

Resizing Images using Various Interpolation Techniques



Annmay Sharma · Follow

5 min read · Jun 11, 2021



Listen



Share

Following up on my previous article on getting crisper images by manual resizing, another method used is through interpolation. Interpolation is a method of constructing new data points within the range of a discrete set of known data points. We interpolate (or estimate) the value of that function for an intermediate value of the independent variable.

There are various types of interpolation. Let's focus on three of them —

1. Nearest Neighbour Interpolation

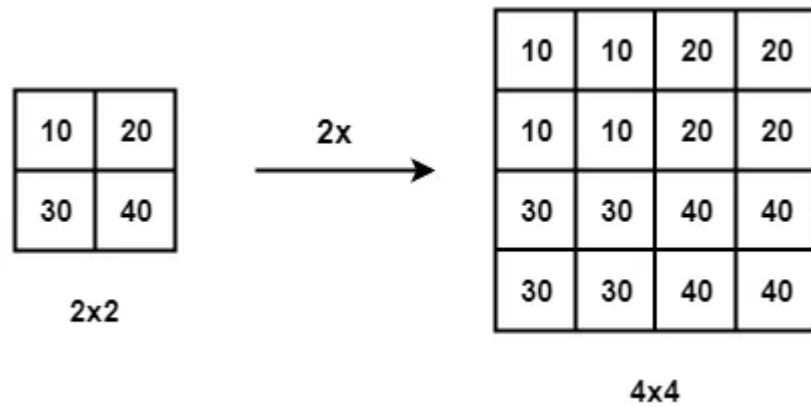
This type of interpolation is the most basic. We simply interpolate the nearest pixel to the current pixel. Assuming, we index the values of the pixels from 0. The pixels of the below 2x2 image will be as follows: {'10':(0,0), '20': (1,0), '30':(0,1), '40': (1,1)}

10	20
30	40

We then project this image on the 4x4 image we require to find the pixels. We find the unknown pixels to be at (-0.5, -0.5), (-0.5, 0.5) and so on...

Now compare the values of the known pixels to the values of the nearest unknown pixels. Thereafter, assign the nearest value i.e $P(-0.5, -0.5)$ as 10 which is the value of the pixel at (0,0).

The result is as follows -



As we can see, we get a 4x4 image where every pixel looks bigger as compared to the original image. Since this method simply looks for the nearest neighbour and computes one pixel at a time, it takes the least processing time.

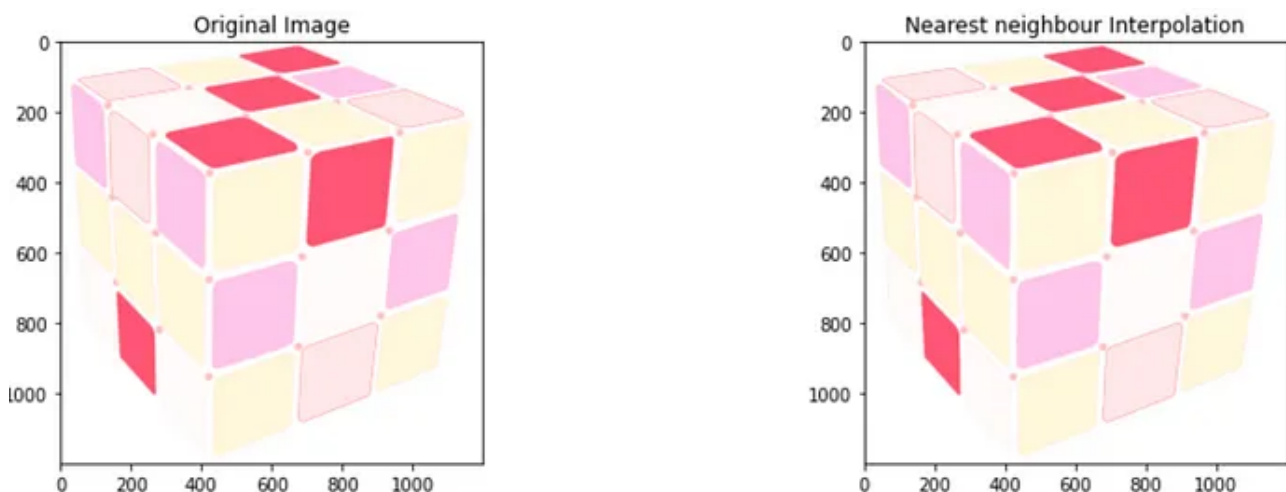
To use this type of interpolation to resize an image in openCV we use the `resize` function with the `cv2.INTER_NEAREST` interpolation flag.

