

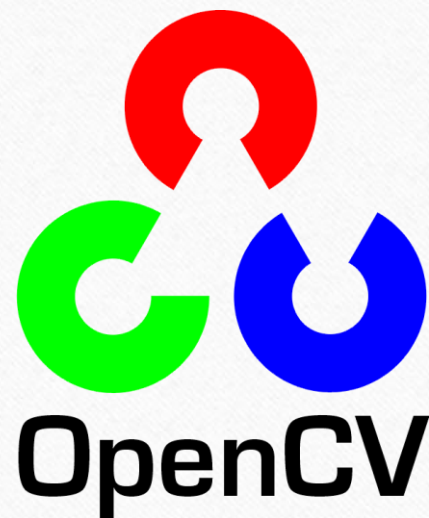
دوره آموزشی بینایی ماشین کاربردی

آکادمی رباتک - آزمایشگاه تعامل انسان و ربات

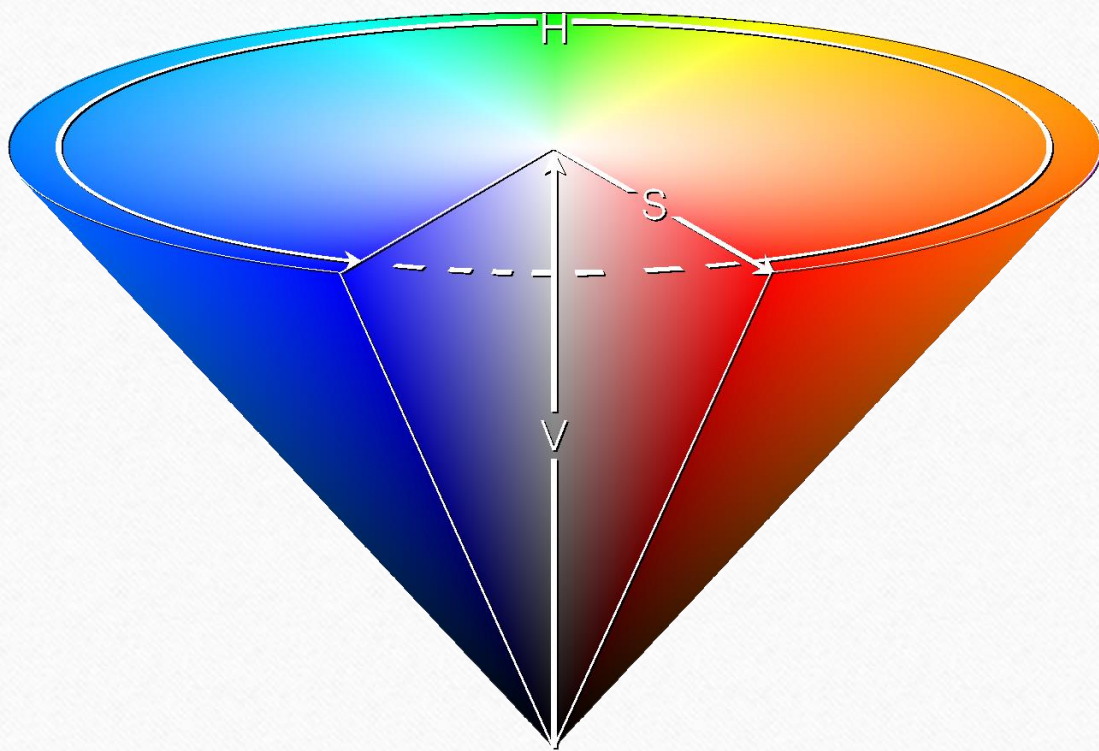
جلسه 4 - الگوریتم های تشخیص ویژگی



رباتک



آنچه گذشت ؟ !



آشنایی با الگوریتم های تشخیص ویژگی

با کمک **Keypoint** ها و **Descriptor** ها



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ \vdots \\ x_{n-3} \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} \quad \dots \quad \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \vdots \\ \vdots \\ u_{n-3} \\ u_{n-2} \\ u_{n-1} \\ u_n \end{bmatrix}$$

Descriptor چیست ؟

✓ توصیفی از نقاط کلیدی

✓ از گرادیان نقاط در اطراف نقطه کلیدی میتوان برای ایجاد descriptor استفاده کرد.

✓ معمولا به صورت برداری از اعداد ارایه می شود.

Keypoint چیست ؟

✓ نقاط کلیدی تصویر

نقطه 3



نقطه 1



نقطه 2



✓ مکان هایی که تغییر گرادیان در اطراف آنها زیاد است.

الگوریتم SIFT [Scale Invariant Feature Transform]



مستقل از Scale و rotation و نور و نقطه مشاهده است.



