

[Open in app](#)

Search



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



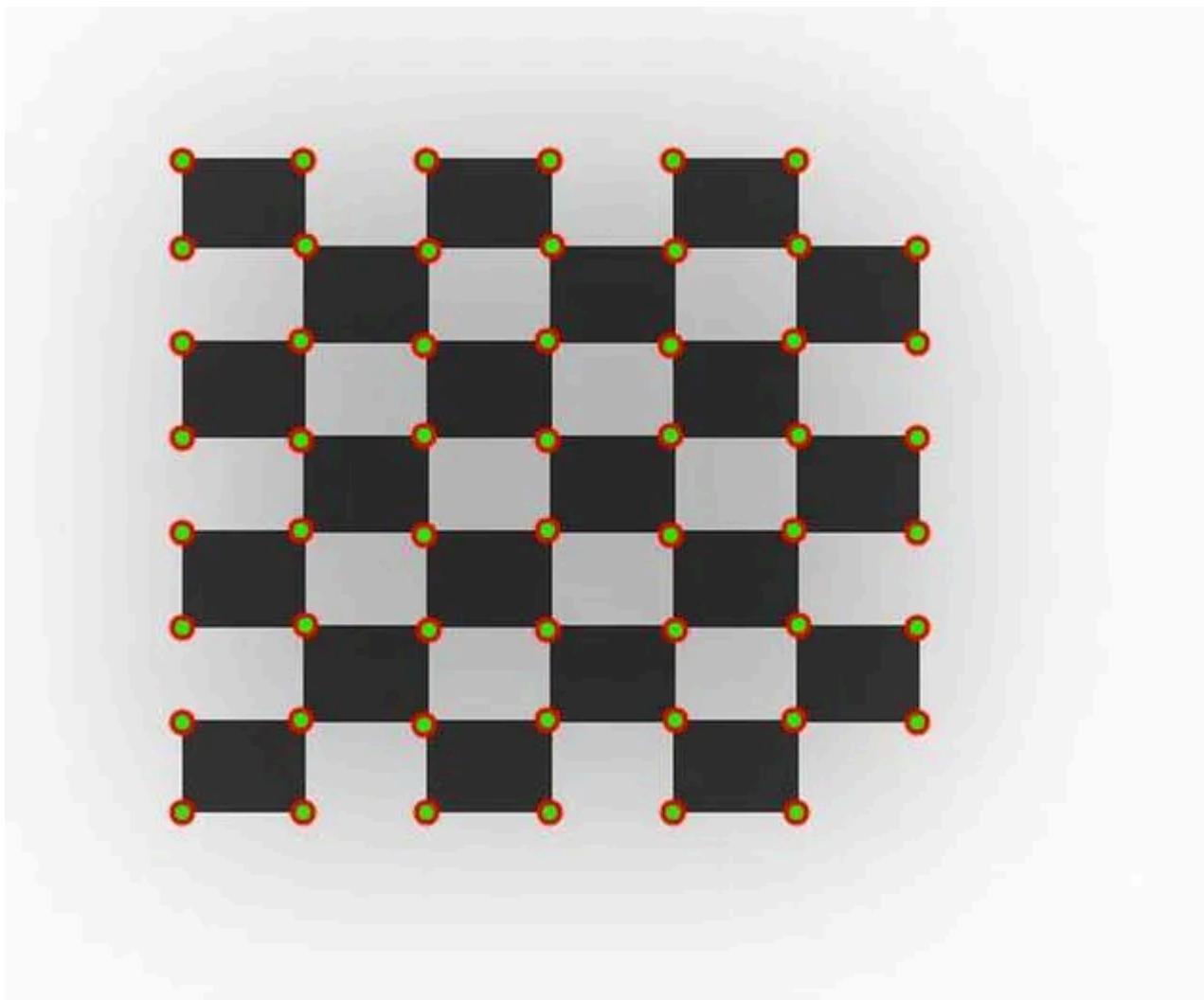
Introduction to Harris Corner Detector

Deep · [Follow](#)

5 min read · Mar 16, 2019

[Listen](#)[Share](#)[More](#)

Harris Corner Detector is a corner detection operator that is commonly used in computer vision algorithms to extract corners and infer features of an image. It was first introduced by Chris Harris and Mike Stephens in 1988 upon the improvement of Moravec's corner detector. Compared to the previous one, Harris' corner detector takes the differential of the corner score into account with reference to direction directly, instead of using shifting patches for every 45-degree angles, and has been proved to be more accurate in distinguishing between edges and corners. Since then, it has been improved and adopted in many algorithms to preprocess images for subsequent applications.

