



Python OpenCV – Canny() Function

Last Updated : 03 Jan, 2023

In this article, we will see the **Canny Edge** filter in OpenCV. Canny() Function in OpenCV is used to detect the edges in an image.

Syntax: `cv2.Canny(image, T_lower, T_upper, aperture_size, L2Gradient)`

Where:

- *Image: Input image to which Canny filter will be applied*
- *T_lower: Lower threshold value in Hysteresis Thresholding*
- *T_upper: Upper threshold value in Hysteresis Thresholding*
- *aperture_size: Aperture size of the Sobel filter.*
- *L2Gradient: Boolean parameter used for more precision in calculating Edge Gradient.*

Canny Edge detection is an Algorithm consisting of 4 major steps:

- Reduce Noise using Gaussian Smoothing.
- Compute image gradient using Sobel filter.
- Apply Non-Max Suppression or NMS to just keep the local maxima
- Finally, apply Hysteresis thresholding which takes 2 threshold values T_upper and T_lower which is used in the Canny() function.

Input Image:



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



Basic example of Canny() function

Python3

```
import cv2

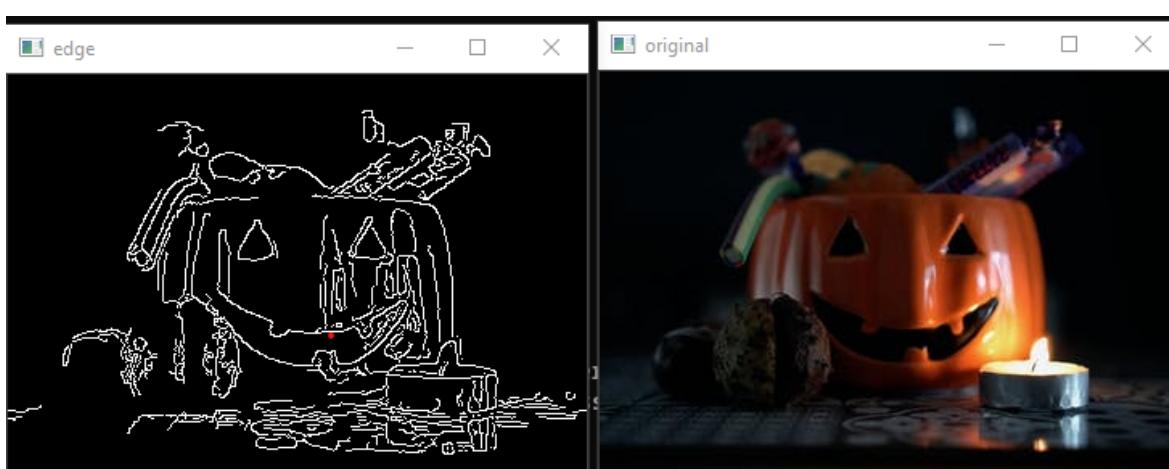
img = cv2.imread("test.jpeg") # Read image

# Setting parameter values
t_lower = 50 # Lower Threshold
t_upper = 150 # Upper threshold

# Applying the Canny Edge filter
edge = cv2.Canny(img, t_lower, t_upper)

cv2.imshow('original', img)
cv2.imshow('edge', edge)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:



Canny() function with Aperture_size

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

value is 3 and its value should be odd between 3 and 7. You can increase the Aperture size when you want to detect more detailed features.

Python3

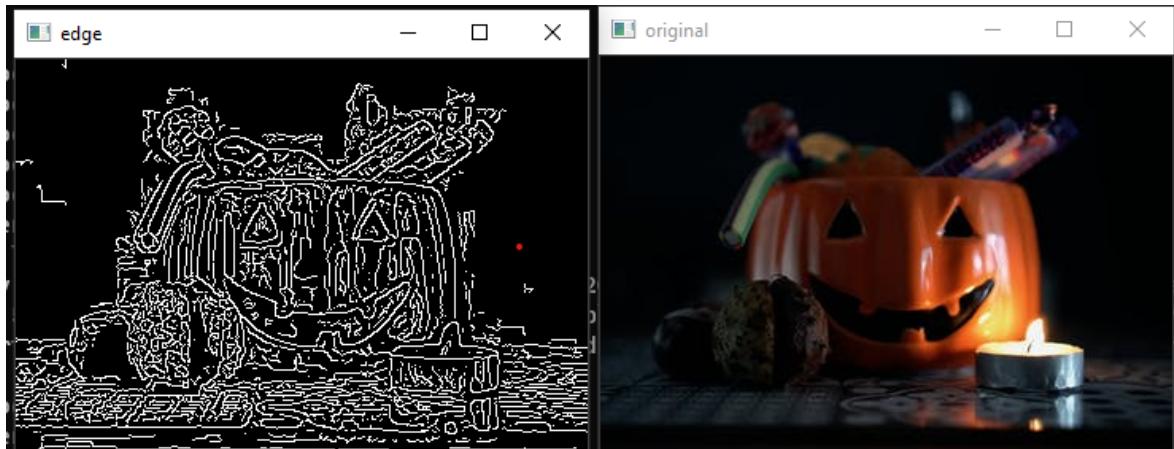
```
import cv2