David Lowe

 FOLLOW

Computer Science Dept., [University of British Columbia](#)
Verified email at cs.ubc.ca - [Homepage](#)
[Computer Vision](#) [Object Recognition](#)

TITLE

CITED BY

YEAR

[Distinctive image features from scale-invariant keypoints](#)

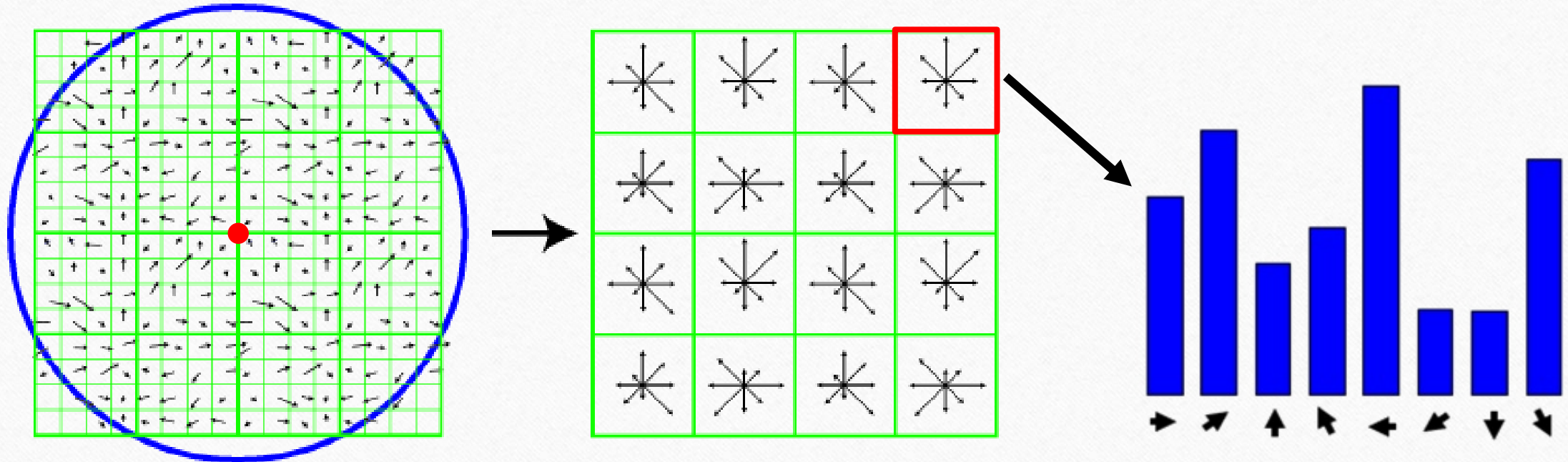
53951

2004

DG Lowe

International journal of computer vision 60 (2), 91-110

SIFT Descriptor



برای تک تک بلوک ها، این نمودار استخراج می شود و در نهایت یک بردار $16 * 8$ یعنی 128 تایی از اعداد تولید می شود.

دستور 32 : تعریف یک Object از SIFT

`cv2.xfeatures2d.SIFT_create`



یک Object از کلاس
SIFT



```
sift = cv2.xfeatures2d.SIFT_create()
```

مثال :

دستور 32 : پیدا کردن Keypoint ها

تصویر

mask

sift.detect

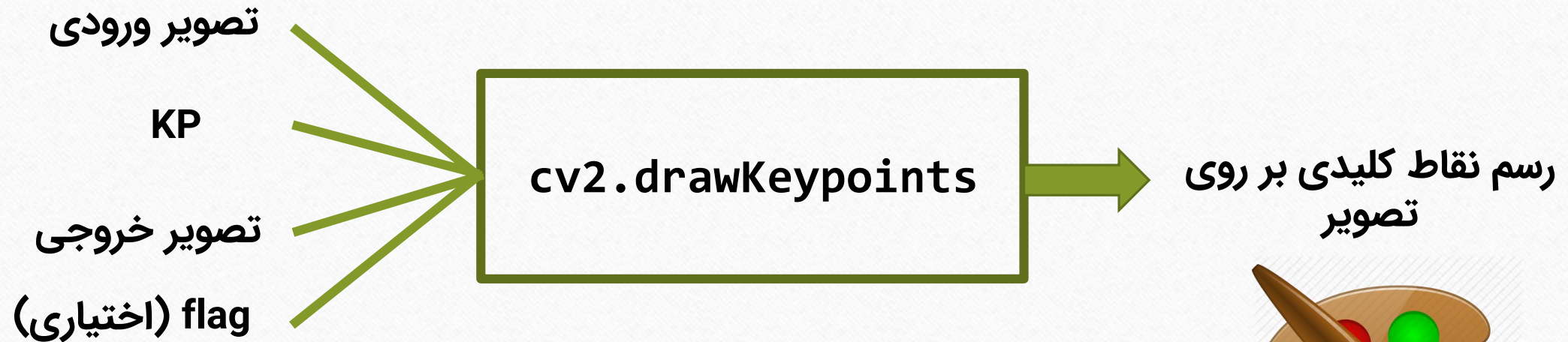
نقاط کلیدی تصویر



```
kp = sift.detect(img, None)
```