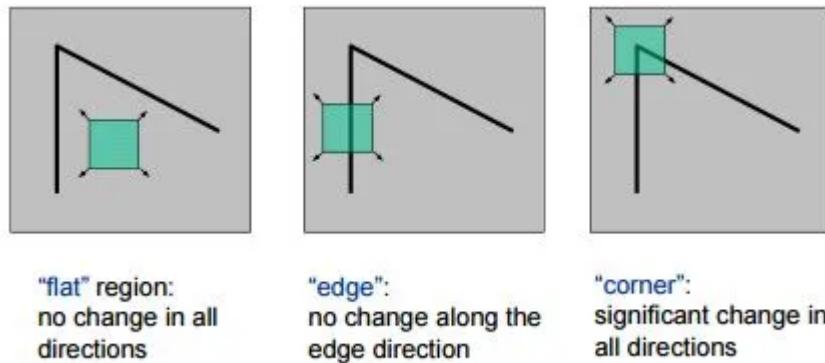


This is part of a 7-series Feature Detection and Matching. Other articles included

- [Introduction To Feature Detection And Matching](#)
- [Introduction to SIFT \(Scale-Invariant Feature Transform\)](#)
- [Introduction to SURF \(Speeded-Up Robust Features\)](#)
- [Introduction to FAST \(Features from Accelerated Segment Test\)](#)
- [Introduction to BRIEF \(Binary Robust Independent Elementary Features\)](#)
- [Introduction to ORB \(Oriented FAST and Rotated BRIEF\)](#)

Corner

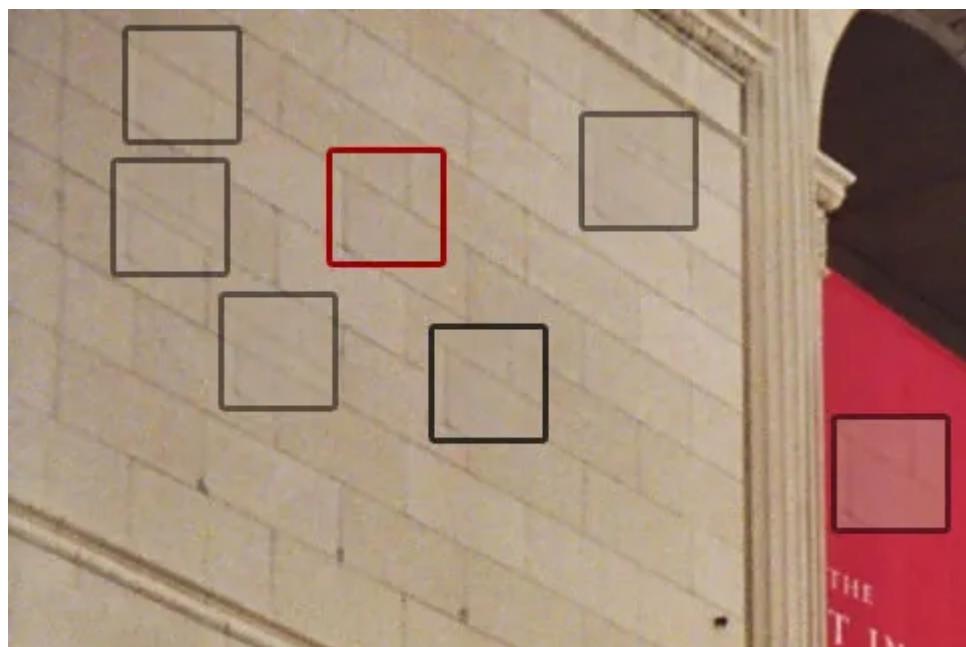
A corner is a point whose local neighborhood stands in two dominant and different edge directions. In other words, a corner can be interpreted as the junction of two edges, where an edge is a sudden change in image brightness. Corners are the important features in the image, and they are generally termed as interest points which are invariant to translation, rotation, and illumination.



So let's understand why corners are considered better features or good for patch mapping. In the above figure, if we take the flat region then no gradient change is observed in any direction. Similarly, in the edge region, no gradient change is observed along the edge direction. So both flat region and edge region are bad for patch matching since they are not very distinctive (there are many similar patches in the flat and edge regions). While in corner region we observe a significant gradient change in all directions. Due to this corners are considered good for patch matching (shifting the window in any direction yields a large change in appearance) and generally more stable over the change of viewpoint.

Corner Detection

The idea is to consider a small window around each pixel p in an image. We want to identify all such pixel windows that are unique. Uniqueness can be measured by shifting each window by a small amount in a given direction and measuring the amount of change that occurs in the pixel values.