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

img = cv2.imread("cube.png", -1);
h, w, c = img.shape;

img_n = cv2.resize(img, (w//2, h//2), interpolation =
cv2.INTER_NEAREST);
img_n = cv2.resize(img_n, (w, h), interpolation =
cv2.INTER_NEAREST);
```

This snippet of code outputs the following result -



This form of interpolation simply makes each pixel bigger and is often useful when we want to resize images which do not have as complex details like a bar code.

2. Bilinear interpolation

In bilinear interpolation we take the values of four nearest known neighbours (2x2 neighbourhood) of unknown pixels and then take the average of these values to assign the unknown pixel.

Let's first understand how this would work on a simple example. Suppose we take a random point say (0.75, 0.25) which is in the middle of four points -

(0,0),(0,1),(1,0),(1,1).

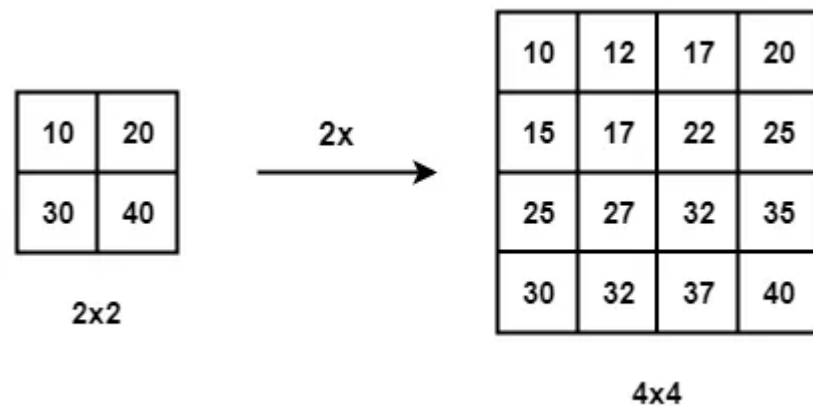
We first find the values at points A(0.75, 0) and B(0.75, 1) using linear interpolation. Linear interpolation is basically taking an approximation of a point between two points by scaling the point according to how close it is to the respective two points.

We then find the value of the pixel required (0.75,0.25) using linear interpolation on points A and B.

Now that we understand how the values are found, let us take it in the context of the 2x2 image that had taken for the nearest neighbour interpolation.

Consider the 2x2 image to be projected onto a 4x4 image but only the corner pixels retain the values. The remaining pixels which are technically in the middle of the four are then calculated by using a scale to assign weights depending on the closer pixel.

For example, consider pixel (0,0) to be 10 and pixel (0,3) to be 20. Pixels (0,1) will be calculated by taking $(0.75 * 10) + (0.25 * 20)$ which gives us 12.5. Similarly on running linear interpolation on the image while resizing, the effect will be as follows:

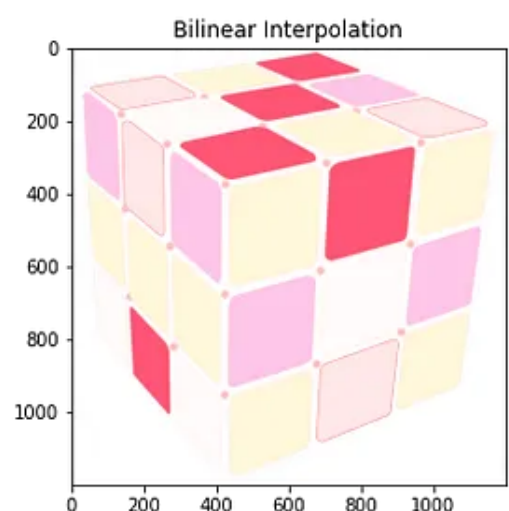
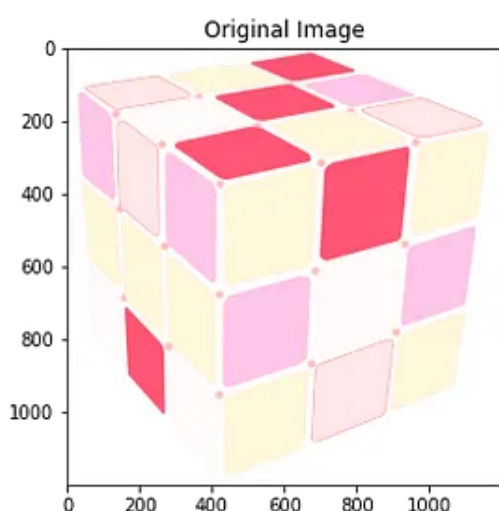


Bilinear interpolation has a longer processing time than nearest neighbour interpolation as it takes in the value of four pixels to compute the interpolated pixel. However, it gives a smoother output.

To use this type of interpolation to resize an image in openCV we use the `resize` function with the `cv2.INTER_LINEAR` interpolation flag. Import the same libraries as given under the nearest neighbour interpolation method above, read the image using `cv2` and then resize with the `cv2.INTER_LINEAR` interpolation flag.

