



Introduction to Harris Corner Detector



Deepanshu Tyagi · [Follow](#)

Published in Data Breach

5 min read · Mar 16, 2019

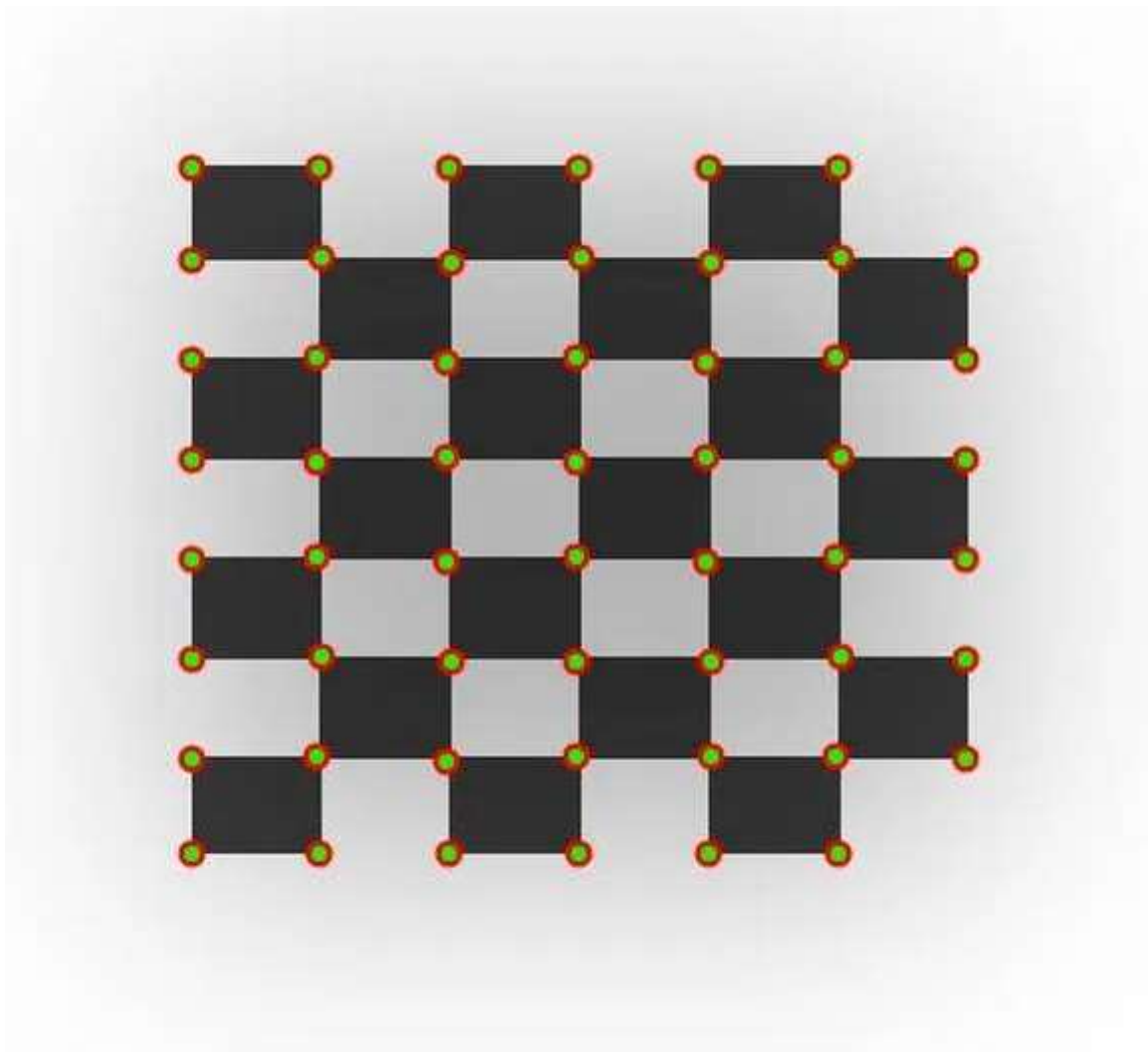


Listen



Share

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.