More formally, we take the sum squared difference (SSD) of the pixel values before and after the shift and identifying pixel windows where the SSD is large for shifts in all 8 directions. Let us define the change function $E(u,v)$ as the **sum** of all the sum squared differences (SSD), where u,v are the x,y coordinates of every pixel in our 3×3 window and I is the intensity value of the pixel. The features in the image are all pixels that have large values of $E(u,v)$, as defined by some threshold.

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2$$

The equation

We have to maximize this function $E(u,v)$ for corner detection. That means, we have to maximize the second term. Applying Taylor Expansion to the above equation and using some mathematical steps, we get the final equation as:

$$E(u, v) \approx [u \quad v] \left(\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) [u \quad v]$$

Now, we rename the summed-matrix, and put it to be M :

$$M = \sum w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

So the equation now becomes:

$$E(u, v) \approx [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

Remember that we want the SSD to be large in shifts for all eight directions, or conversely, for the SSD to be small for none of the directions. By solving for the eigenvectors of M , we can obtain the directions for both the largest and smallest increases in SSD. The corresponding eigenvalues give us the actual value amount of these increases. A score, R , is calculated for each window:

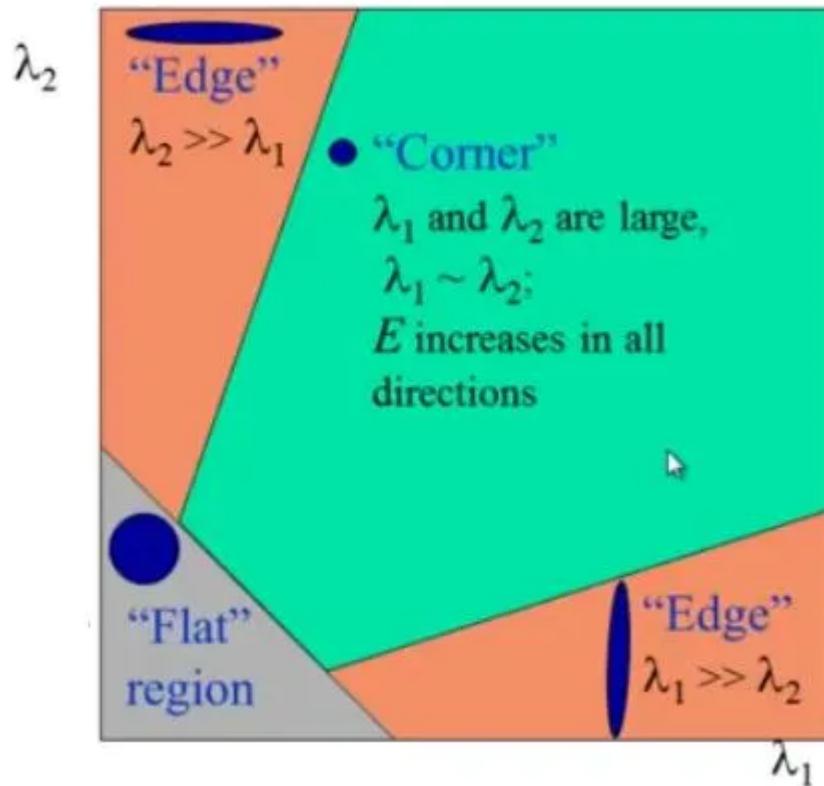
$$R = \det M - k(\text{trace } M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

λ_1 and λ_2 are the eigenvalues of M . So the values of these eigenvalues decide whether a region is a corner, edge or flat.

- When $|R|$ is small, which happens when λ_1 and λ_2 are small, the region is flat.
- When $R < 0$, which happens when $\lambda_1 \gg \lambda_2$ or vice versa, the region is an edge.
- When R is large, which happens when λ_1 and λ_2 are large and $\lambda_1 \sim \lambda_2$, the region is a corner.



High-level pseudocode

1. Take the grayscale of the original image
2. Apply a Gaussian filter to smooth out any noise
3. Apply Sobel operator to find the x and y gradient values for every pixel in the grayscale image

4. For each pixel p in the grayscale image, consider a 3×3 window around it and compute the corner strength function. Call this its Harris value.
5. Find all pixels that exceed a certain threshold and are the local maxima within a certain window (to prevent redundant dupes of features)
6. For each pixel that meets the criteria in 5, compute a feature descriptor.

Implementation

I was able to implement the Harris Corner Detector using OpenCV. Here's how I did it:

Loading

harris.ipynb hosted with ❤ by GitHub

[view raw](#)

Github link for the code: <https://github.com/deepanshut041/feature-detection/tree/master/harris>

References

- https://docs.opencv.org/3.4/dc/d0d/tutorial_py_features_harris.html

- <https://in.udacity.com/course/computer-vision-nanodegree--nd891>
- <http://aishack.in/tutorials/harris-corner-detector/>

Thanks for reading! If you enjoyed it, hit that clap button below and follow Data Breach for more updates

Machine Learning

Computer Vision

Image Processing



Follow

Written by Deep

455 Followers · 10 Following

Responses (7)



What are your thoughts?

Respond



Przemysław Dolata

Sep 15, 2021

...

This text is basically a merger of the Wikipedia article and OpenCV documentation, a plagiarism. PS: you did not *implement* the Harris detector using OpenCV, you merely used an OpenCV implementation.



10

[Reply](#)



Tushar
Oct 12, 2020

...

The concept was not explained properly, the jumps were significant and it was assumed that we already know it.



7

[Reply](#)

Kirill Lastovirya
Jul 13, 2020

...

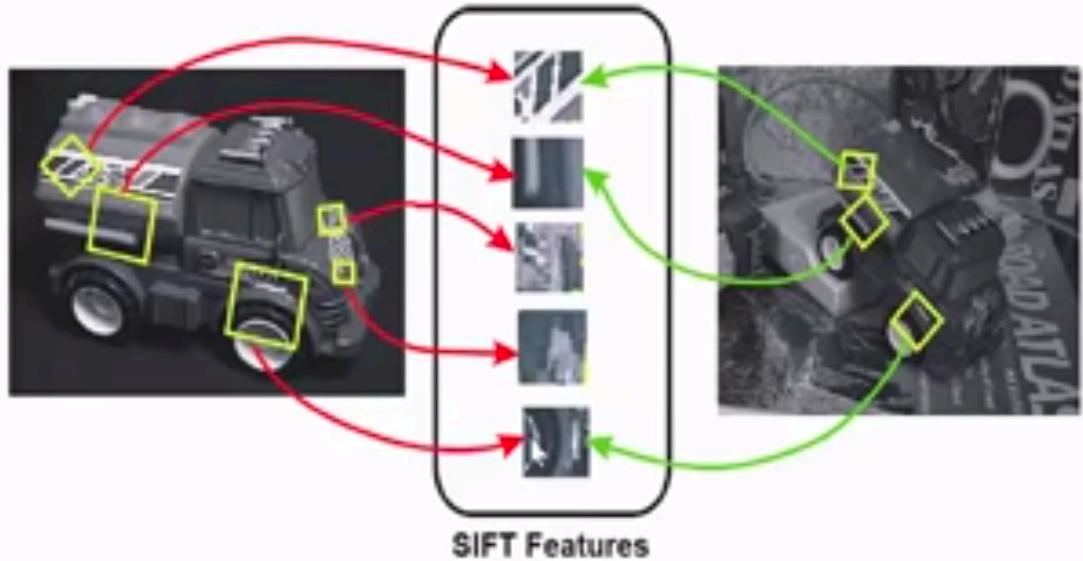
Reading a paper with formulas I expect that symbols used are explained above, so there is no guessing what some symbol means. There are some issues with that in the paper — see my...



3

[See all responses](#)