مثال :

دستور 32 : رسم KeyPoint ها



`flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS`

مثال :

`cv2.drawKeypoints(img, kp, img)`

دستور 32 : پیدا کردن KeyPoint ها و descriptor ها

تصویر ورودی

نقاط کلیدی تصویر

mask

sift.detectAndCompute

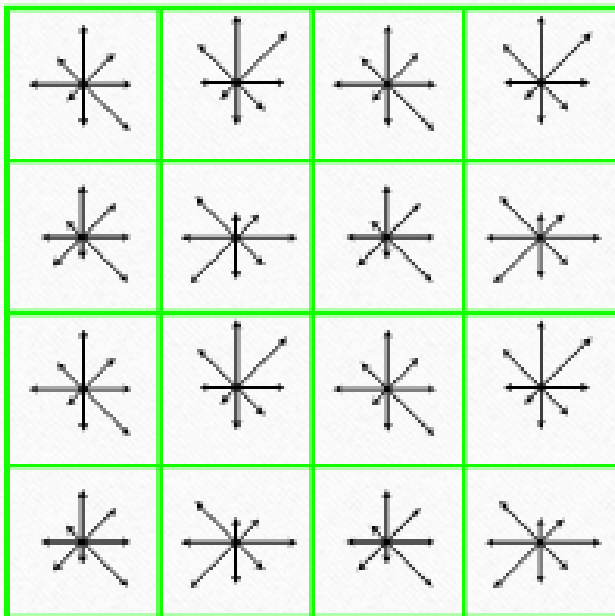
Descriptor نقاط



مثال :

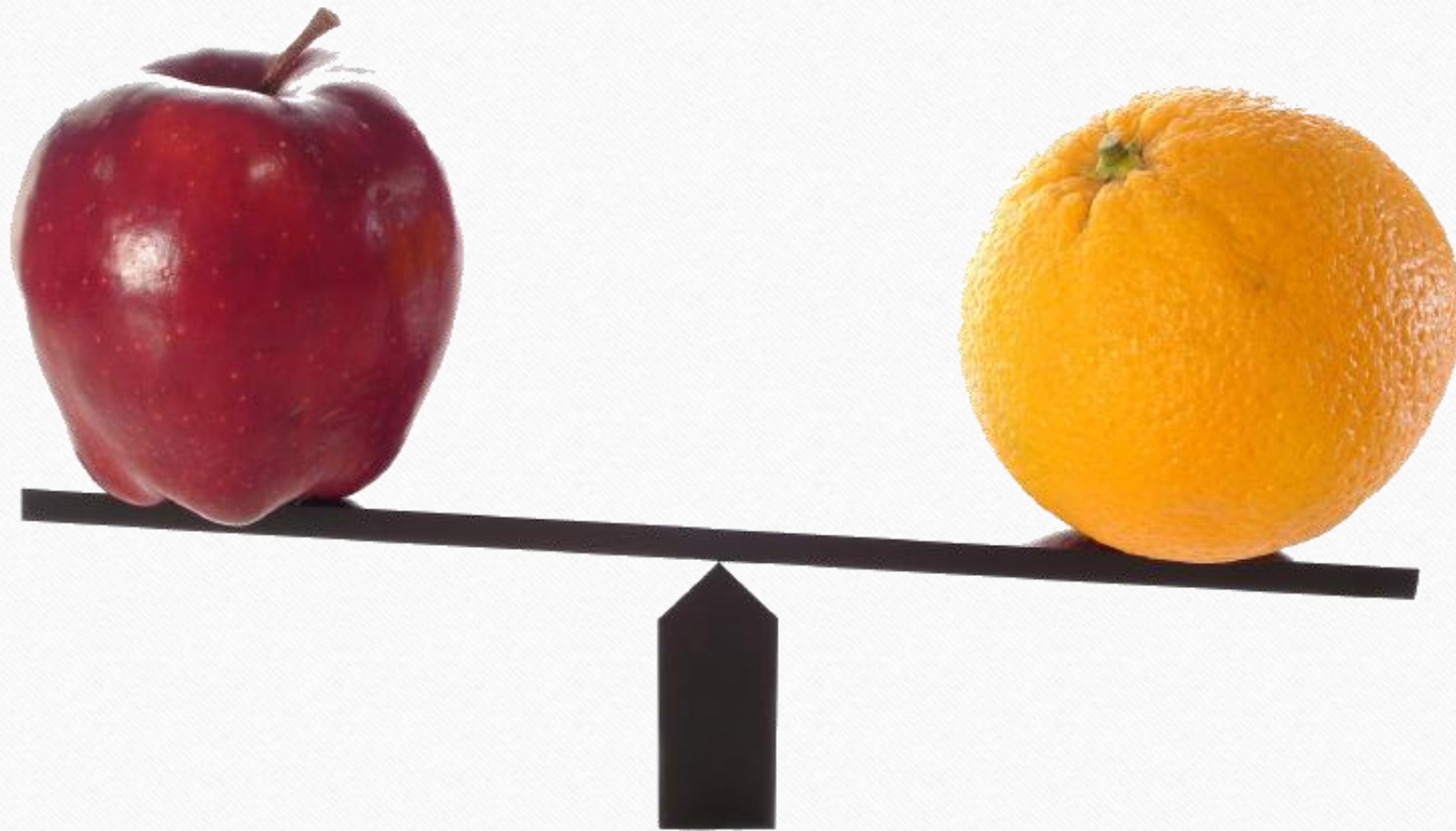
```
kp, des= sift.detectAndCompute(img, None)
```

