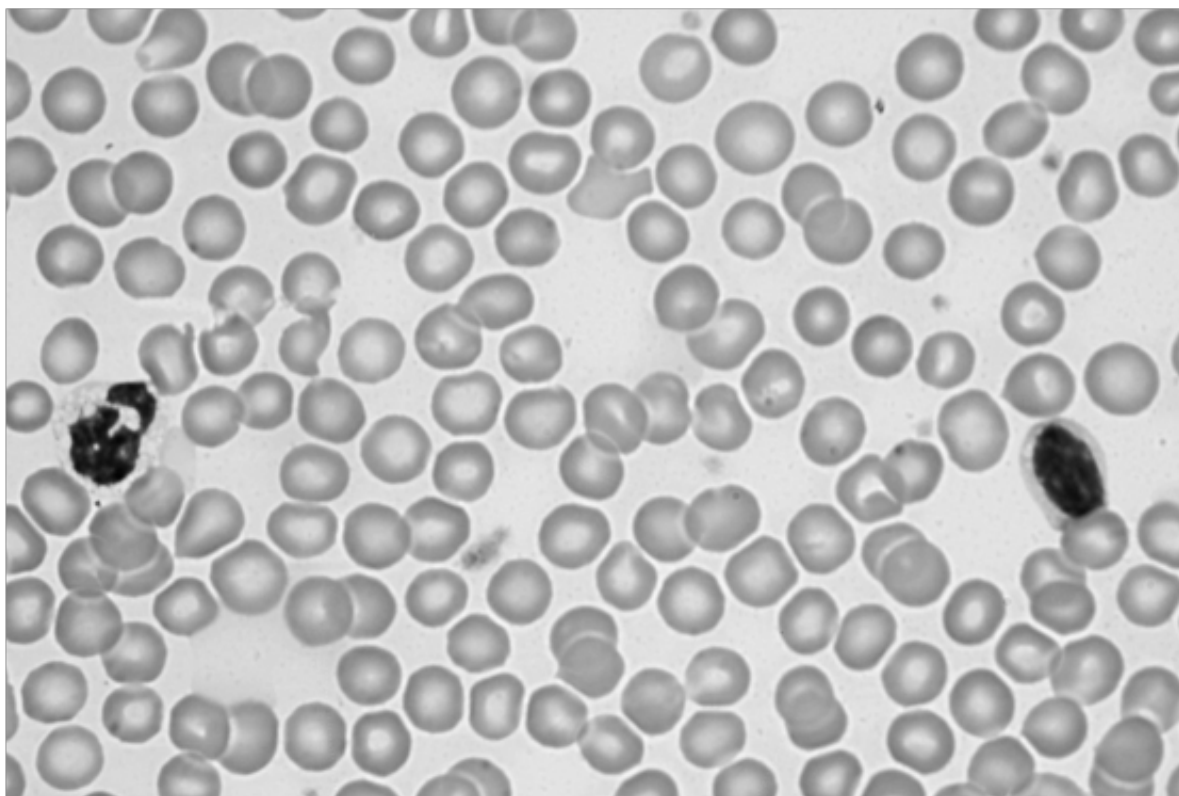
```
void cv::HoughCircles ( InputArray  image,
                        OutputArray circles,
                        int          method,
                        double       dp,
                        double       minDist,
                        double       param1 = 100 ,
                        double       param2 = 100 ,
                        int          minRadius = 0 ,
                        int          maxRadius = 0
                      )
```

Enumerator	
HOUGH_STANDARD Python: cv.HOUGH_STANDARD	classical or standard Hough transform. Every line is represented by two floating-point numbers (ρ, θ) , where ρ is a distance between $(0,0)$ point and the line, and θ is the angle between x-axis and the normal to the line. Thus, the matrix must be (the created sequence will be) of CV_32FC2 type
HOUGH_PROBABILISTIC Python: cv.HOUGH_PROBABILISTIC	probabilistic Hough transform (more efficient in case if the picture contains a few long linear segments). It returns line segments rather than the whole line. Each segment is represented by starting and ending points, and the matrix must be (the created sequence will be) of the CV_32SC4 type.
HOUGH_MULTI_SCALE Python: cv.HOUGH_MULTI_SCALE	multi-scale variant of the classical Hough transform. The lines are encoded the same way as HOUGH_STANDARD.
HOUGH_GRADIENT Python: cv.HOUGH_GRADIENT	basically <i>21HT</i> , described in [287]
HOUGH_GRADIENT_ALT Python: cv.HOUGH_GRADIENT_ALT	variation of HOUGH_GRADIENT to get better accuracy

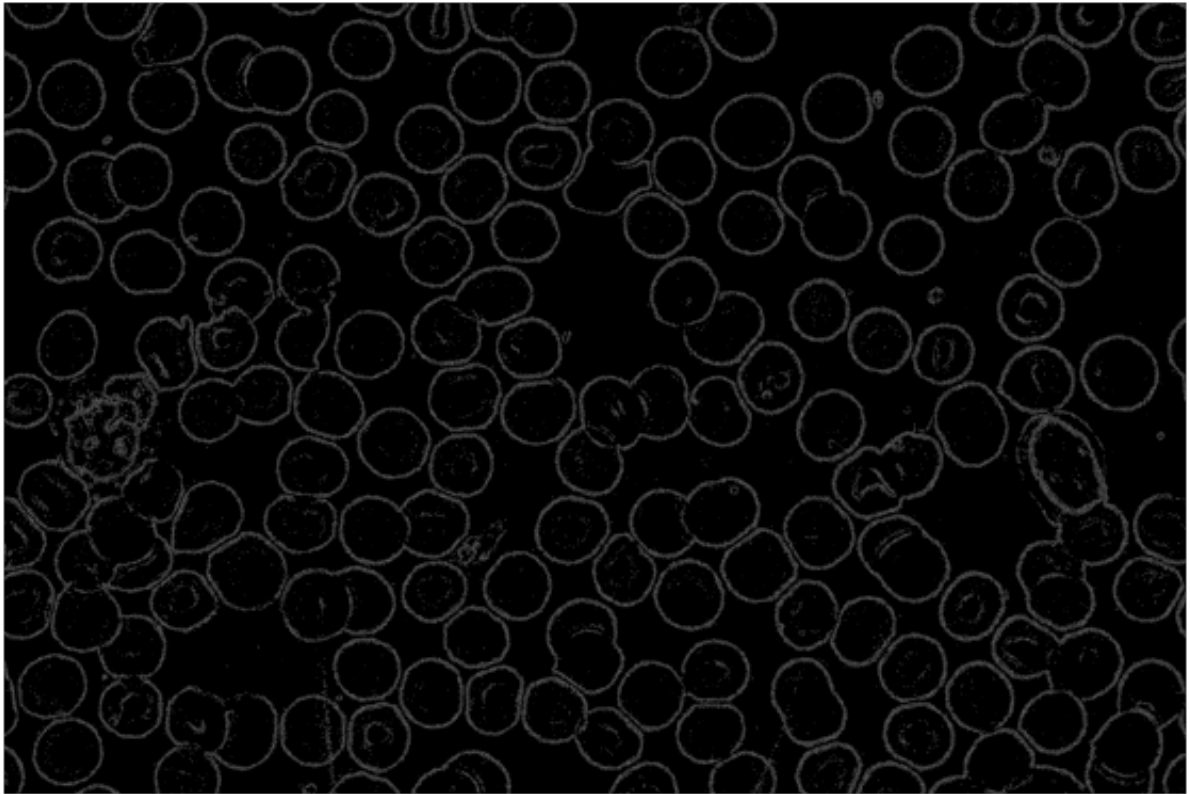
Canny edge detector یک عملگر تشخیص لبه است که با استفاده از یک الگوریتم چند مرحله ای طیف وسیعی از لبه ها را در تصاویر تشخیص می دهد. این کاری توسط جان اف کانی در سال 1986 ساخته شد. الگوریتم تشخیص لبه Canny از 5 مرحله تشکیل شده است:

1. کاهش نویز؛
2. محاسبه گرادیان
3. سرکوب غیر حداکثر Non-maximum suppression
4. آستانه دو برابر Double threshold
5. ردیابی لبه توسط Hysteresis.