More from Deep

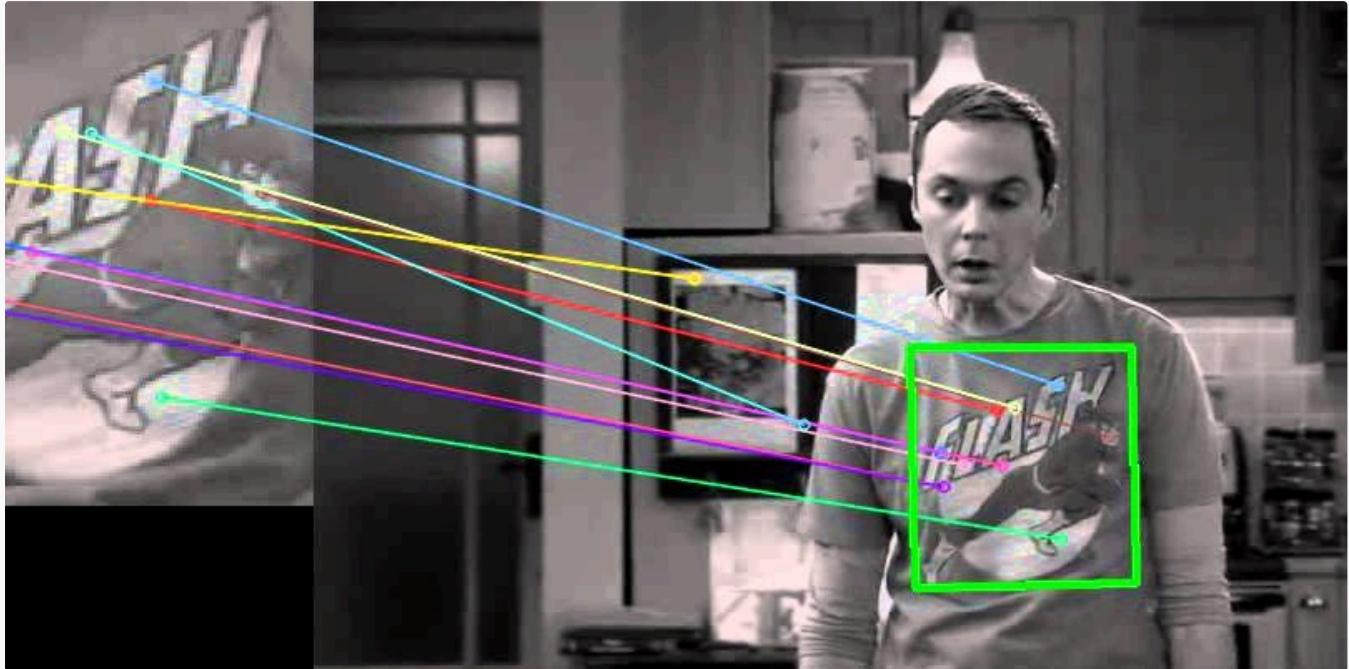


Deep

Introduction to SIFT(Scale Invariant Feature Transform)

SIFT stands for Scale-Invariant Feature Transform and was first presented in 2004, by D.Lowe, University of British Columbia. SIFT is...

Mar 16, 2019 1.6K 8

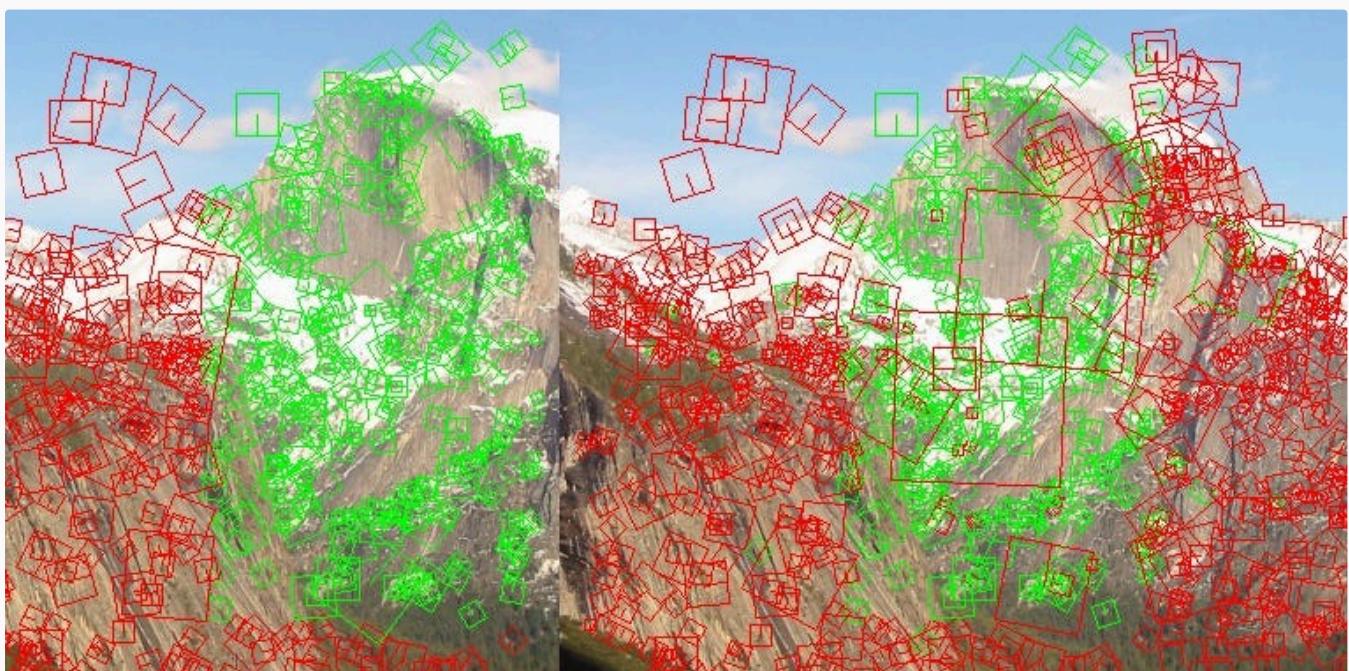


Deep

Introduction to SURF (Speeded-Up Robust Features)

The SURF method (Speeded Up Robust Features) is a fast and robust algorithm for local, similarity invariant representation and comparison...