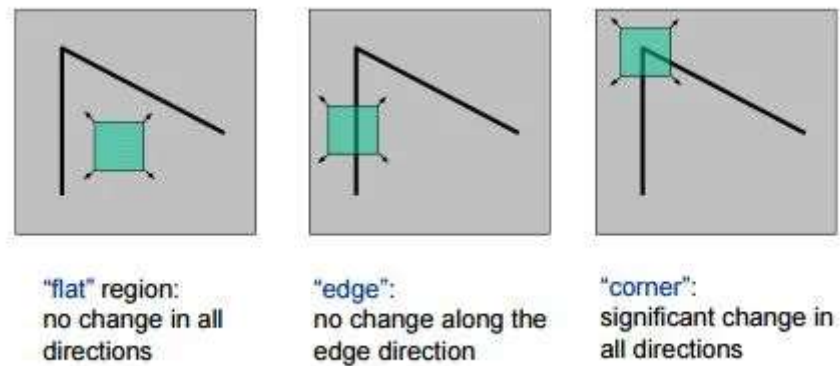
This is part of a 7-series Feature Detection and Matching. Other articles included

- [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

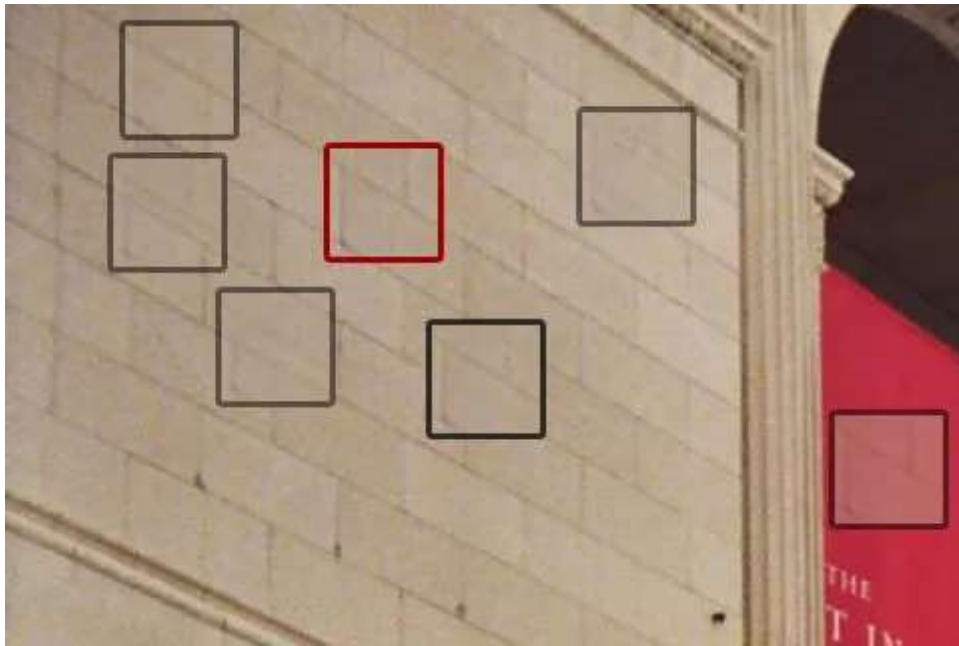
important features in the image, and they are generally termed as interest points which are invariant to translation, rotation, and illumination.



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 along edge in edge region). While in corner region we observe a significant gradient change in all direction. 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:

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

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:

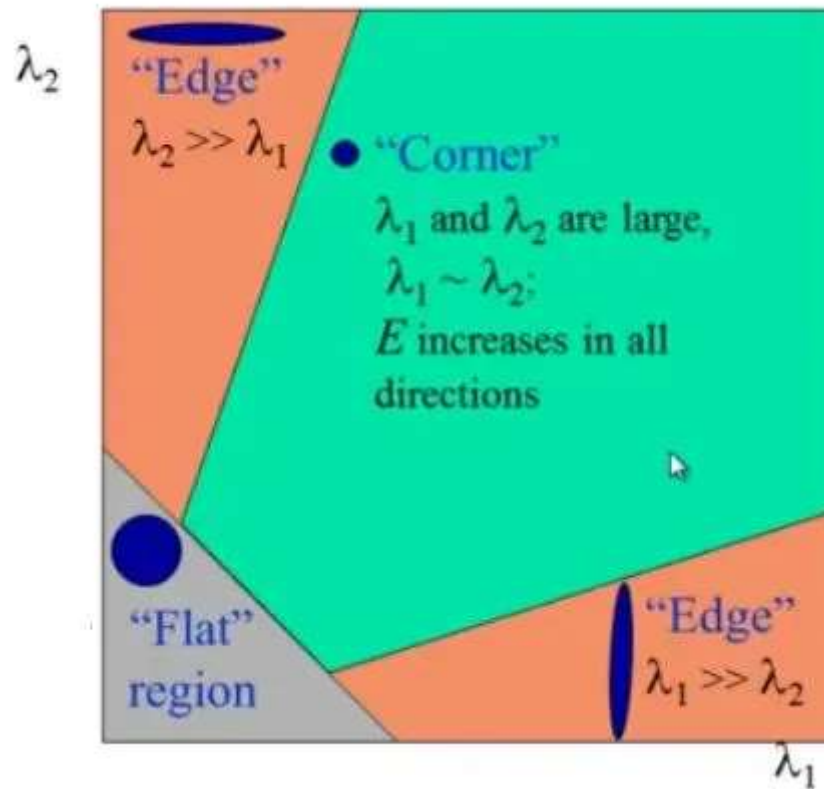
$$R = \det M - k(\text{trace } M)^2$$

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

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

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:

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

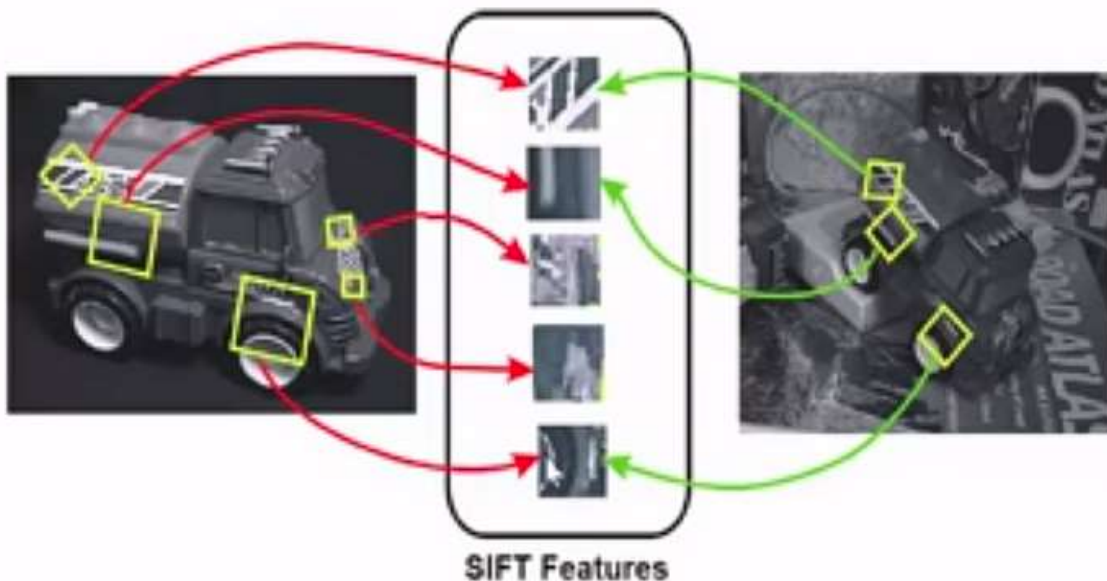


Follow

Written by Deepanshu Tyagi

350 Followers · Editor for Data Breach

More from Deepanshu Tyagi and Data Breach





Deepanshu Tyagi in Data Breach

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...