img = cv2.imread("test.jpeg") # Read image

# Setting All parameters
t_lower = 100 # Lower Threshold
t_upper = 200 # Upper threshold
aperture_size = 5 # Aperture size

# Applying the Canny Edge filter
# with Custom Aperture Size
edge = cv2.Canny(img, t_lower, t_upper,
                  apertureSize=aperture_size)
cv2.imshow('original', img)
cv2.imshow('edge', edge)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:



Canny() function with L2Gradient

It's a boolean parameter that specifies if you want to calculate the usual gradient equation or the L2Gradient algorithm. Again, it's an optional parameter. L2gradient is nothing my $\sqrt{\text{gradient}_x^2 + \text{gradient}_y^2}$

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Python3

```
import cv2

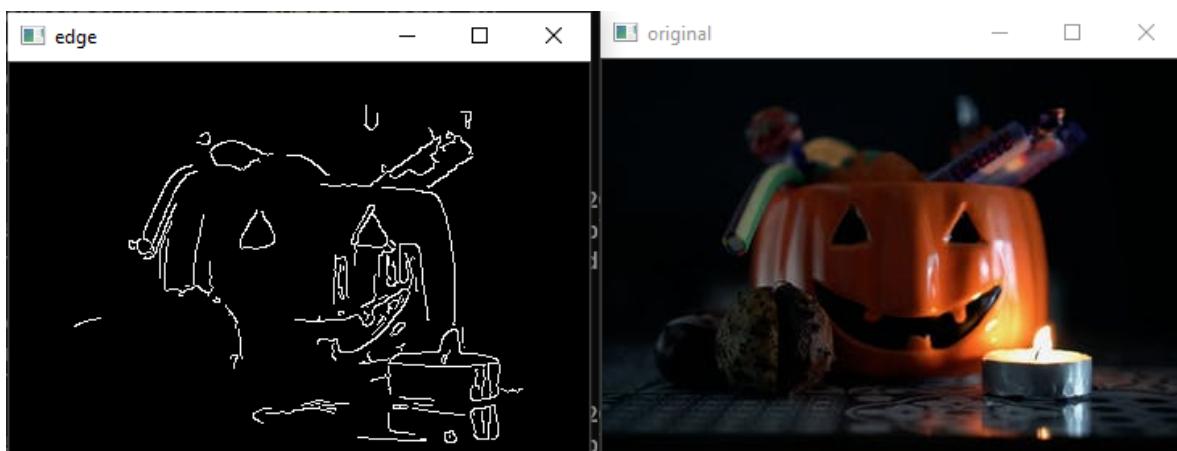
img = cv2.imread("test.jpeg") # Read image

t_lower = 100 # Lower Threshold
t_upper = 200 # Upper threshold
aperture_size = 5 # Aperture size
L2Gradient = True # Boolean

# Applying the Canny Edge filter with L2Gradient = True
edge = cv2.Canny(img, t_lower, t_upper, L2gradient = L2Gradient )

cv2.imshow('original', img)
cv2.imshow('edge', edge)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:



Canny() function with both Aperture size and L2gradient

Here we will use both attributes within the function.

Python3

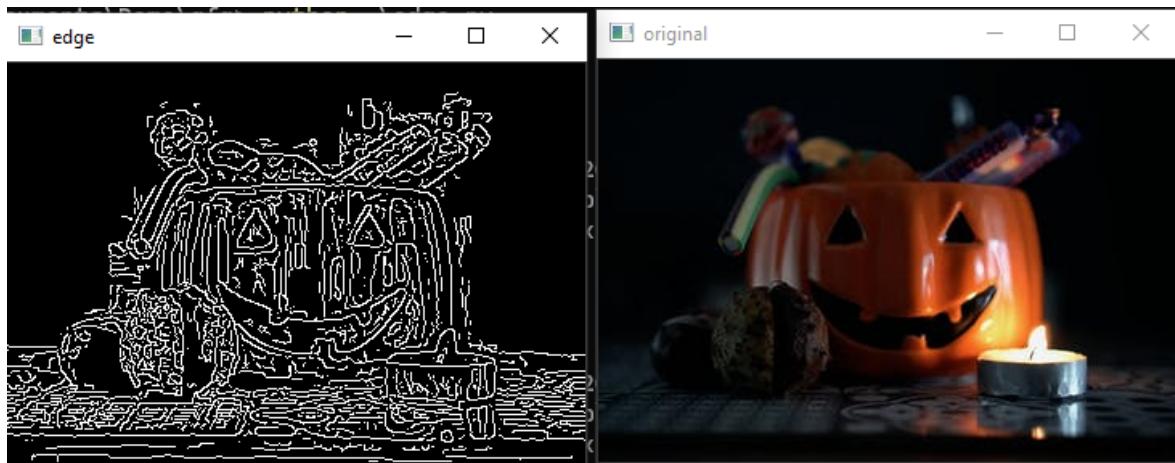
```
import cv2
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
t_upper = 200 # Upper threshold  
aperture_size = 5 # Aperture size  
L2Gradient = True # Boolean  
  