```
img_b = cv2.resize(img, (w//2, h//2), interpolation =
cv2.INTER_LINEAR);

img_b = cv2.resize(img_b, (w, h), interpolation = cv2.INTER_LINEAR);
```



3. Bicubic Interpolation

In bicubic interpolation we take 16 pixels around the pixel to be interpolated (4x4 neighbourhood) as compared to the 4 pixels (2x2 neighbourhood) we take into account for bilinear interpolation.

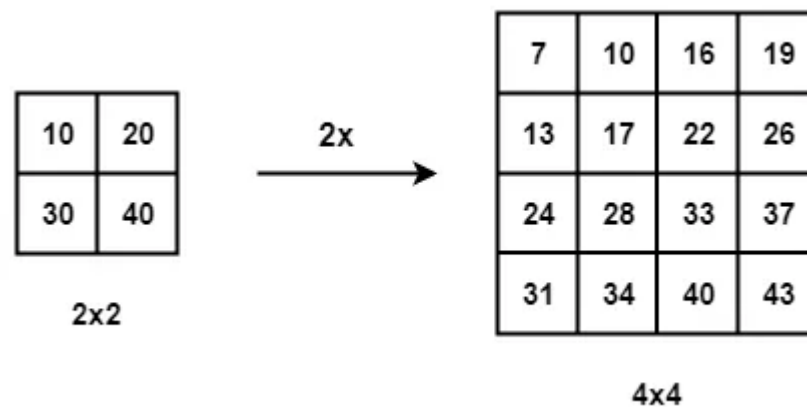
Considering a 4x4 surface, we can find the values of the interpolated pixels using this formula:

$$p(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j.$$

Source(Wikipedia)

The interpolation problem consists of determining the 16 coefficients a_{ij} . These coefficients can be determined from the $p(x, y)$ values which are attained from the matrix of pixels and partial derivatives of individual pixels.

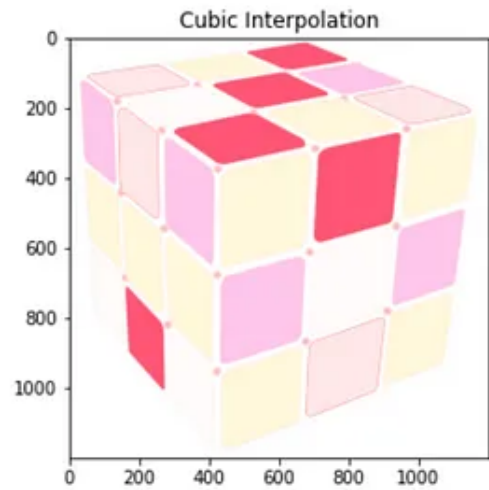
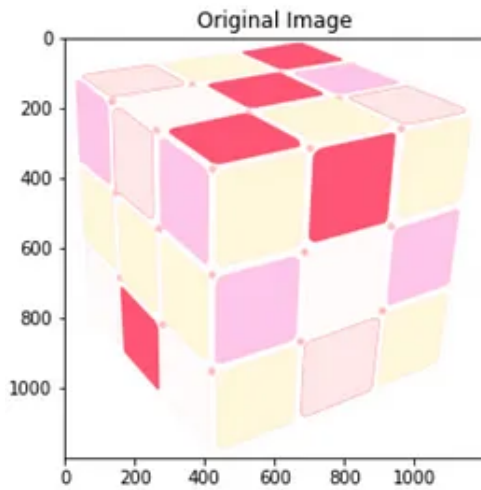
Upon calculating the coefficients, we then multiply them with the weights of the known pixels and interpolate the unknown pixels. Let us take the same input 2x2 image we took in the two examples above. Upon bicubic interpolation, we get the following result:



Now, in order to execute this interpolation using cv2, we will once again call the `resize` function but this time with the `cv2.INTER_CUBIC` interpolation flag. Once again, run the code after importing all the necessary libraries and reading the image using `cv2.imread()`