Mar 20, 2019 597 2



 Deep

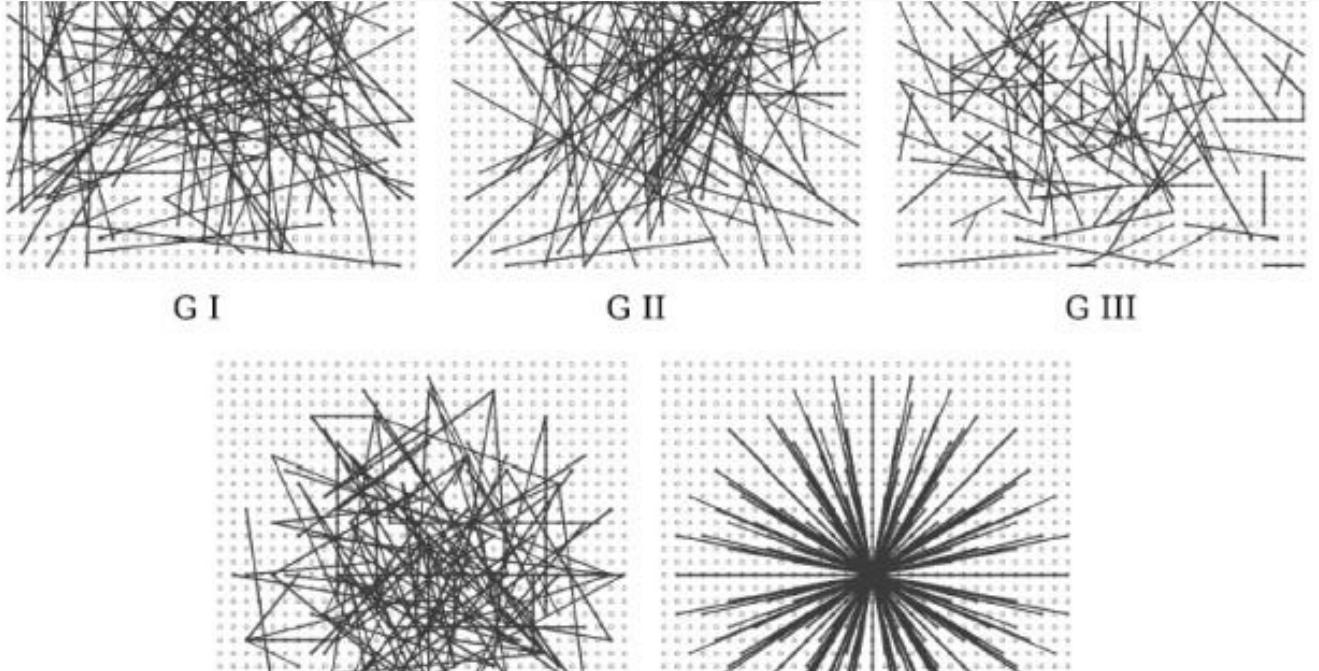
Introduction To Feature Detection And Matching

Feature detection and matching is an important task in many computer vision applications, such as structure-from-motion, image retrieval...

Jan 3, 2019  1.1K  4



...


 Deep

Introduction to BRIEF(Binary Robust Independent Elementary Features)

In this article, we use binary strings as an efficient feature point descriptor, which is called BRIEF. BRIEF is very fast both to build ...

Mar 19, 2019  443  7



...

See all from Deep

Recommended from Medium

 Abisha

Image Feature Extraction using Python - Part I

Basics of Image feature extraction techniques using python

 Sep 5, 2024 623 13

...

 AI Papers Academy

Sapiens by Meta AI: Foundation for Human Vision Models

In this post we dive into Sapiens, a new family of computer vision models by Meta AI that show remarkable advancement in human-centric...

Aug 26, 2024

130



...