# Applying the Canny Edge filter  
# with Aperture Size and L2Gradient  
edge = cv2.Canny(img, t_lower, t_upper,  
                  apertureSize = aperture_size,  
                  L2gradient = L2Gradient )  
  
cv2.imshow('original', img)  
cv2.imshow('edge', edge)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

Output:



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

[Previous Article](#)

Python OpenCV: Optical Flow with Lucas-Kanade method

[Next Article](#)

Python OpenCV - getRotationMatrix2D() Function

Similar Reads

Real-Time Edge Detection using OpenCV in Python | Canny edge...

Edge detection is one of the fundamental image-processing tasks used in various Computer Vision tasks to identify the boundary or sharp changes in...

5 min read

Implement Canny Edge Detector in Python using OpenCV

In this article, we will learn the working of the popular Canny edge detection algorithm developed by John F. Canny in 1986. Usually, in Matlab and...

5 min read

Wand canny() function - Python

The canny() function is an inbuilt function in the Python Wand ImageMagick library which is used to detect edges by leveraging a multi-stage Canny...

2 min read

OpenCV - Facial Landmarks and Face Detection using dlib and OpenCV

Content has been removed on Author's request.

1 min read

Transition from OpenCV 2 to OpenCV 3.x

OpenCV is one of the most popular and most used Computer vision libraries. It contains tools to carry out image and video processing. When OpenCV...

2 min read

Top Books for Learning OpenCV: Computer Vision with OpenCV Library

OpenCV or Open Source Computer Vision Library, is an open-source

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Automatic Document Scanner using OpenCV (OpenCV Document...)

An automatic document scanner using OpenCV is a computer vision application that automatically detects and extracts documents from images...

6 min read

Python OpenCV - setWindowTitle() Function

Python OpenCV setWindowTitle() method used for giving the title of the windows. It takes 2 parameters that are windows name and the title that...