```
img_c = cv2.resize(img, (w//2, h//2), interpolation =
cv2.INTER_CUBIC);

img_c = cv2.resize(img_b, (w, h), interpolation = cv2.INTER_CUBIC);
```

[Open in app](#)[Sign up](#)[Sign in](#)**Medium**

Thus, we can see that different interpolation techniques have different use cases. Hence, it is important to understand the type of interpolation that will be most useful when resizing images.

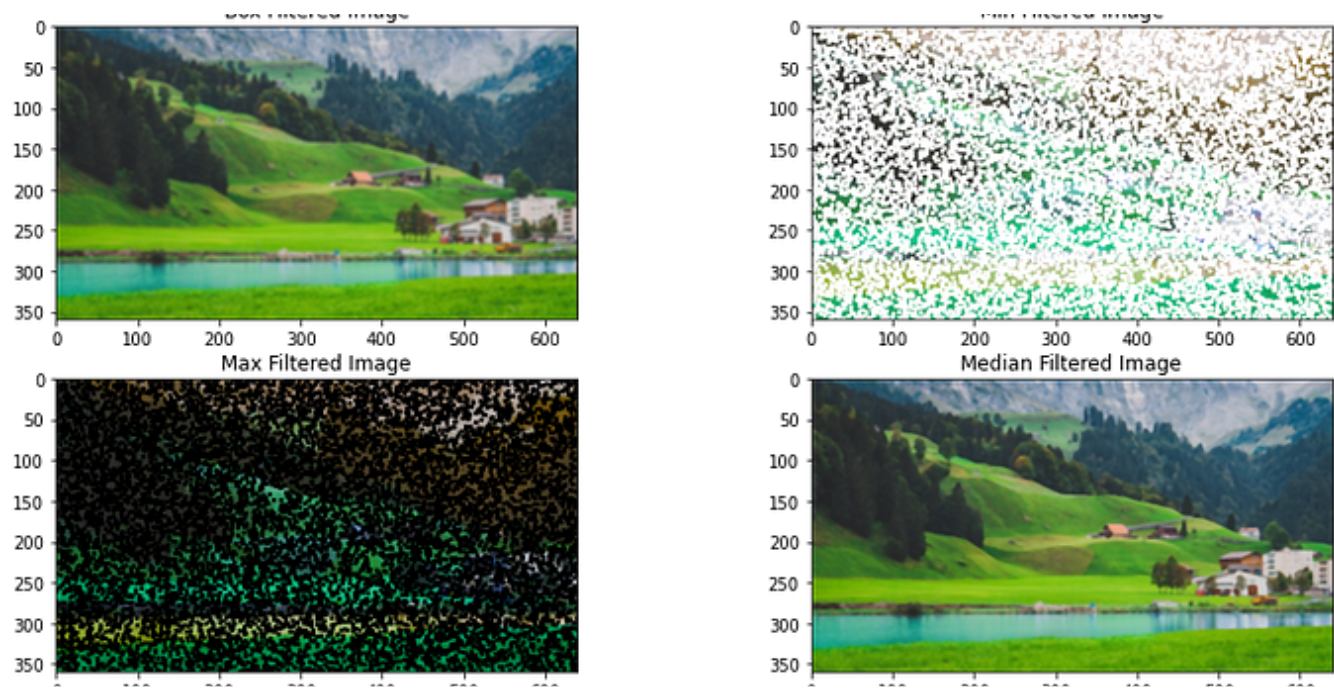
Happy coding to you!


[Computer Vision](#)[Opencv](#)[Image Processing](#)[Machine Learning](#)[Interpolation](#)[Follow](#)**Written by Annmay Sharma**

6 Followers

Data Science and Machine Learning Enthusiast

More from Annmay Sharma



 Annmay Sharma

How Image Filtering is used to Improve Picture Quality