توصیف گر (Descriptor) چه شکلی دارد؟



(*Number of Keypoints* * 128)

مقایسه بین descriptor ها و match کردن آنها



الگوریتم BFMatcher :

یک descriptor از تصویر اول را می گیرد و با تمام descriptor های تصویر دوم مقایسه می کند و نزدیکترین descriptor را بر میگرداند.

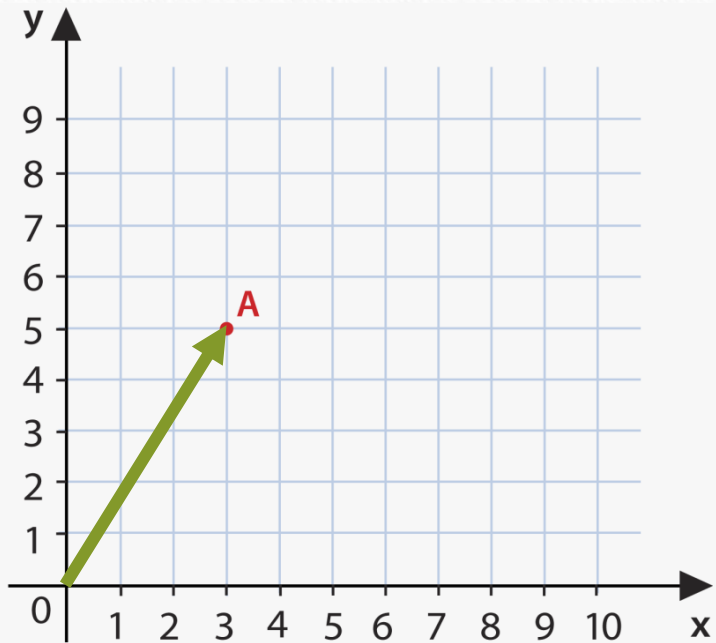
مقایسه بین اعداد با روش های مختلفی از جمله NORM_L1 و NORM_L2 ... انجام می گیرد.

در کنار FLANN based matcher از الگوریتم های اصلی مقایسه می باشد.



یادآوری NORM در ریاضی

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ \dots \\ x_n \end{bmatrix} \longrightarrow \text{نرم بردار } x = \text{طول بردار } x = \|X\|_p$$



$$\|X\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}$$

نرم مینکوفسکی

$$\|X\|_2 = \sqrt{(3 - 0)^2 + (5 - 0)^2} \quad \text{مثال :}$$

NORM L1 : $\|X\|_1 = |x_1| + |x_2| + \dots + |x_n|$

نرم منهتن یا city-block

NORM L2 : $\|X\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$

نرم اقلیدسی

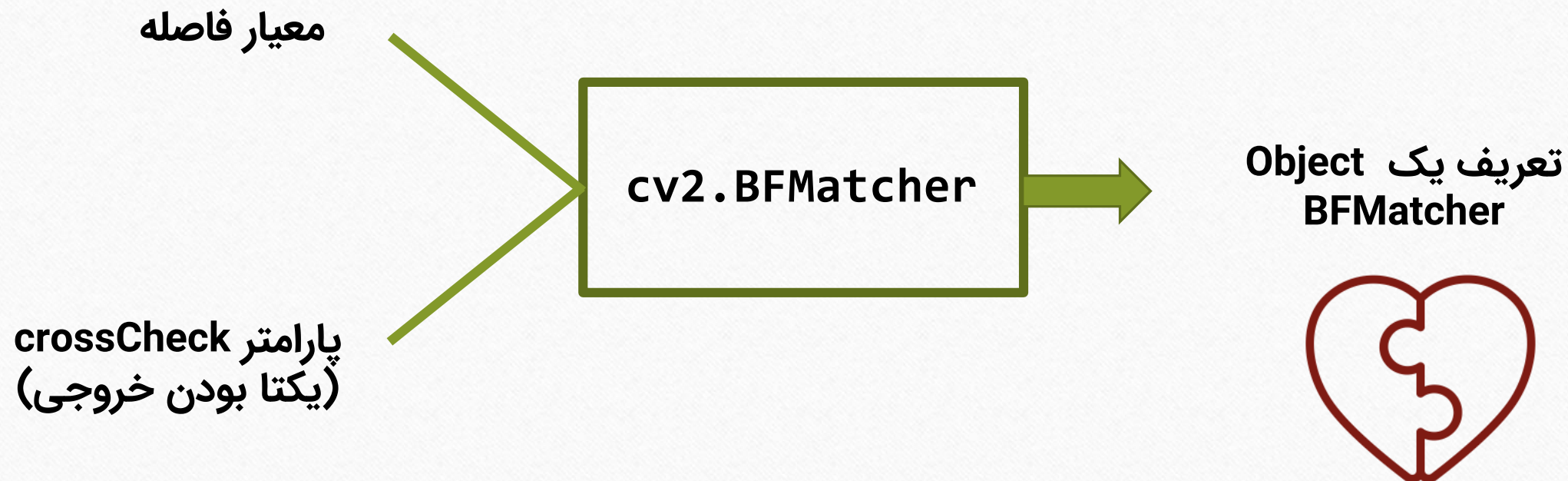
NORM L0 : $\|X\|_0 = \text{Numer of nonzero elements}$