1 min read

Python OpenCV - imencode() Function

Python OpenCV imencode() function converts (encodes) image formats into streaming data and stores it in-memory cache. It is mostly used to compress...

2 min read

Python OpenCV - resizeWindow() Function

resizeWindow() method in Python OpenCV is used to resize window displaying images/videos to a specific size. The specified window size is for...

1 min read

Python OpenCV - waitKey() Function

waitkey() function of Python OpenCV allows users to display a window for given milliseconds or until any key is pressed. It takes time in milliseconds a...

1 min read

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

also returns the full key code. The key code which is returned is...

2 min read

...

Python OpenCV - getRotationMatrix2D() Function

cv2.getRotationMatrix2D() function is used to make the transformation matrix M which will be used for rotating a image. Syntax:...

3 min read

Python OpenCV - destroyAllWindows() Function

Python Opencv destroyAllWindows() function allows users to destroy or close all windows at any time after exiting the script. If you have multiple...

2 min read

Python OpenCV - namedWindow() Function

Python OpenCV namedWindow() method is used to create a window with a suitable name and size to display images and videos on the screen. The...

3 min read

Python OpenCV - selectROI() Function

In this article, we are going to see an interesting application of the OpenCV library, which is selectROI(). With this method, we can select a range of...

3 min read

Python OpenCV - imdecode() Function

Python cv2.imdecode() function is used to read image data from a memory cache and convert it into image format. This is generally used for loading th...

2 min read

Python OpenCV - getTrackbarPos() Function

getTrackbarPos() is Function in Python OpenCV that returns the current position of the specified trackbar. It takes two arguments. The first is for the...

2 min read

Python OpenCV - setTrackbarPos() Function

setTrackbarPos() function sets the position of the specified trackbar in the trackbar window.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Python OpenCV - getgaussiankernel() Function

Python OpenCV getGaussianKernel() function is used to find the Gaussian filter coefficients. The Gaussian kernel is also used in Gaussian Blurring....

3 min read

Article Tags :[Python](#)[OpenCV](#)[Python-OpenCV](#)**Practice Tags :**[python](#)

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh Nagar,
Uttar Pradesh, 201305

**Company**[About Us](#)[Legal](#)**Languages**[Python](#)[Java](#)

We use cookies to ensure you have the best browsing experience on our website. By

using our site, you acknowledge that you have read and understood our [Cookie Policy](#) &

Got It !

[Privacy Policy](#)

GFG Corporate Solution
 Placement Training Program
 GeeksforGeeks Community

SQL
 R Language
 Android Tutorial
 Tutorials Archive

DSA

Data Structures
 Algorithms
 DSA for Beginners
 Basic DSA Problems
 DSA Roadmap
 Top 100 DSA Interview Problems
 DSA Roadmap by Sandeep Jain
 All Cheat Sheets

Data Science & ML
 Data Science With Python
 Data Science For Beginner
 Machine Learning
 ML Maths
 Data Visualisation
 Pandas
 NumPy
 NLP
 Deep Learning

Web Technologies

HTML
 CSS
 JavaScript
 TypeScript
 ReactJS
 NextJS
 Bootstrap
 Web Design

Python Tutorial

Python Programming Examples
 Python Projects
 Python Tkinter
 Web Scraping
 OpenCV Tutorial
 Python Interview Question
 Django

Computer Science

Operating Systems
 Computer Network
 Database Management System
 Software Engineering
 Digital Logic Design
 Engineering Maths
 Software Development
 Software Testing

DevOps

Git
 Linux
 AWS
 Docker
 Kubernetes
 Azure
 GCP
 DevOps Roadmap

System Design

High Level Design
 Low Level Design
 UML Diagrams
 Interview Guide
 Design Patterns
 OOAD
 System Design Bootcamp
 Interview Questions

Interview Preparation

Competitive Programming
 Top DS or Algo for CP
 Company-Wise Recruitment Process
 Company-Wise Preparation
 Aptitude Preparation
 Puzzles

School Subjects

Mathematics

GeeksforGeeks Videos

DSA

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Social Science
English Grammar
Commerce
World GK

Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !