# Computer Vision Report

Name

Institution Affiliation

Course

Date

Contents

[Computer Vision Report 1](#_Toc84722784)

[1 a.Mean filter 2](#_Toc84722785)

[b.sharpening filter 3](#_Toc84722786)

[2. Harris detector 4](#_Toc84722787)

[2.a SIFT 6](#_Toc84722788)

[2.SURF 7](#_Toc84722789)

[Compsrison between SIFT and SURF 9](#_Toc84722790)

[4. implement k- means clustering-based segmentation 9](#_Toc84722791)

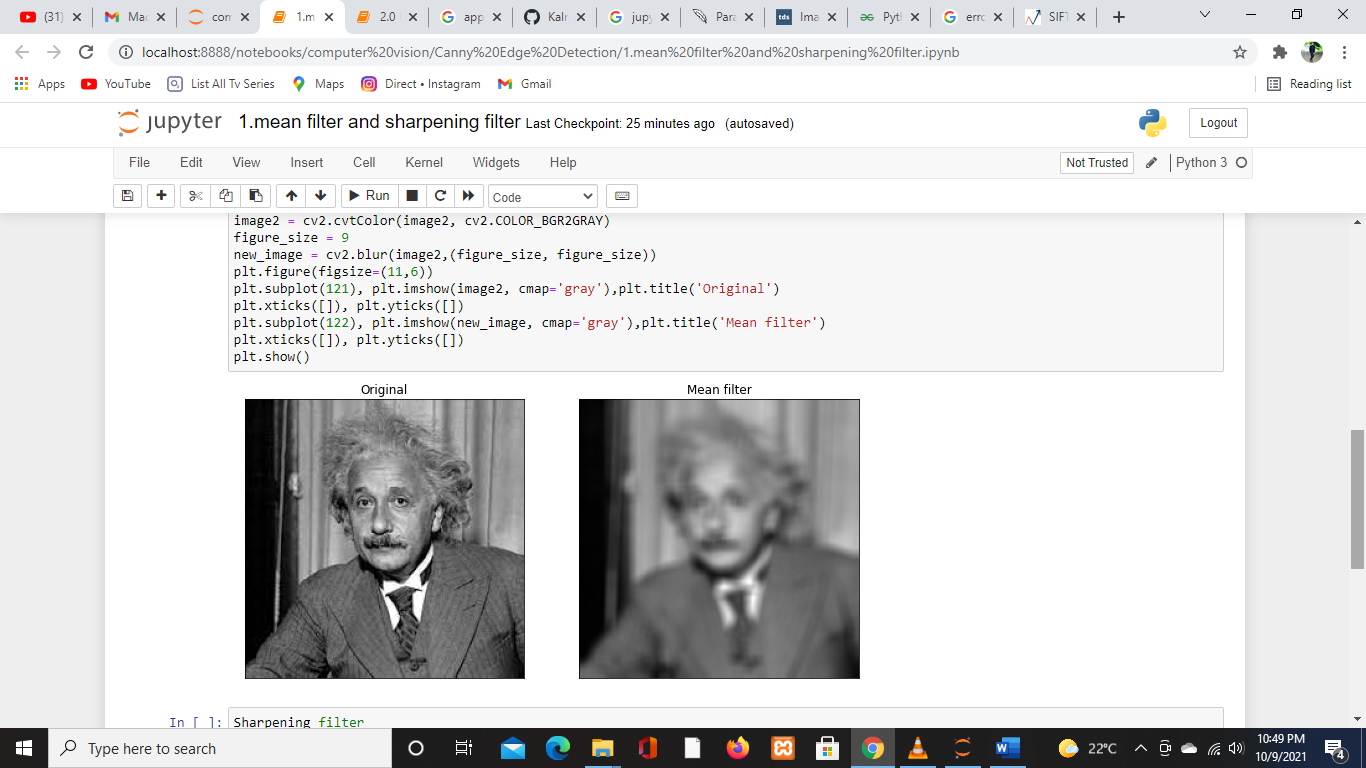
[5 Implement feature detection and matching 10](#_Toc84722792)

[References 11](#_Toc84722793)

# 1 a.Mean filter

To eliminate noise from an image, the mean filter is applied. The mean of the pixel values within an n × n kernel must be determined. After that, the mean replaces the pixel intensity of the central element. This smooths the image's borders and removes some of the image's noise. A mean filter can be applied to a picture with the blur function from the Open-CV library.

When working with color photos, you must first convert from RGB to HSV since RGB's dimensions are interdependent, whereas HSV's three dimensions are independent (this allows us to apply filters to each of the three dimensions separately.)



# 

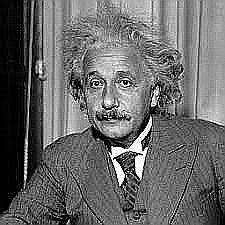
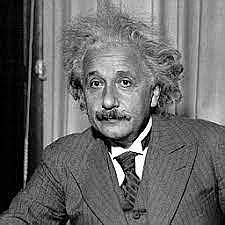
# B.sharpening filter

The Pillow library's ImageFilter.SHARPEN class implements a spatial filter that uses convolution to sharpen a given picture.

The open() function of the Pillow's Image class is used to create an image object by giving the file name of the Image.

The filter() function on the Image object is used to apply a filter to an image. Only the filter's class name is supplied as a parameter.

The name of the filter class supplied in the Python example below is ImageFilter.SHARPEN, and an object of it is created within. The convolution matrix for sharpening is available in ImageFilter.SHARPEN.



**Original sharpened**

# 

# 2. Harris detector

The Harris Corner Detection technique was created to locate an image's internal corners. The corners of an image are defined as the areas where there are considerable fluctuations in gradient intensity in all possible dimensions and directions. Corners extracted can be utilized to extract correct information because they are part of the image features that can be matched with features from other images. Harris Corner Detection is a method for extracting the corners of an input image as well as features from it.

Parameters used :

src – Image Source (Single-channel, 8-bit, or floating-point)

dest – The image that will be used to store the Harris detector answers. The size of the image block is the same as the size of the source image block. Size – The size of the neighborhood (for each pixel value) blockSize \* blockSize of the neighborhood is taken into consideration)

size is a free parameter for the Sobel() operator that controls the aperture.

Free parameter of the Harris detector

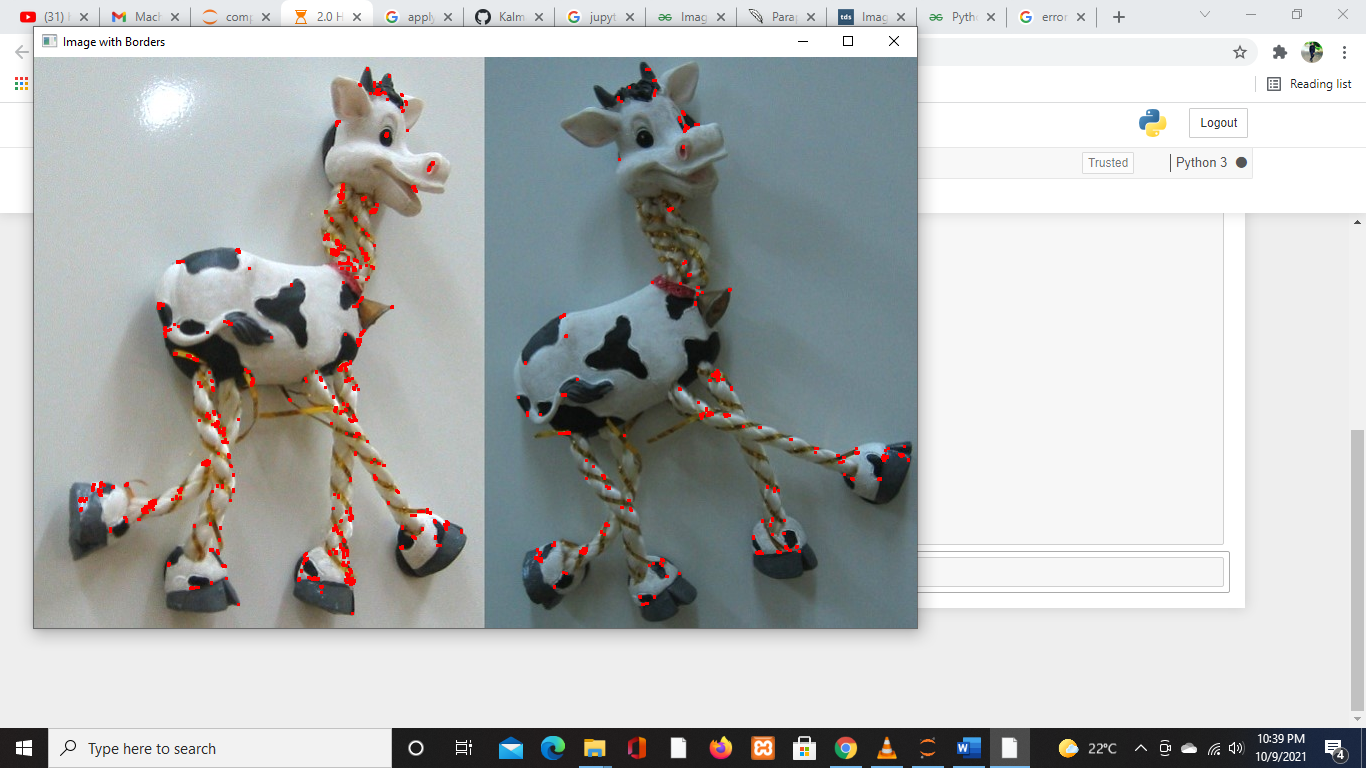
border

Pixel extrapolation method, type ( the extrapolation mode used returns the coordinate of the pixel corresponding to the specified extrapolated pixel )

**Original**



**Transformed**



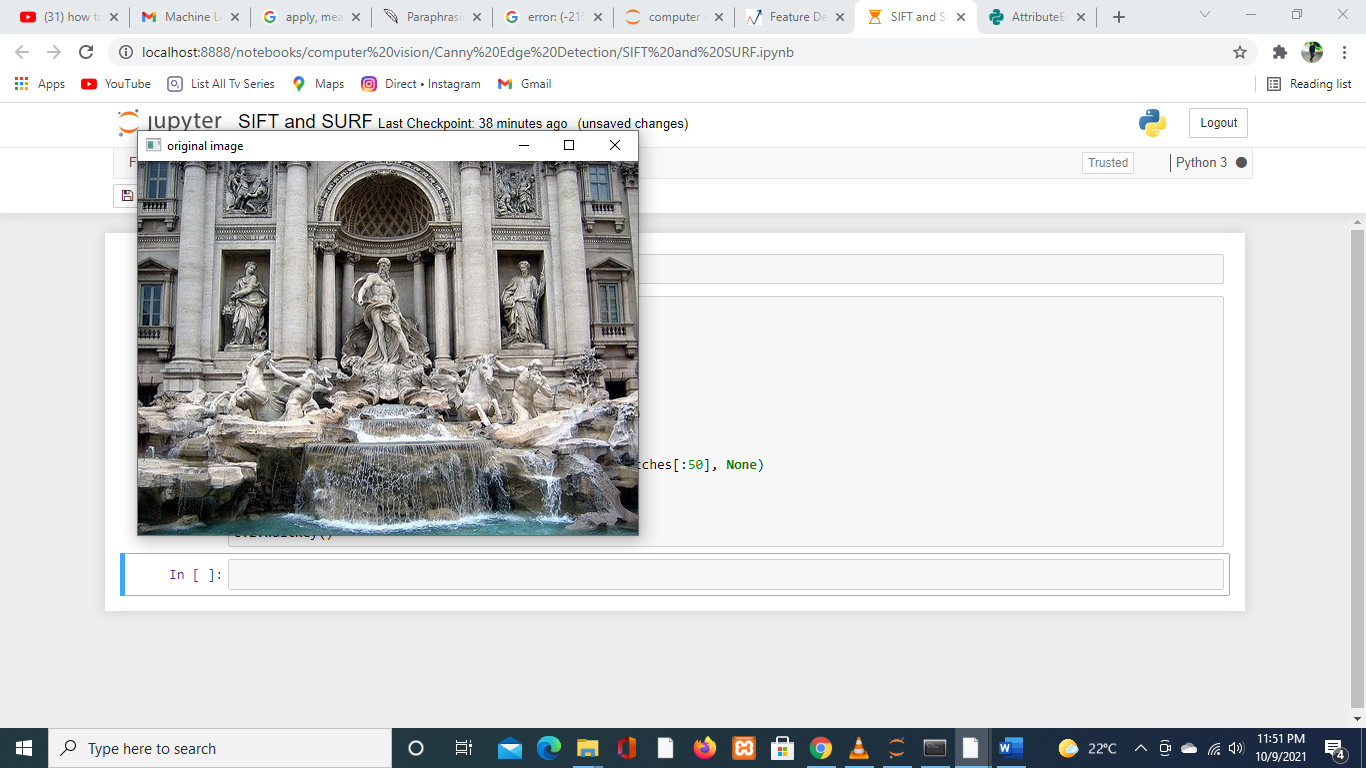
# 3 A. SIFT

The Scale-Invariant Feature Transform (SIFT) was introduced in 2004 by D.Lowe of the University of British Columbia. SIFT stands for image scale and rotation invariance. Because this approach is patented, it is included in OpenCV's Non-free module.

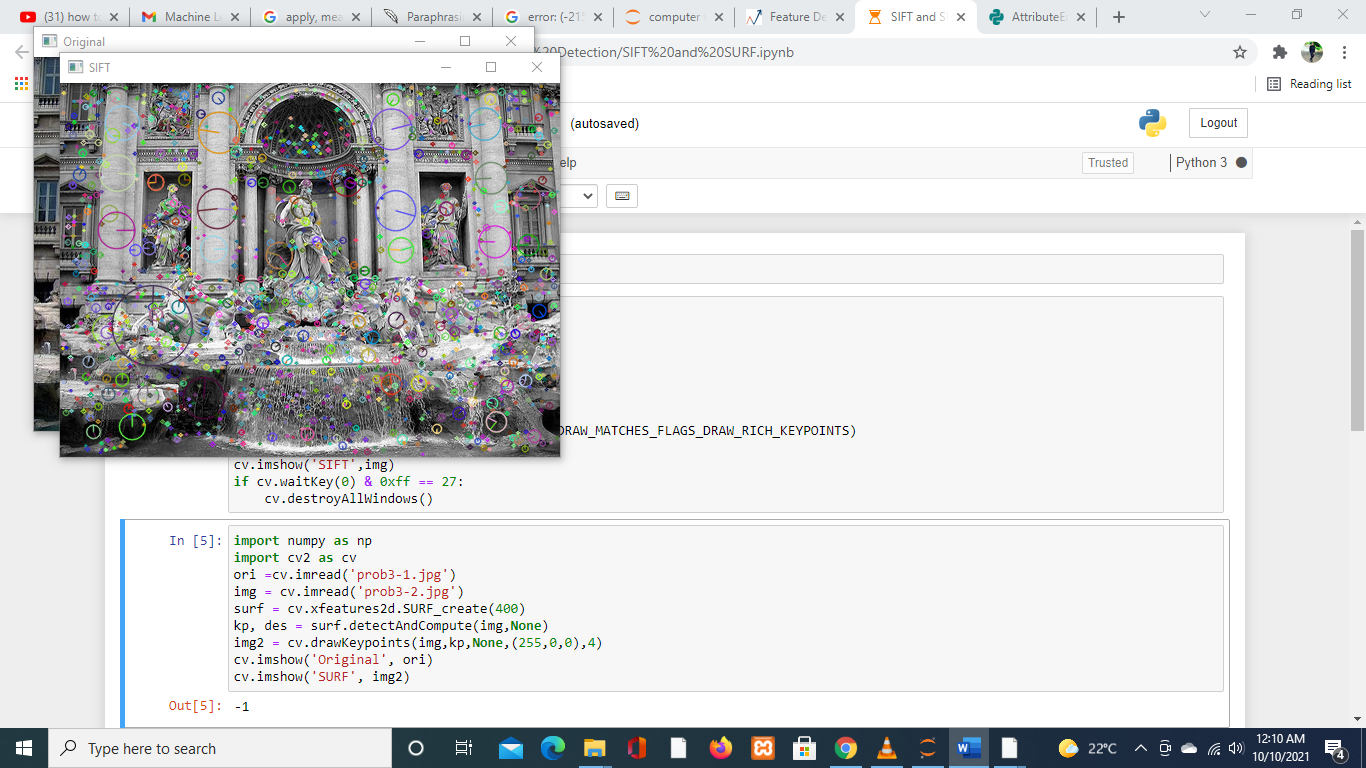
SIFT is a complicated algorithm. The SIFT method consists mostly of four phases.

* Selection of a scale-space peak as a potential place for locating features.
* Keypoint Localization: Locating the feature key points precisely.
* Assignment of Orientation: Assigning orientation to key points.
* The key point descriptor is a high-dimensional vector that describes the key points.
* Matching Keypoints

**Original IMAGE**



**SIFT IMAGE**

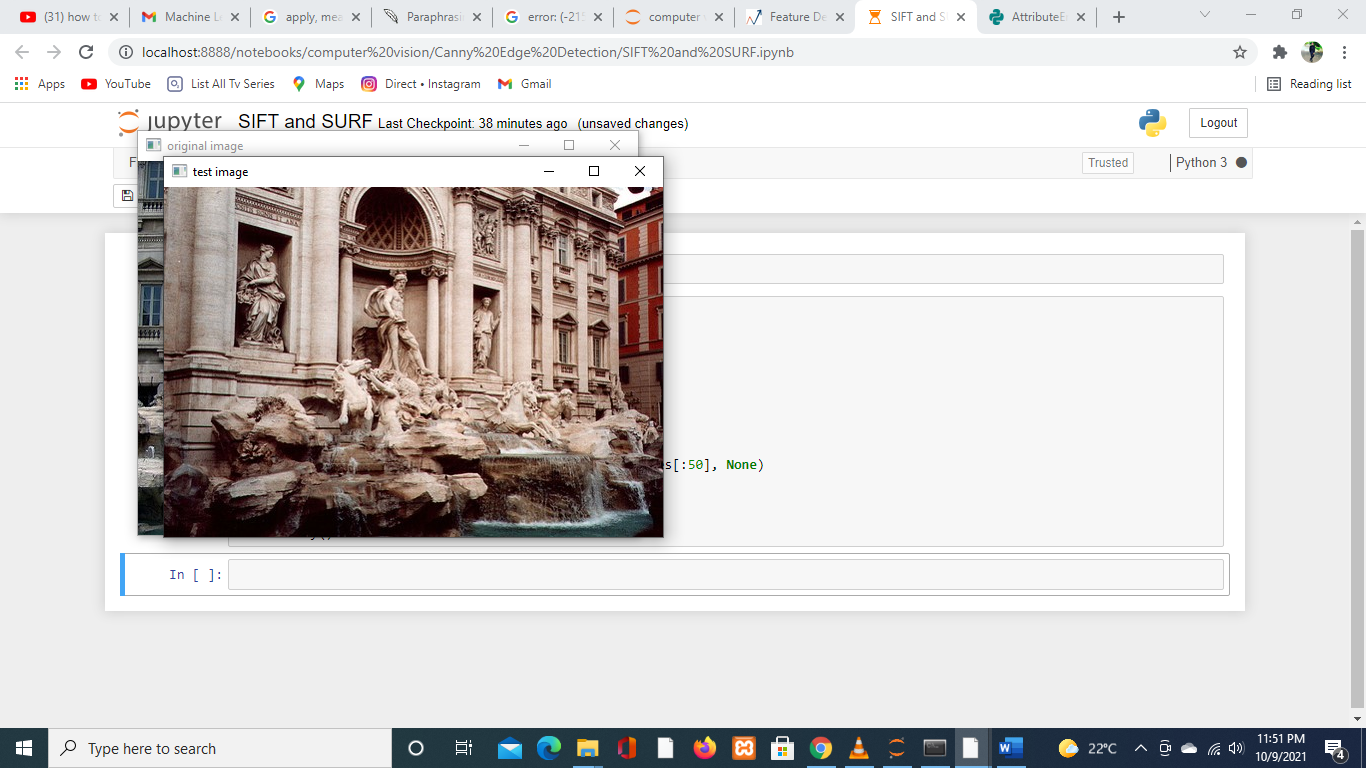


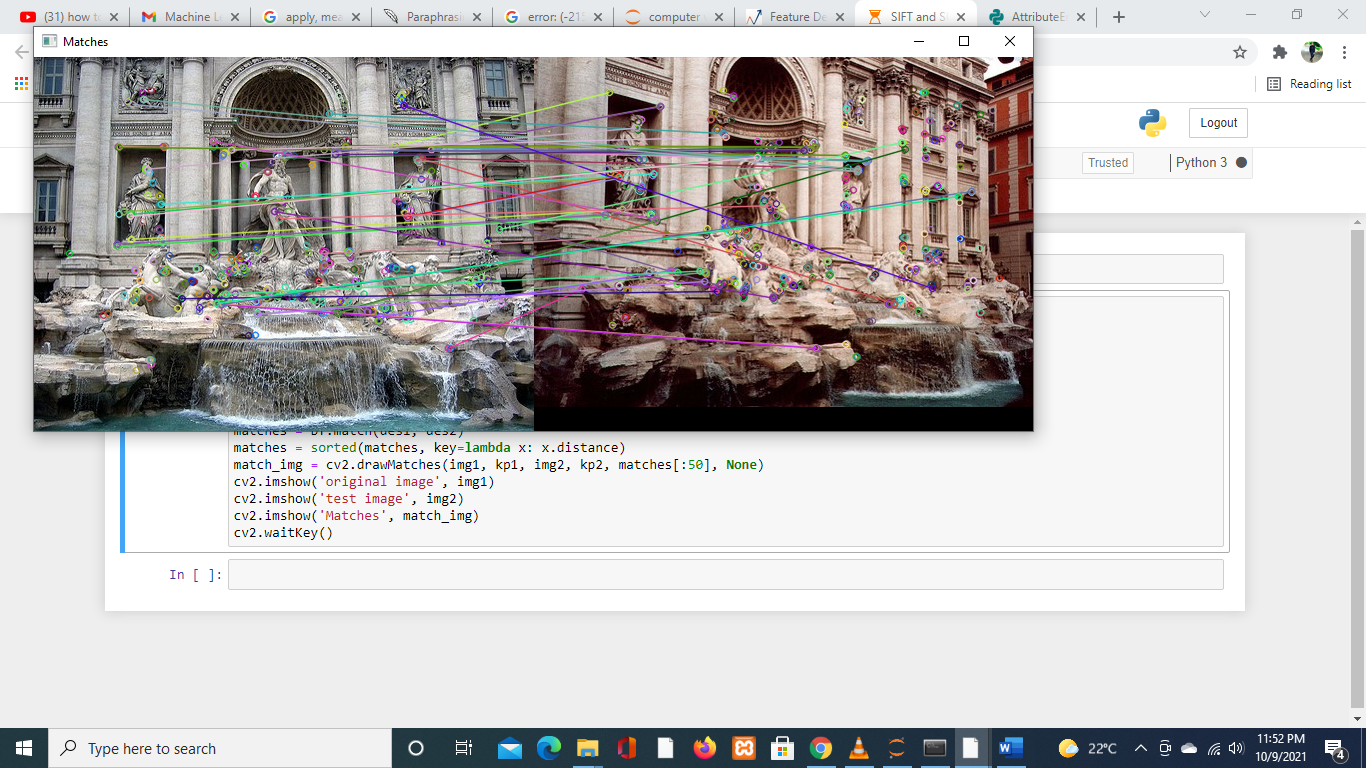
## B. SURF

The SURF method (Speeded Up Robust Features) is a fast and reliable algorithm for representing and comparing images in a local, similarity invariant manner. The fundamental attraction of the SURF technique is its ability to compute operators quickly using box filters, allowing real-time applications like tracking and object recognition. The SURF framework presented in this research is based on H. Bay's Ph.D. thesis [ETH Zurich, 2009], and in particular on a publication co-authored by H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool

Feature matching is like comparing the features of two images which may be different in orientations, perspective, lightening, or even differ in sizes and colors

**Image 2**





# Comparison between SIFT and SURF

SIFT is a feature extraction algorithm that extracts features from photographs. SURF is a fast algorithm that has the same performance as SIFT but with less computing complexity. Although the SIFT algorithm demonstrates its ability in the majority of situations, its performance is still slow. The SURF algorithm is similar to SIFT, however, it is faster and has better performance.

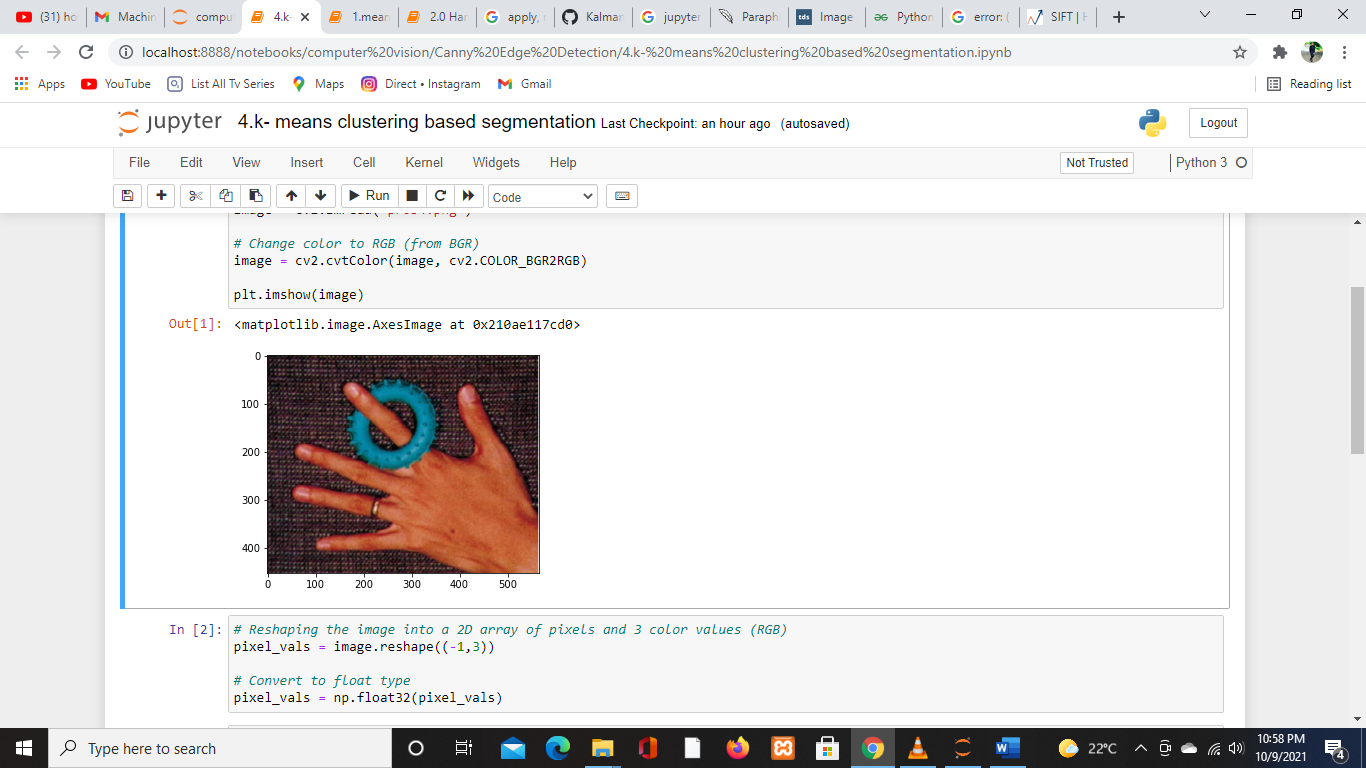
# 4. implement k- means clustering-based segmentation

The clustering algorithm K Means is used. Clustering algorithms are unsupervised algorithms, meaning they don't use labeled data. It is used to distinguish between different classes or clusters of data depending on how similar the data is. Data points from the same group are more similar to each other than data points from other groups.

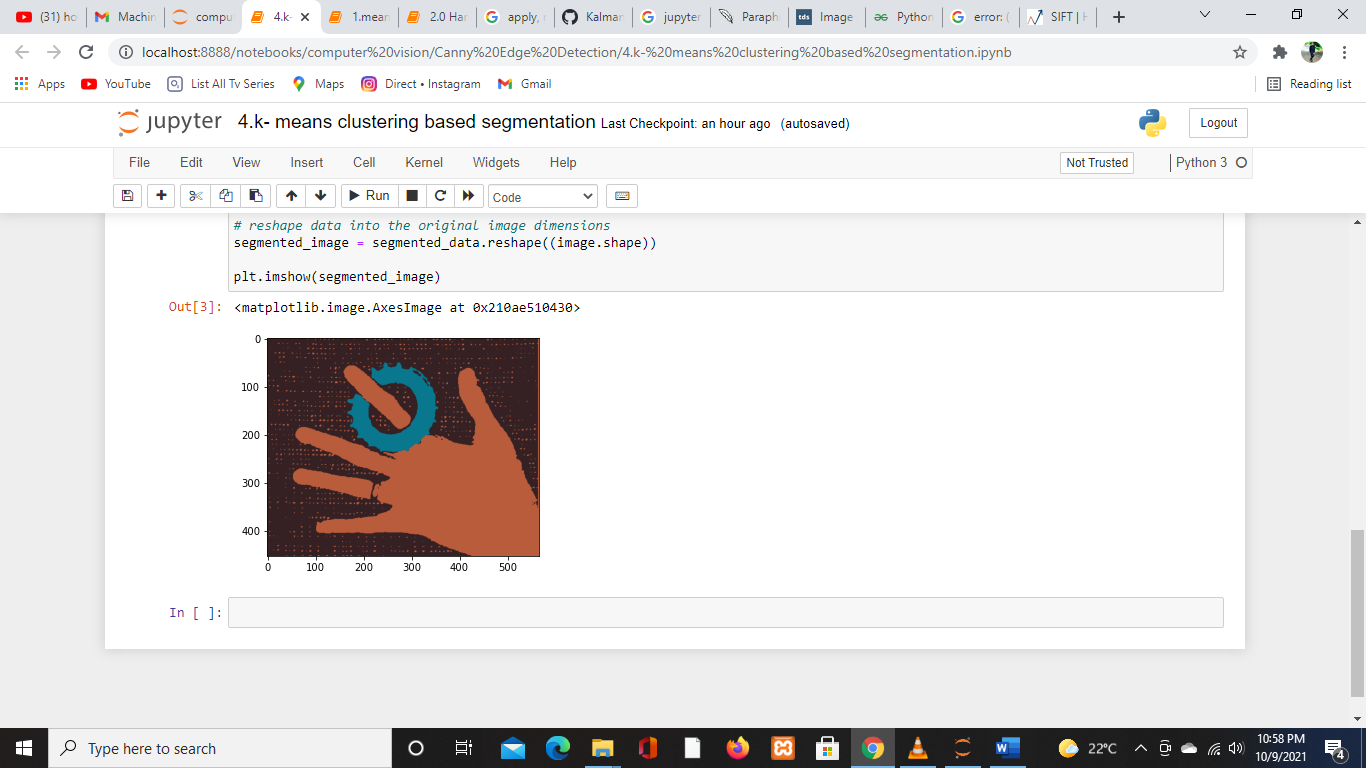
One of the most widely used clustering methods is K-means clustering. The number of clusters is represented by k.

1. Let's have a look at how K-means clustering works Select k as the number of clusters you want to find.
2. Assign the data points to one of the k clusters at random.
3. Then figure out where the clusters' centers are.
4. Calculate the distance between each cluster's center and the data points.
5. Reassign the data points to the clusters closest to them, based on their distance from the cluster.
6. Calculate the new cluster center once more.
7. Repeat steps 4,5 and 6 until the data points do not affect the clusters or the number of iterations is reached.

**Original**



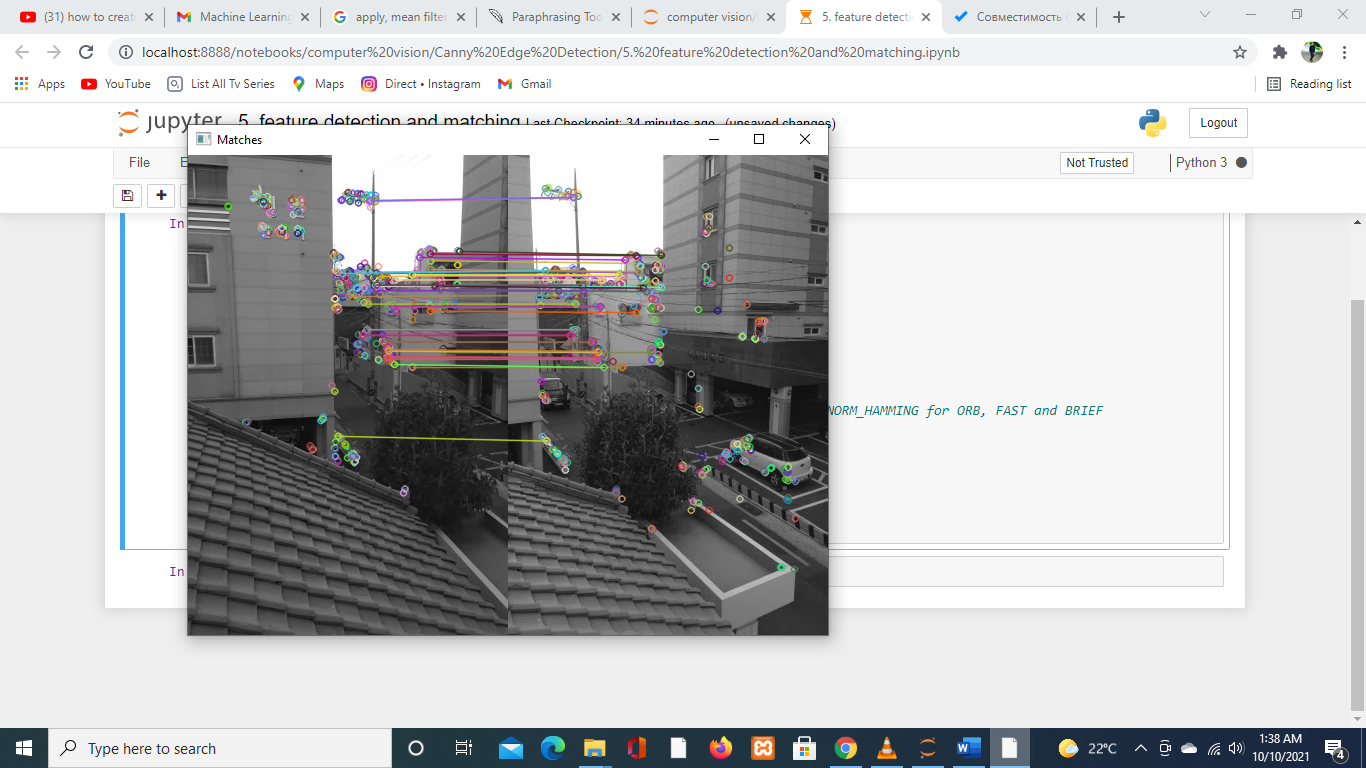
**Transformed**



# 5 Implement feature detection and matching

In OpenCV, you can use a Brute-Force matcher or a FLANN-based matcher to match features between pictures.

BF Matcher compares the descriptor of a feature in one image to the descriptors of all other features in another image and provides the match depending on the distance. It takes a long time because it compares all of the qualities.



# References

Python | Corner detection with Harris Corner Detection method using OpenCV **Last Updated:** 21 Jan 2019

<https://www.geeksforgeeks.org/python-corner-detection-with-harris-corner-detection-method-using-opencv/>

Image Filters in Python [Manvir Sekhon](https://medium.com/@m4nv1r5?source=post_page-----26ee938e57d2--------------------------------)

<https://towardsdatascience.com/image-filters-in-python-26ee938e57d2>

Sharpen-filter Using Pillow - The Python Image Processing Library

<https://pythontic.com/image-processing/pillow/sharpen-filter>

Introduction to SURF (Speeded-Up Robust Features)

<https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e>

Introduction to SIFT( Scale Invariant Feature Transform)

<https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>