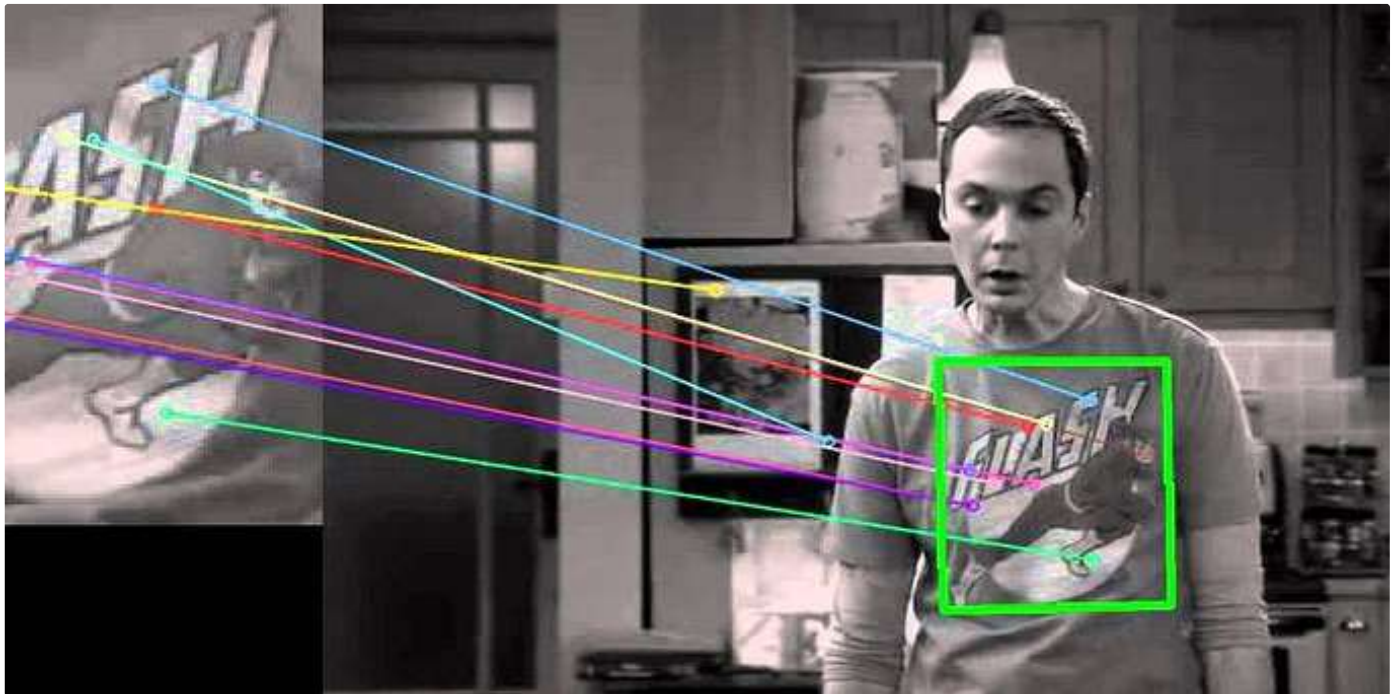
7 min read · Mar 16, 2019



1.3K



7



Deepanshu Tyagi in Data Breach

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...

7 min read · Mar 20, 2019

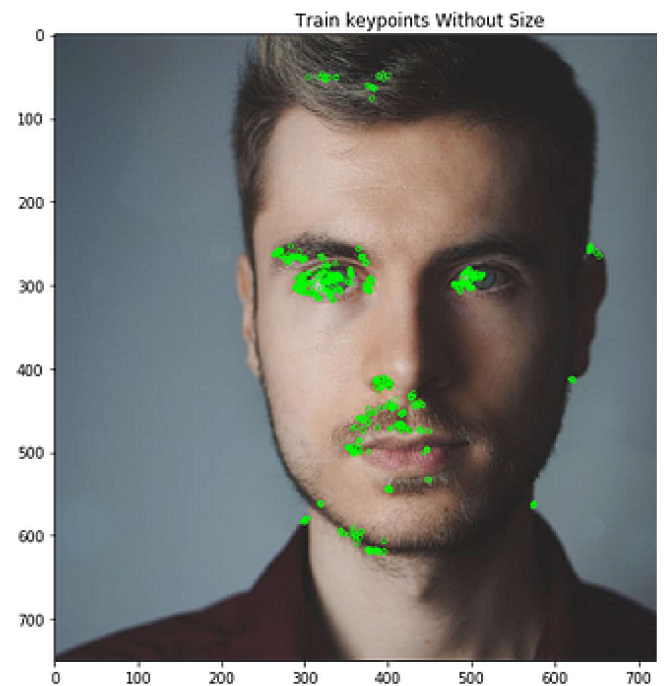
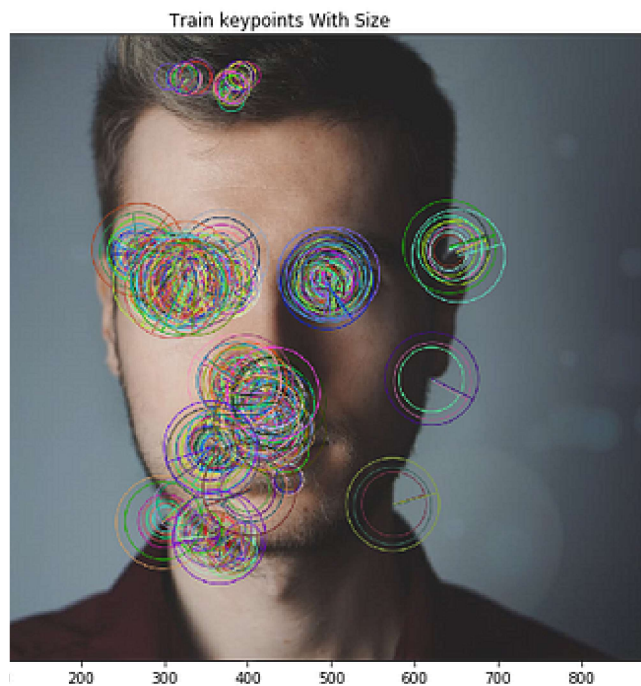



560



2





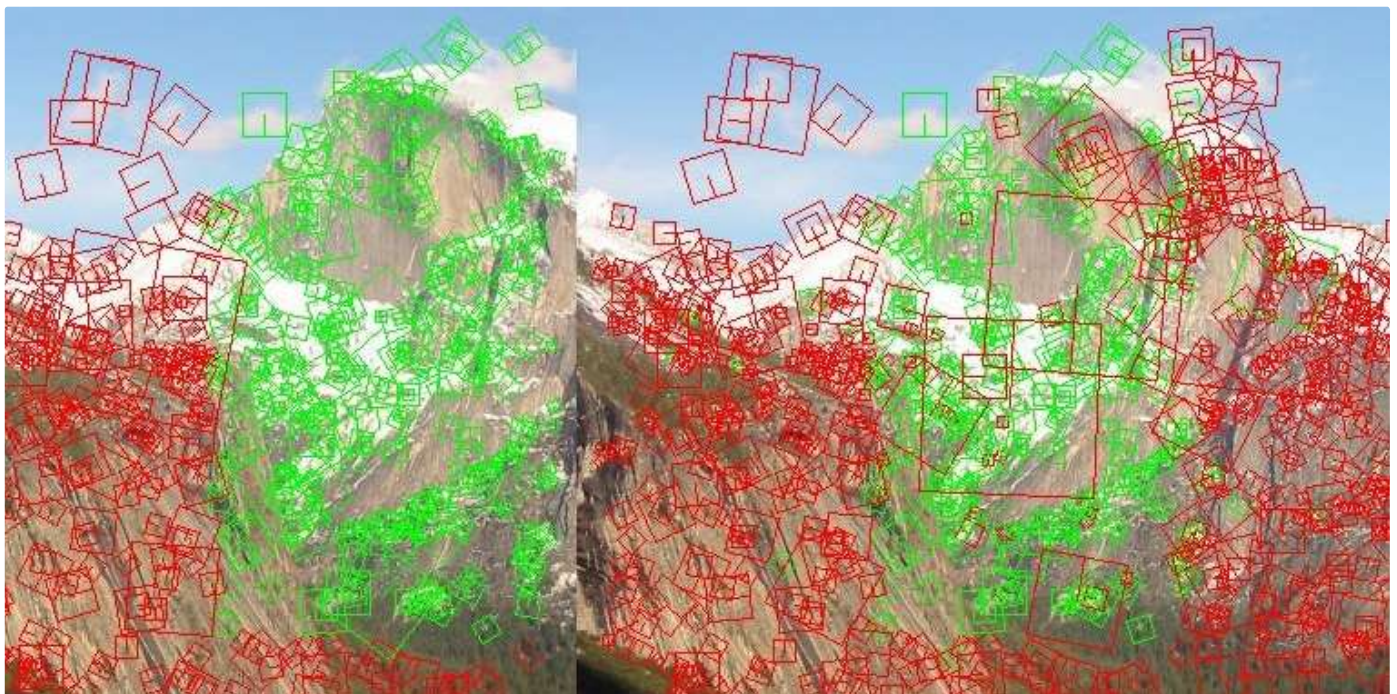
 Deepanshu Tyagi in Data Breach

Introduction to ORB (Oriented FAST and Rotated BRIEF)

Oriented FAST and Rotated BRIEF (ORB) was developed at OpenCV labs by Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski in...