Lists



Predictive Modeling w/ Python

20 stories · 1793 saves



Practical Guides to Machine Learning

10 stories · 2173 saves



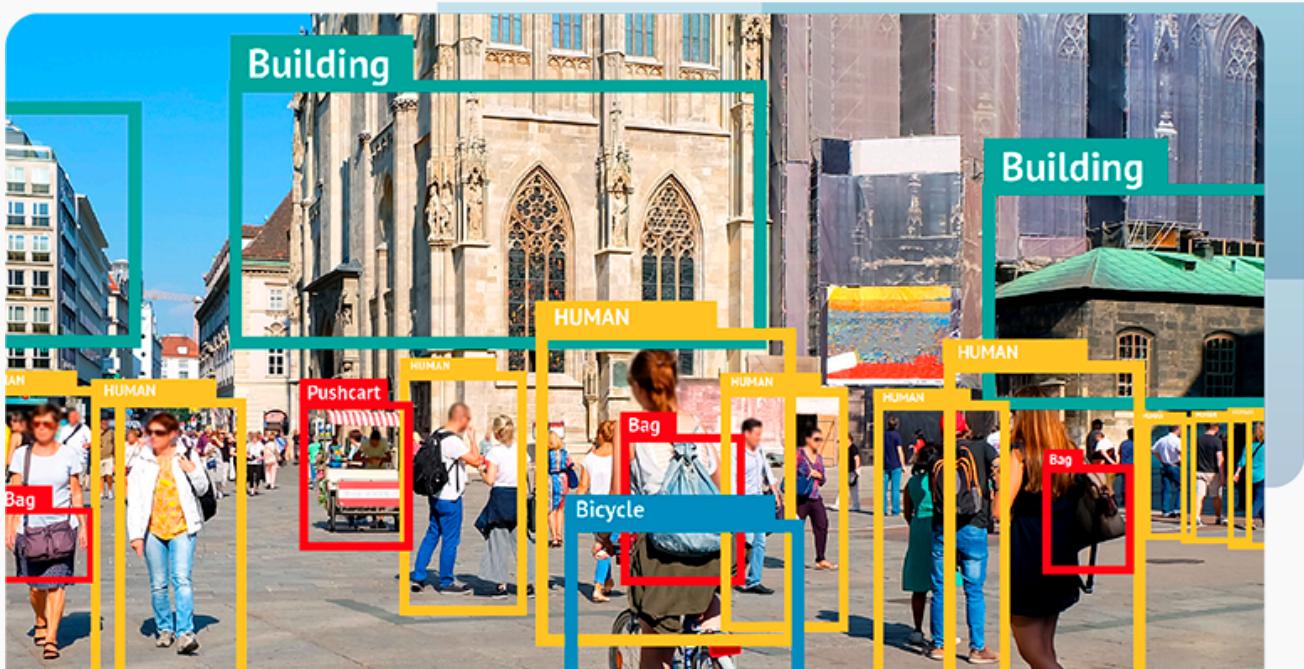
Natural Language Processing

1894 stories · 1556 saves



The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 545 saves



In Tech Spectrum by Aarafat Islam

Top 10 Object Detection Models in 2024

Object detection is a fundamental task in computer vision that involves identifying and localizing objects within an image. Deep learning...

Sep 30, 2024

374

7



...

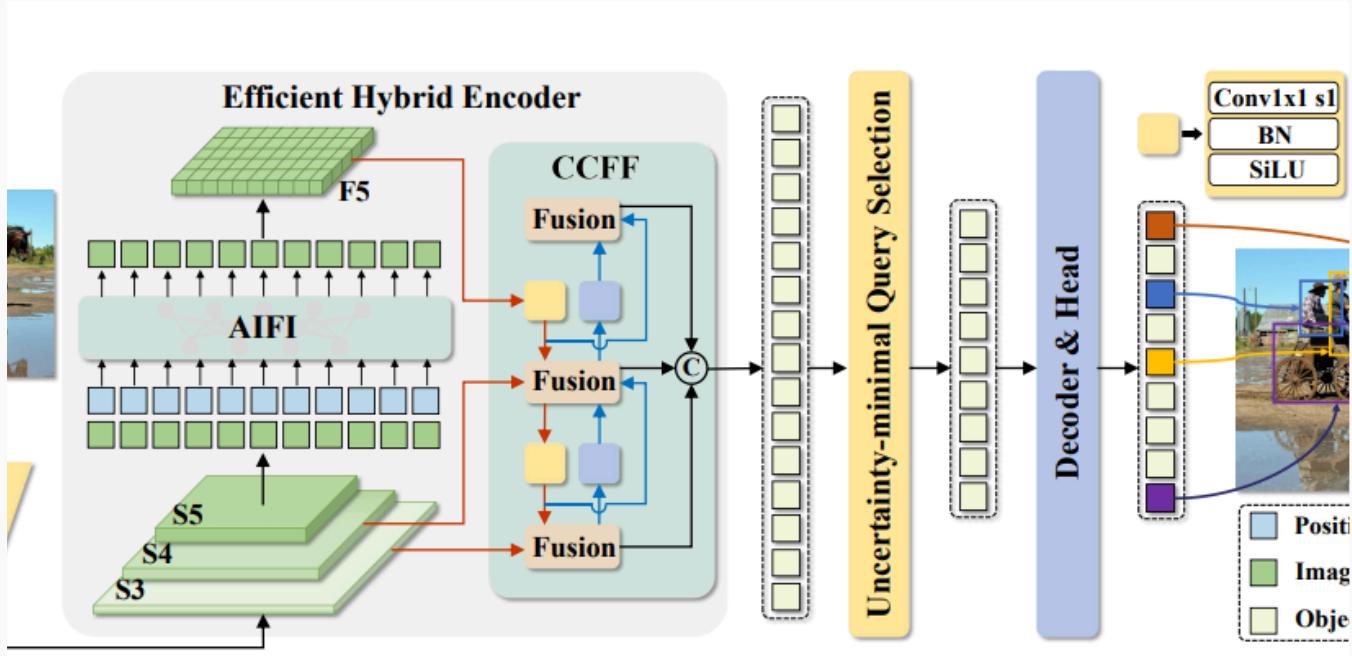
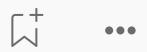


 Jessica Stillman

Jeff Bezos Says the 1-Hour Rule Makes Him Smarter. New Neuroscience Says He's Right

Jeff Bezos's morning routine has long included the one-hour rule. New neuroscience says yours probably should too.

♦ Oct 30, 2024 🙌 20K 🔍 546

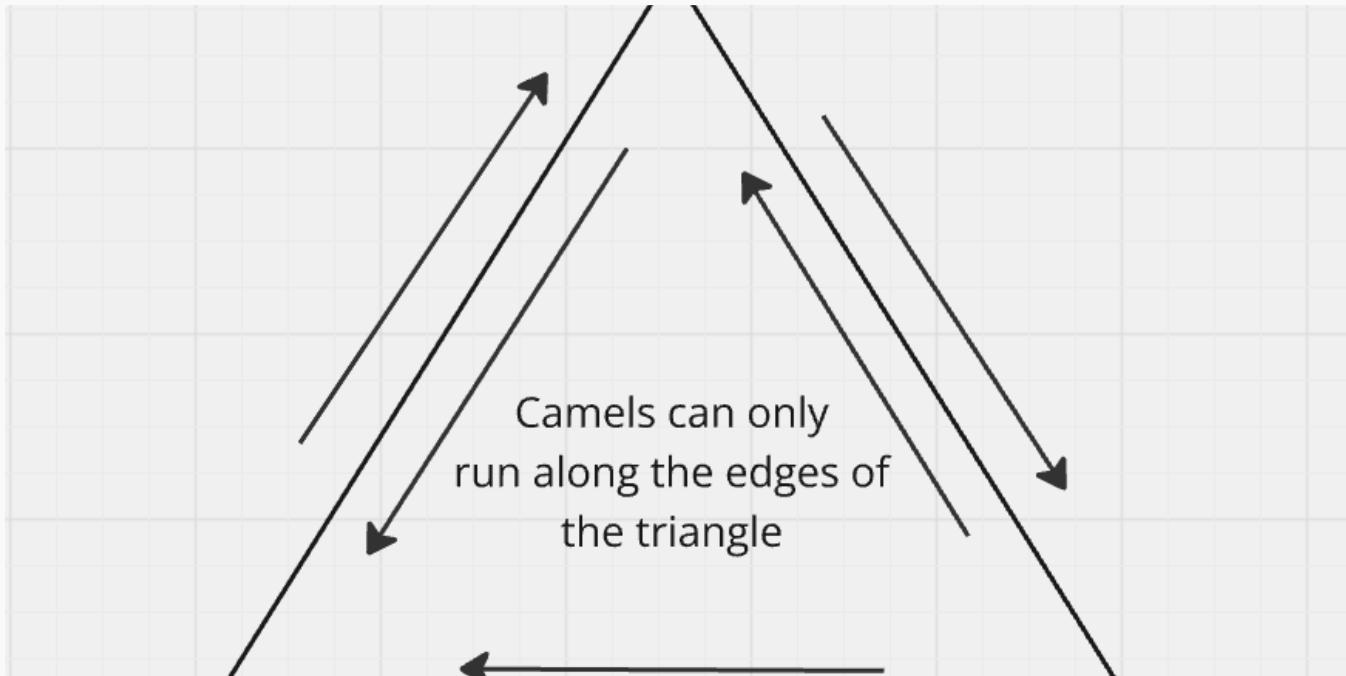


 Antonio Consiglio

RT-DETR: A Faster Alternative to YOLO for Real-Time Object Detection (with Code)

Object detection has always faced a major challenge—balancing speed and accuracy. Traditional models like YOLO have been fast but...

Oct 28, 2024 242 1



 Lucas Samba

3 Probability Questions I was asked in Walmart Data Scientist Interview

Recently I got an opportunity to interview at Walmart for Data Scientist—3 position. All thanks to a referral by my friend working at...

Aug 23, 2024 1.6K 42



See more recommendations