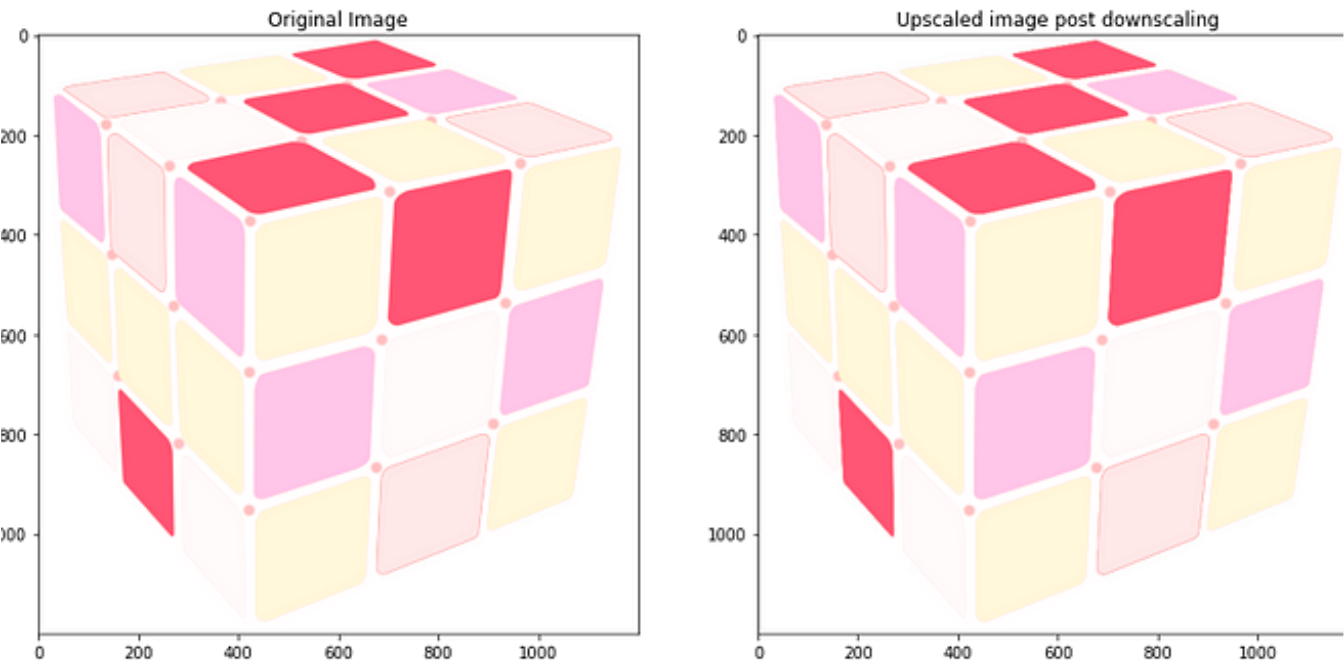
Image Filtering is used everywhere nowadays. Right from social media applications and news articles to various training models to improve...

6 min read · Jun 29, 2021

 5







a

Annmay Sharma

Getting Crisper Images with Manual Resizing

When training vision models, it is common to resize images to a lower dimension ((224 × 224), (299 × 299), etc.) to allow mini-batch...

3 min read · Jun 8, 2021



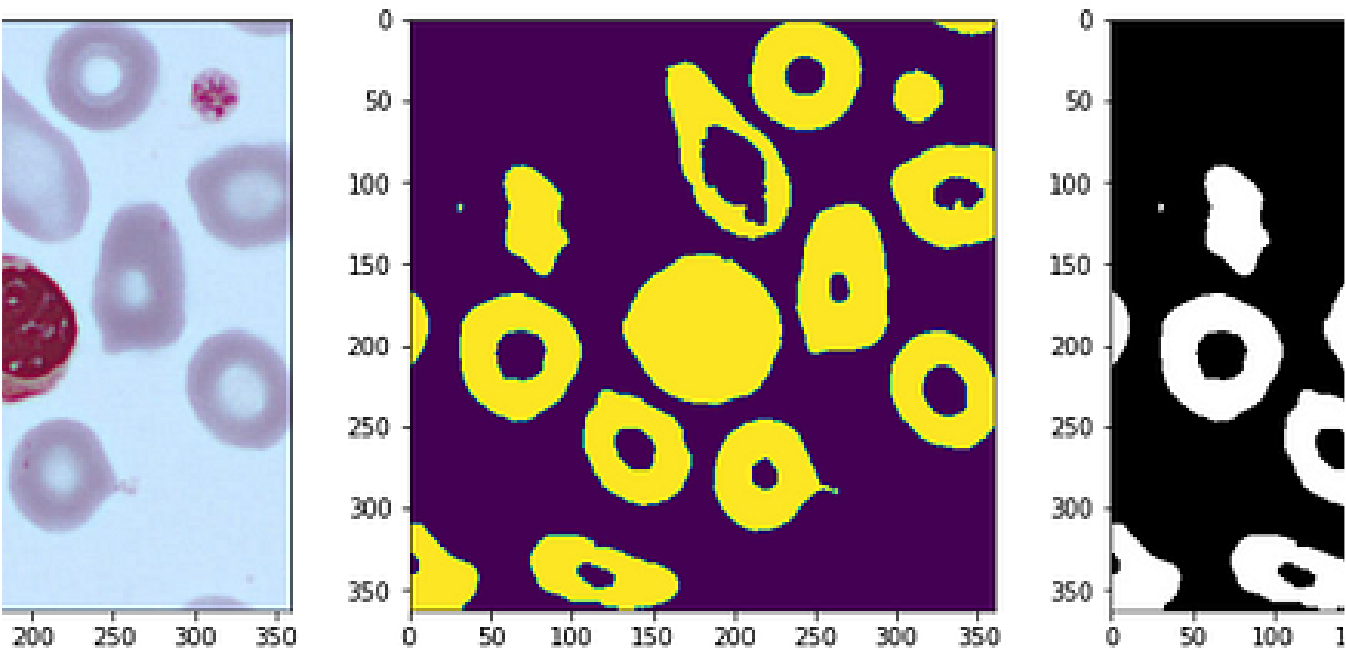
1






See all from Annmay Sharma

Recommended from Medium





Sasani S. Perera

OpenCV: Object Masking

Object masking is very important in Image processing. This article discusses the theories behind object masking and a basic demonstration.