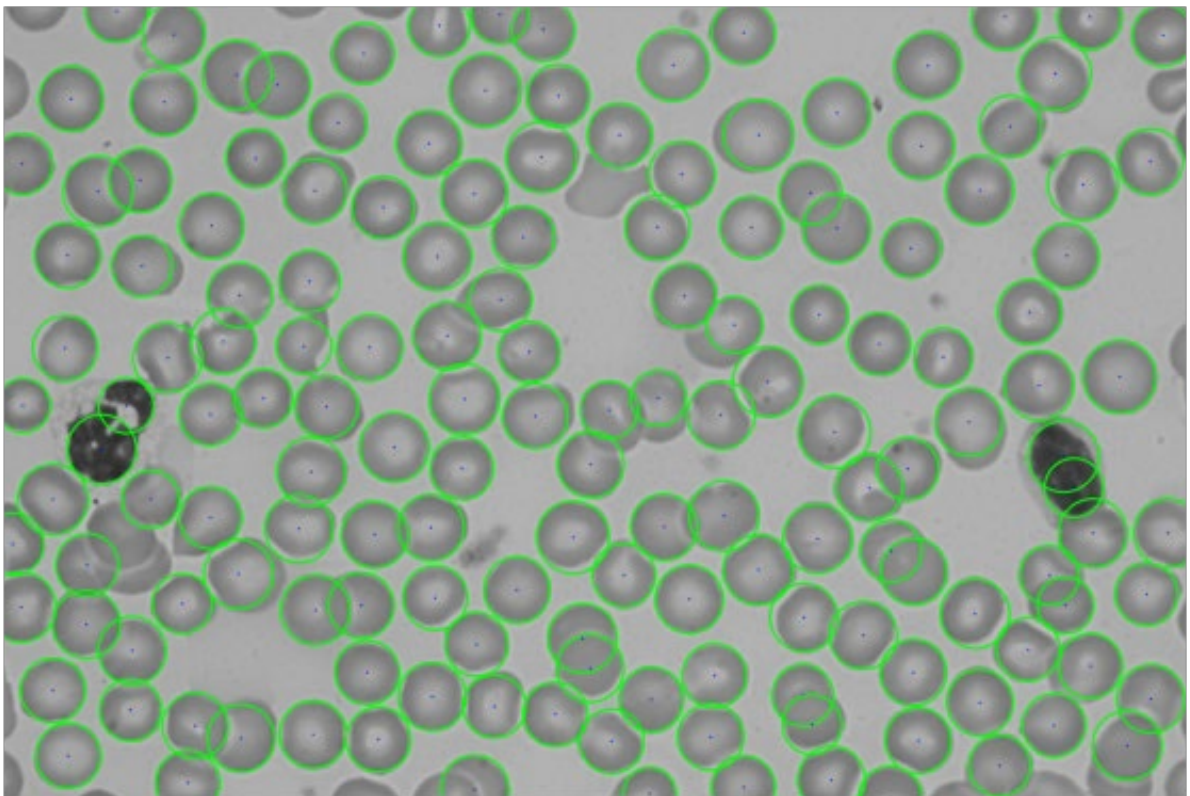
Original Image



Edges Image



detected circles



سوال 4-

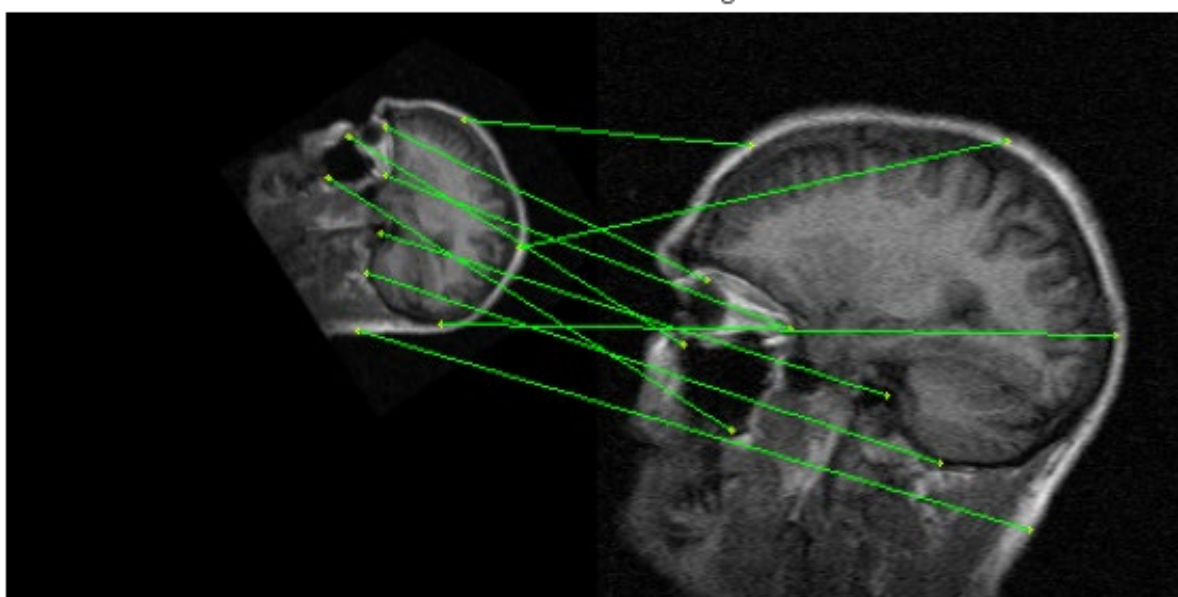
برای این سوال یک رابط کاربری نوشتم که کاربر ابتدا نقطه ای را با کلید کردن از تصویر 1 انتخاب میکند سپس باید نقطه متناظر را با کلید کردن از روی تصویر 2 انتخاب کند. برنامه را طوری نوشتم که اگر کاربر در فرآیند انتخاب دچار اشتباه شد و ترتیب را رعایت نکرد، به او اخطار دهد و بگوید که اکنون باید از کدام تصویر انتخاب کند.

خروجی اخطار:

`⌂_⌂ select from left image - NOT RIGHT !!`

همچنین هر نقطه را از تصویر به تصویر دیگر با خطی وصل میکنم که کاربر در جریان انتخاب هایش باشد.

Feature Selected image



سپس نقاط را در ماتریس های بنام features ذخیره میکنم و با استفاده از توابع OpenCV تبدیل را محاسبه میکنم، سپس تبدیل را روی تصویر اعمال میکنم.

خروجی برنامه برای تبدیل:

Conversion matrix values :

```
[[ 2.63195272e+00  3.76099201e+00 -5.22930777e+02]
 [-2.41489176e+00  2.37199496e+00  5.19551954e+02]
 [ 6.30450041e-03  1.25577125e-03  1.00000000e+00]]
```

خروجی تبدیل اعمال شده:

Registered image