6 min read · Jan 1, 2019

 1.1K  6





Deepanshu Tyagi in Data Breach

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...

5 min read • Jan 3, 2019



1K



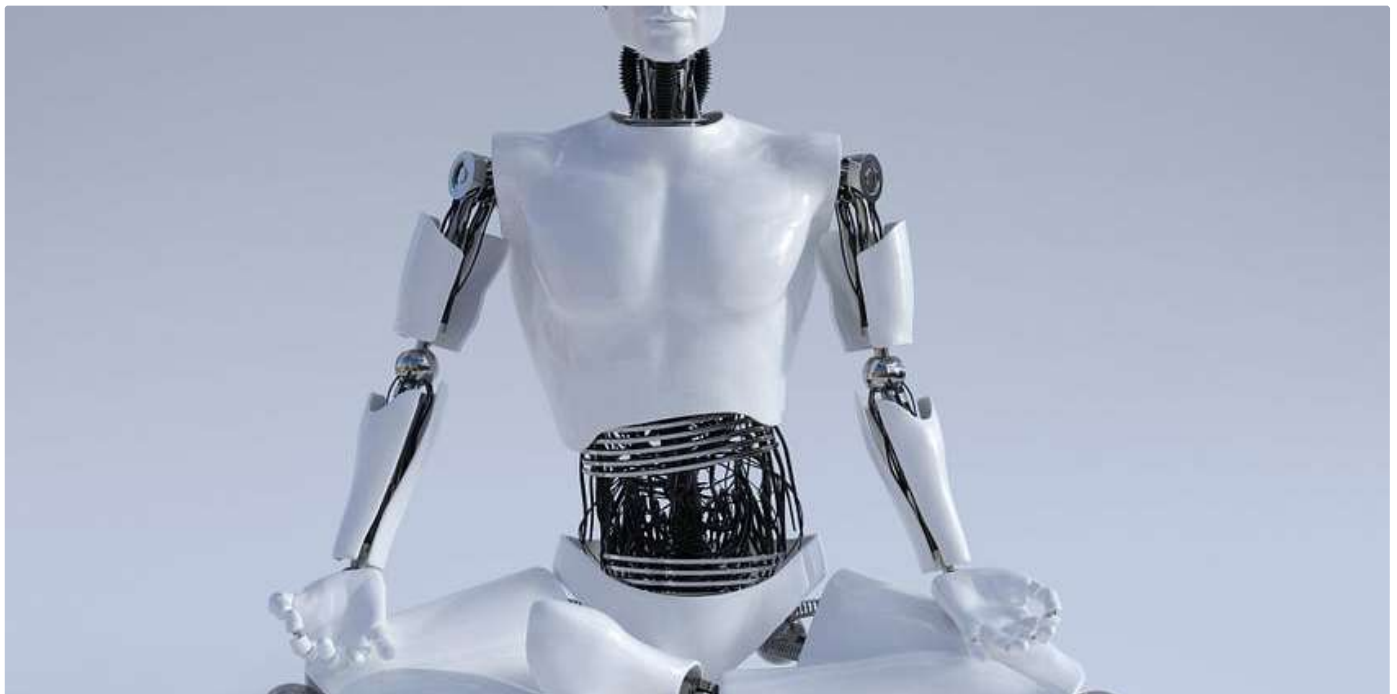
4



See all from Deepanshu Tyagi

See all from Data Breach

Recommended from Medium





The PyCoach in Artificial Corner

You're Using ChatGPT Wrong! Here's How to Be Ahead of 99% of ChatGPT Users

Master ChatGPT by learning prompt engineering.

🌟 • 7 min read • Mar 17



18.1K



328



Somnath Singh in JavaScript in Plain English

Coding Won't Exist In 5 Years. This Is Why

Those who won't adapt would cease to exist.