نرم Hamming

NORM L ∞ : $\|X\|_\infty = \max(x_i)$

نرم بی نهایت

دستور 32 : تعریف یک Object از BFMatcher



مثال :

```
bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)
```

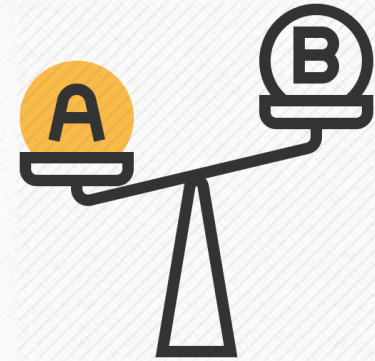
دستور 32 : مقایسه بین descriptor ها

اول descriptor

دوم descriptor

bf.match

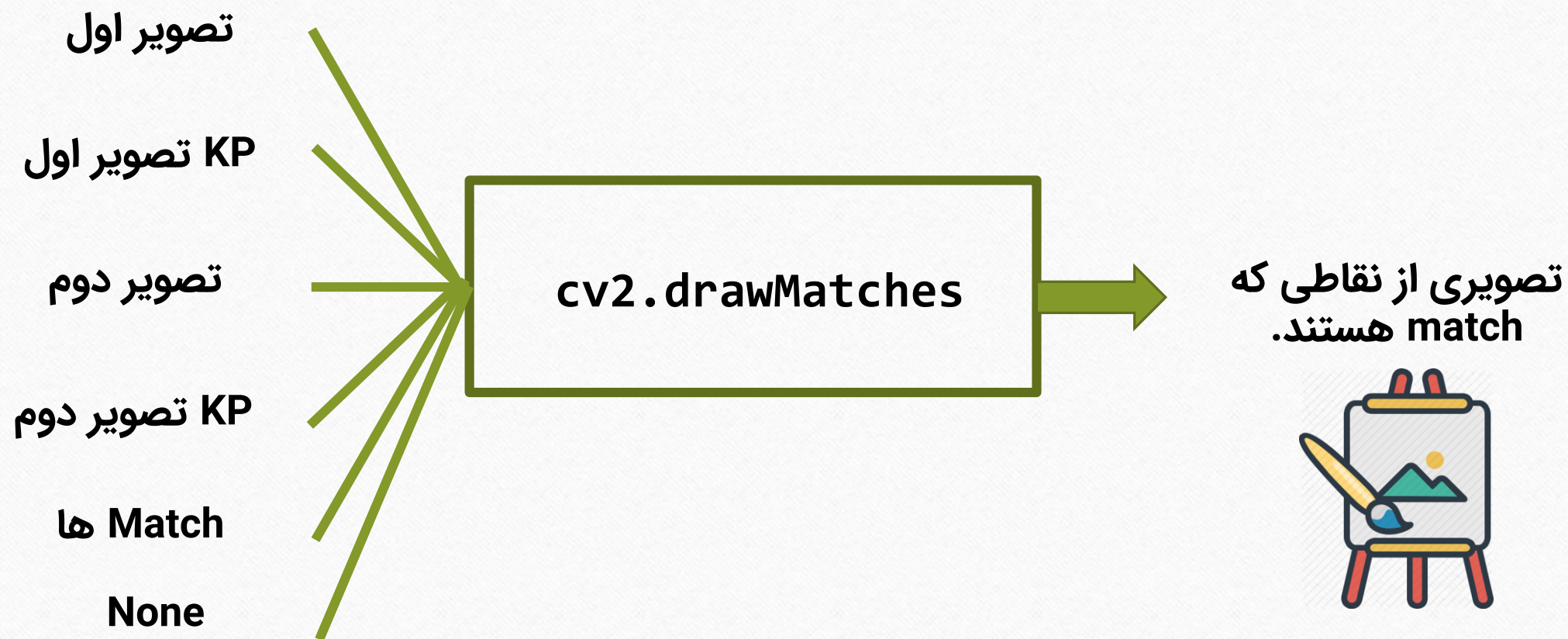
بهترین descriptor هایی
که با هم match هستند.



مثال :

```
match = bf.match(des1, des2)
```


دستور 32 : رسم نقاطی که match هستند



مثال :

```
matching_result = cv2.drawMatches(book1, kp1, book2, kp2, match, None)
```

اشکالات را چگونه رفع کنیم ؟

روش 2 : ratio test

روش پیشنهادی lowe در مقاله
SIFT



روش 1 : sort کردن

Sort کردن نقاط match بر حسب
فاصله



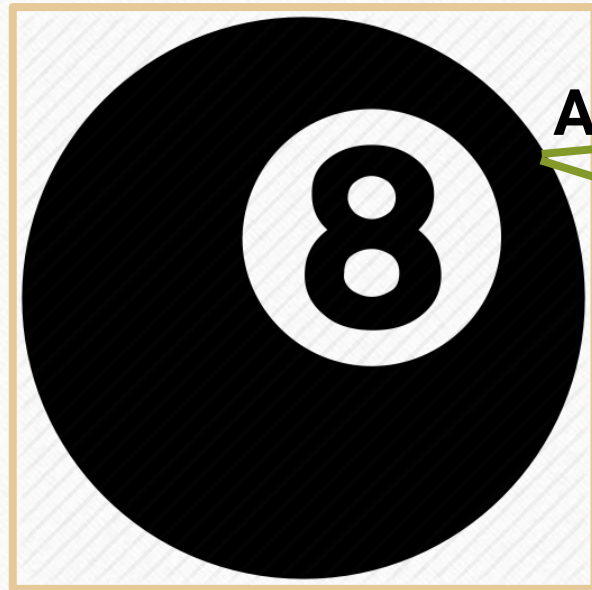
روش 1 : sort کردن

```
match = sorted(match, key = lambda x:x.distance)
```

تنظیم معیار بر حسب فاصله

```
matching_result = cv2.drawMatches(book1, kp1, book2, kp2, match[:40], None)
```

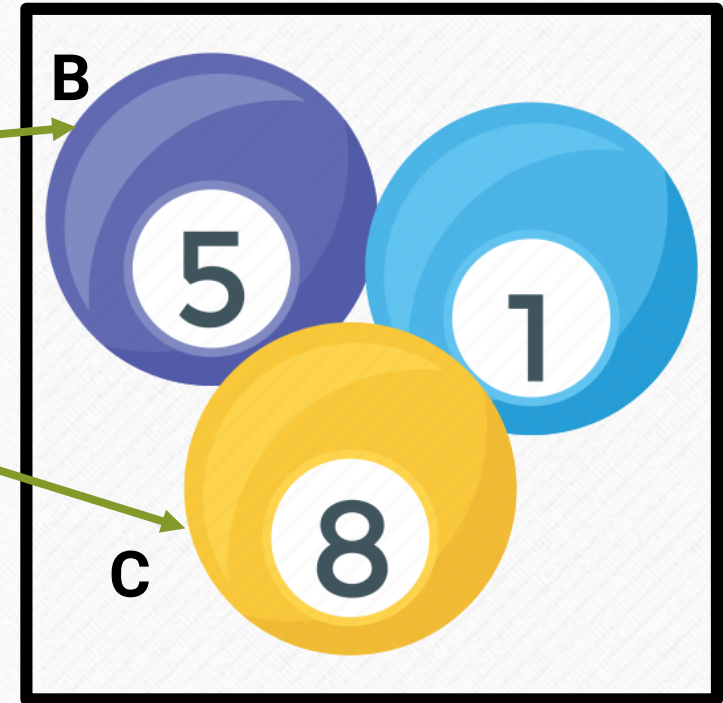

قانون lowe Ratio test



تصویر پایه

$d1 = 120$

$d2 = 123$



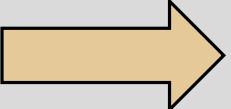
تصویر اصلی

سوال : آیا میتوان با قطعیت گفت نقطه A متناظر با نقطه B است ؟

آزمون **lowe Ratio**



اگر دو feature فاصله تقریباً یکسانی با یک feature دیگر داشته باشند، **احتمالاً** گزینه های مناسبی نیستند.

$\frac{d_1}{d_2} < 0.7$  A و B متناظر هستند.

دستور 32 : مقایسه بین descriptor ها با KNN

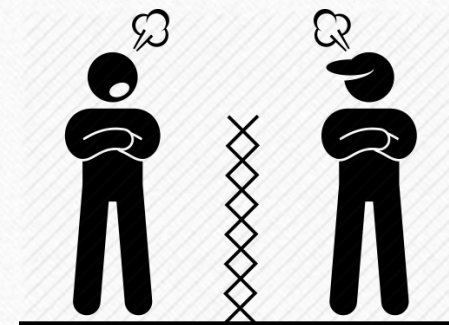
اول descriptor

دوم descriptor

پارامتر K

bf.knnMatch

K تا از بهترین descriptor
برای هر descriptor



نکته : CrossCheck باید حتما false شود.

مثال :

```
matches = bf.knnMatch(des1,des2, k=2)
```


حل یک مثال:

تشخیص حضور یک جسم به کمک الگوریتم SIFT

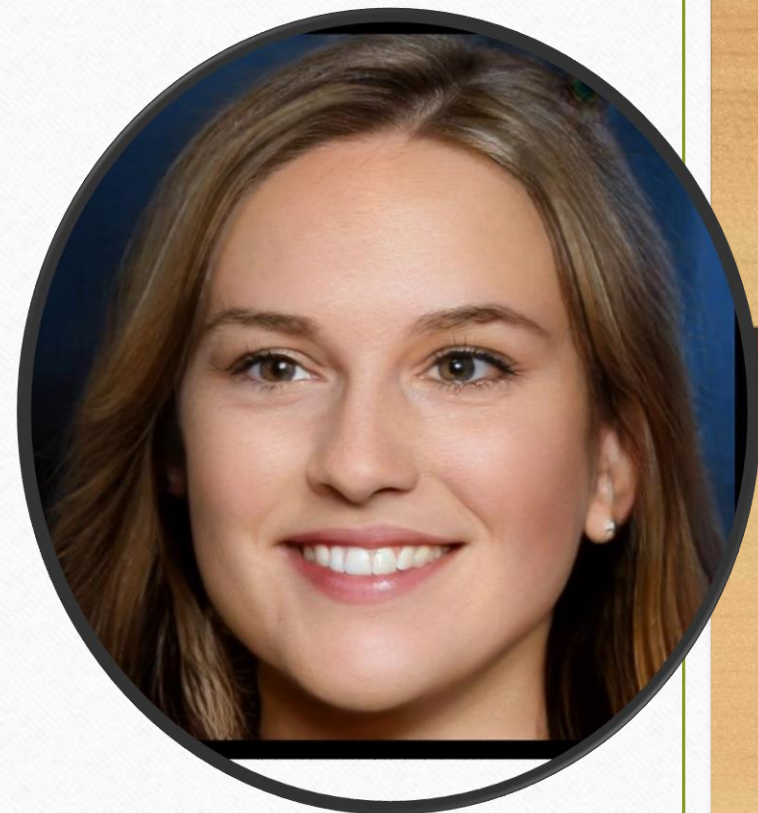
شخصیت هفته : Ian Goodflew



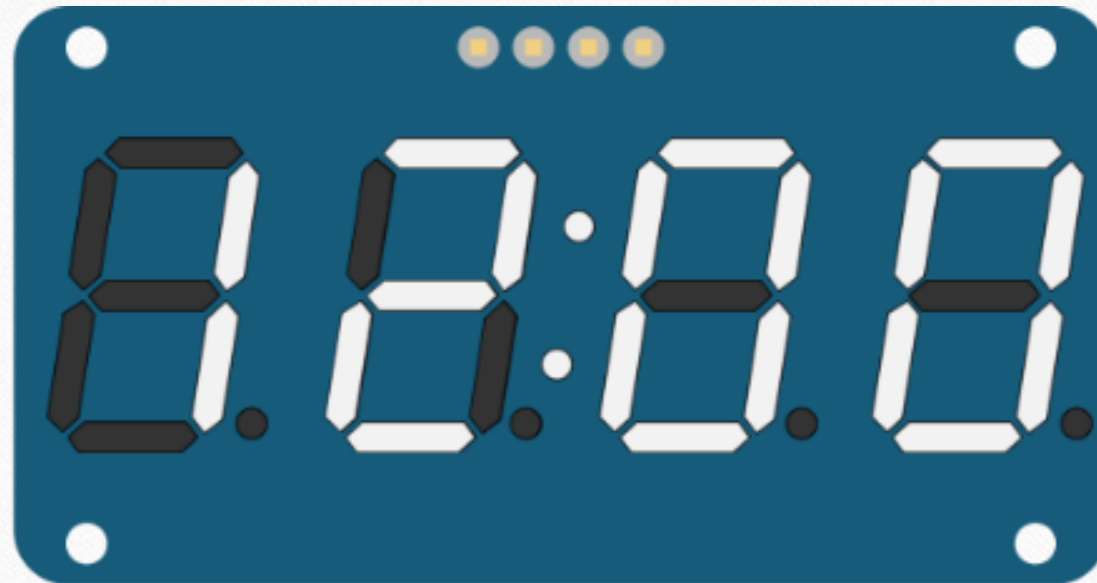
نویسنده اول مقاله معروف GAN

جزو 100 متفکر برتر دنیا در 2019

OpenAI , Apple , Google



مثال : تشخیص اعداد به کمک SIFT



1



بردار ویژگی 1

2



بردار ویژگی 2

⋮

9



بردار ویژگی 9

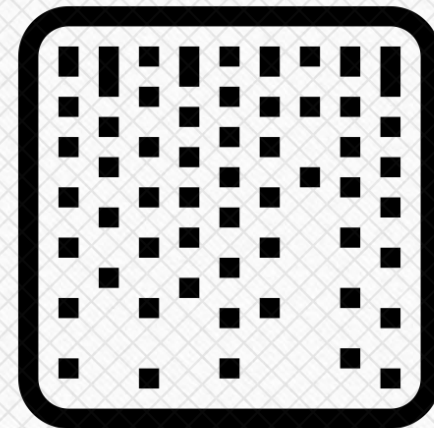
10



بردار ویژگی 10

مرحله 1 : تشکیل ماتریس ویژگی

تشکیل ماتریس ویژگی



مرحله 2 : جداسازی اعداد از تصویر اصلی

خواندن تصویر و تبدیل به Gray

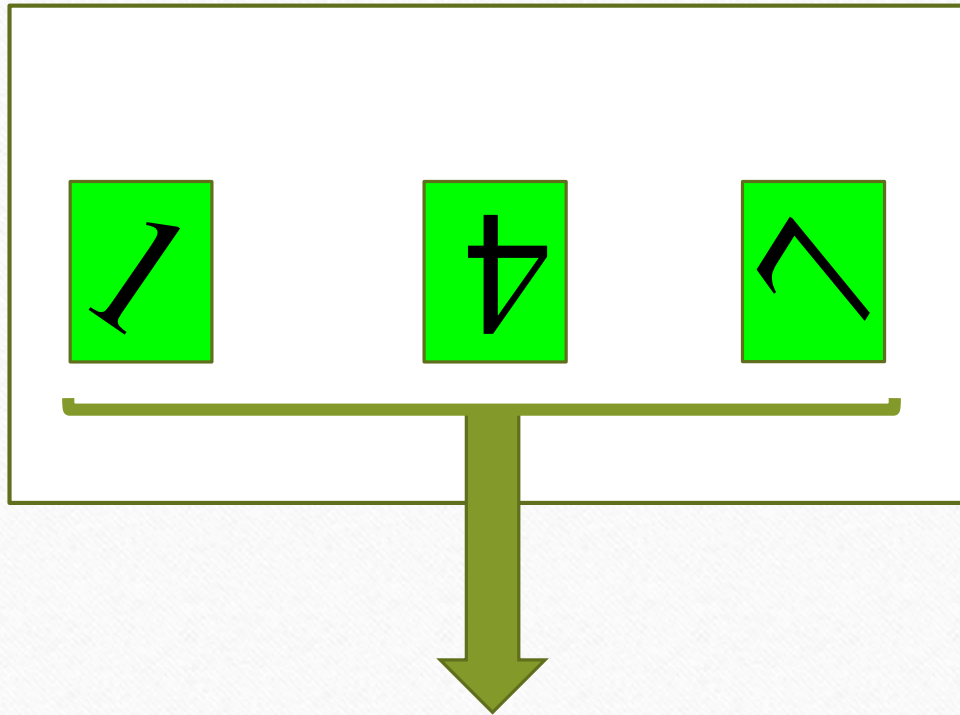
نويز گيري آستانه گذاري inverse

محاسبه کانتور

جداسازی ناحیه مربوط به رقم



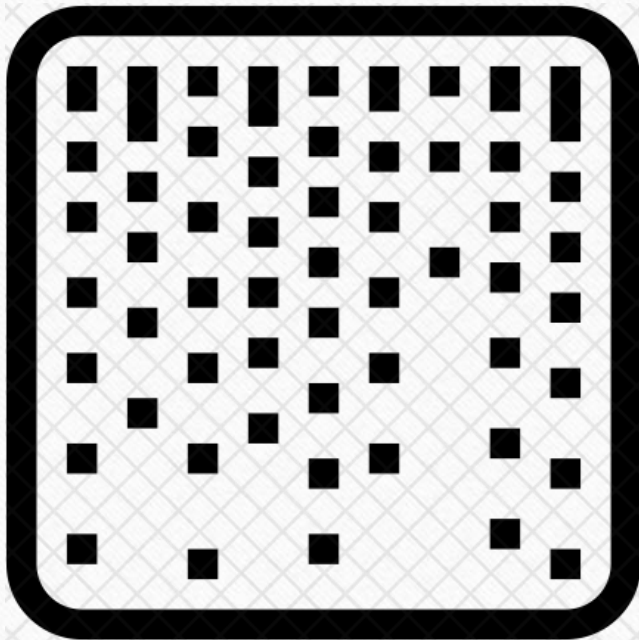
مرحله 3 : محاسبه بردار ویژگی اعداد جدا شده



محاسبه بردار ویژگی برای هر کدام از این اعداد

مرحله 3 : مقایسه بردار ویژگی با ماتریس ویژگی

ماتریس ویژگی



Feature vector