6 min read · Jul 13



10



Girish Venkatachalam

How to do image filters with fabric.js?

Look at some 12 filters on images

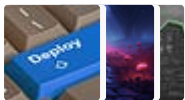
4 min read · Oct 1



1



Lists



Predictive Modeling w/ Python

20 stories · 685 saves



Practical Guides to Machine Learning

10 stories · 782 saves



Natural Language Processing

976 stories · 468 saves



The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 236 saves

IMAGE FILTERING IN COMPUTER VISION

 Sumitkrsharma

Image Filtering in Computer vision

This post is about, brief understanding of image filtering step in computer vision, what is need of applying filters over images during...

6 min read · Jul 28





Flippy Garcia

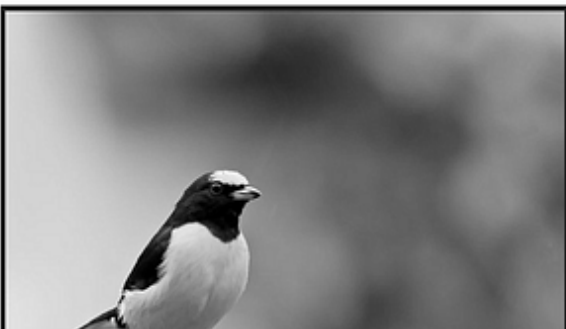
Template Matching using Python

In this part of the journey, we will discuss how to do template matching in an image. Using a template image, you will find where the...

5 min read · Jun 17



5



Nutan

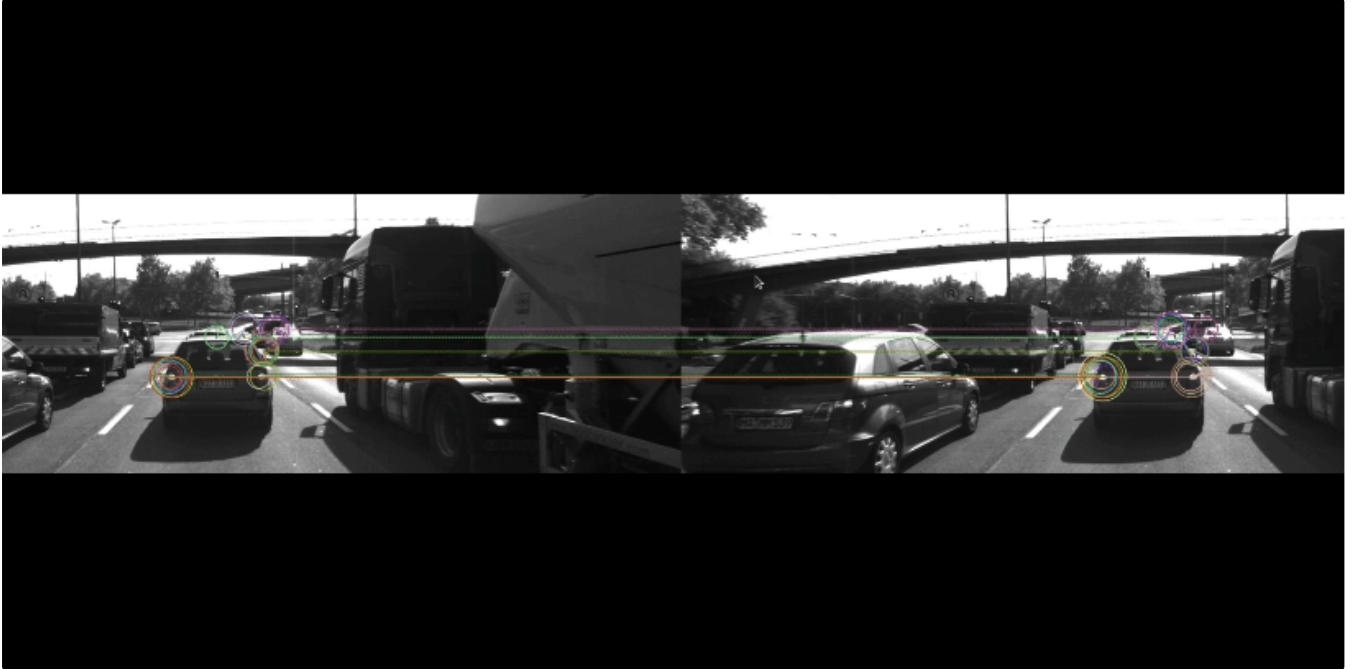
Convert a Color Image to Gray Scale in Python

In this blog, we will convert color images into grayscale using the Pillow and CV2 packages.

5 min read · Jul 27



100



Nikhil Nair

2D Feature Tracking—Part 3: Feature Matching

In the first two parts of this series, we learned about detecting keypoints in an image, which are significant landmarks, and describing...

9 min read · Sep 17



9



See more recommendations