🌟 • 8 min read • Jan 20



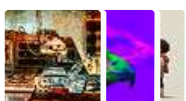
7.8K



262



Lists



What is ChatGPT?

9 stories • 22 saves



Staff Picks

300 stories · 61 saves



Alexander Nguyen in Level Up Coding

Why I Keep Failing Candidates During Google Interviews...

They don't meet the bar.

★ · 4 min read · Apr 13

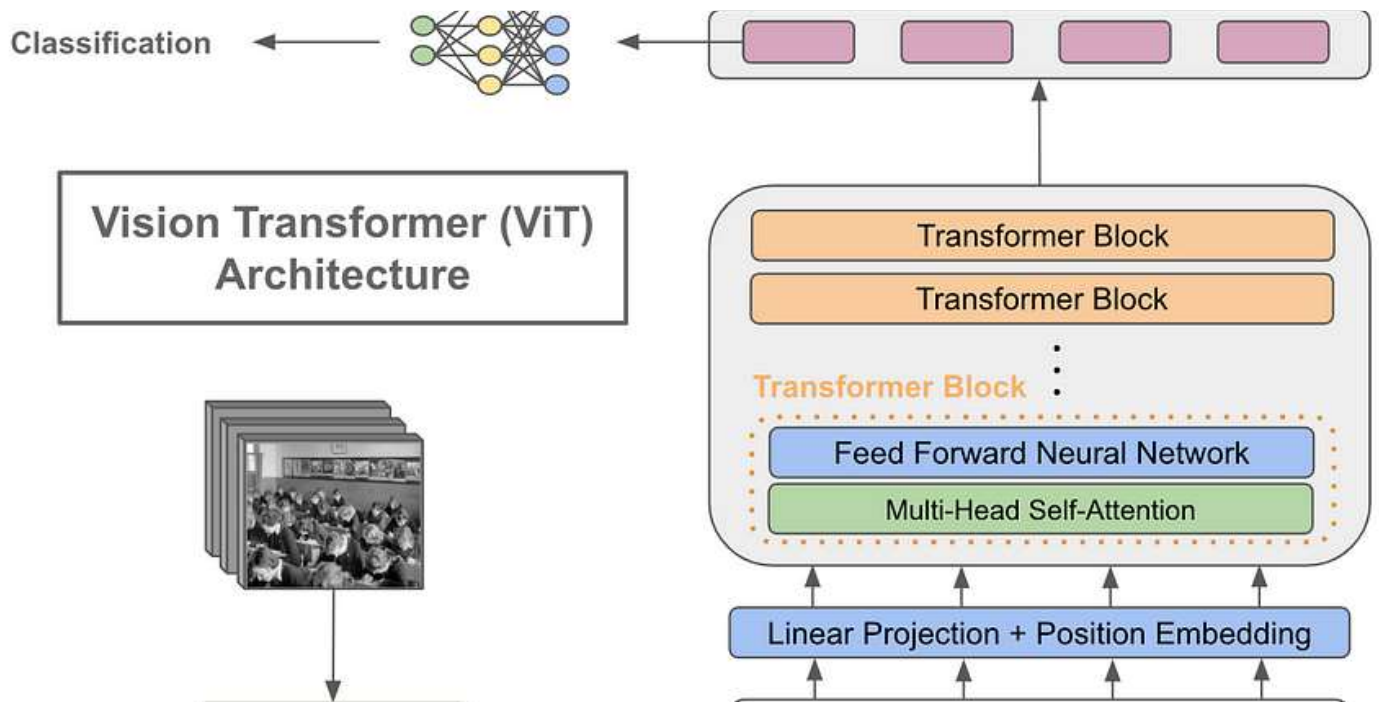



3.5K



109





 Cameron R. Wolfe in Towards Data Science

Using Transformers for Computer Vision

Are Vision Transformers actually useful?

★ • 13 min read • Oct 5, 2022

 148  4





Bryan Ye in Better Humans

How To Wake Up at 5 A.M. Every Day

An unconventional and compassionate guide to becoming an early bird

★ • 15 min read • Oct 3, 2019



85K



639



Unbecoming

10 Seconds That Ended My 20 Year Marriage

It's August in Northern Virginia, hot and humid. I still haven't showered from my morning trail run. I'm wearing my stay-at-home mom...

★ • 4 min read • Feb 16, 2022



46K



756



See more recommendations