Image Processing

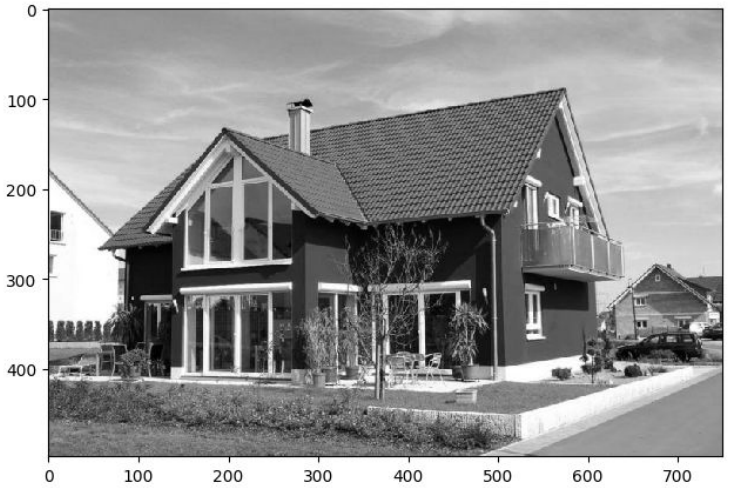
Read Image as RGB

```
from skimage.io import imread, imshow
image1 = imread('house.jpg')
imshow(image1)
```

Read image as Grayscale

```
image2 = imread('house.jpg', as_gray=True)
imshow(image2)
print ((image2))
```

```
[[0.64879843 0.64879843 0.64879843 ... 0.59284784 0.5903251  0.5903251 ]
 [0.64879843 0.64879843 0.65272    ... 0.59676941 0.5903251  0.5903251 ]
 [0.64741451 0.64741451 0.64741451 ... 0.60069098 0.59424667 0.59424667]
 ...
 [0.46538824 0.45949451 0.45443412 ... 0.55186275 0.55578431 0.55186275]
 [0.44126275 0.47038039 0.44235608 ... 0.55045608 0.54962275 0.54962275]
 [0.43311412 0.4605651  0.43477294 ... 0.55045608 0.54962275 0.54962275]]
```



Shape and Size

```
print(image1.shape)
print(image2.shape)
```

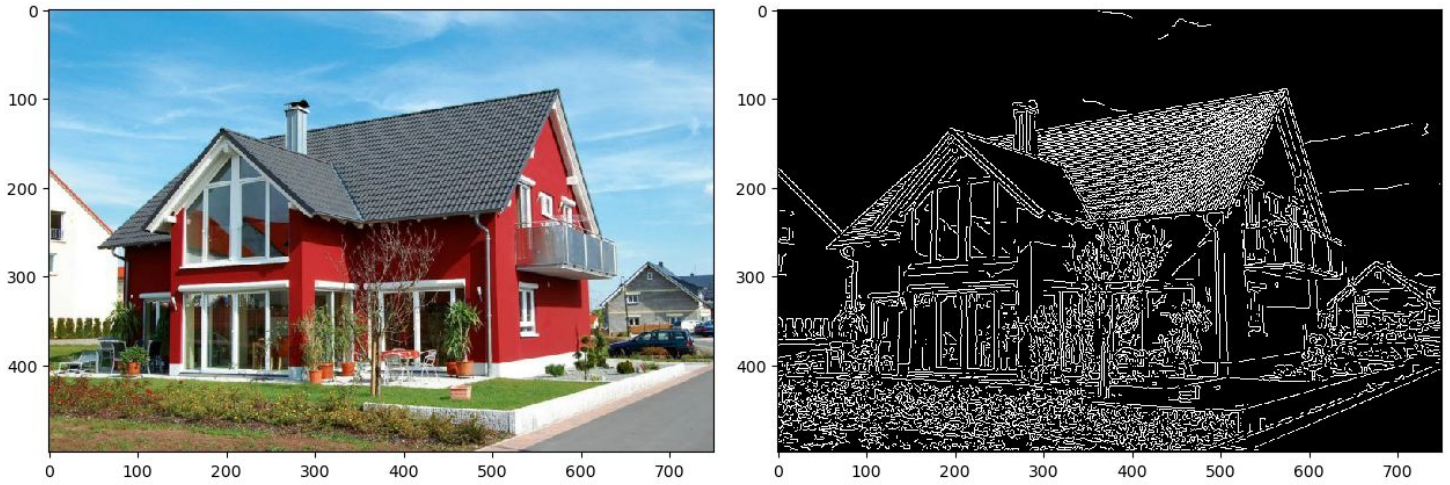
```
(498, 750, 3)
(498, 750)
```

```
print(image1.size)
print(image2.size)
```

```
1120500
373500
```

Edge detection

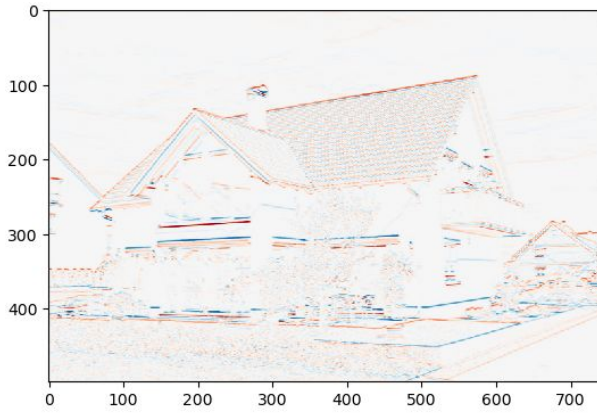
- Edge detection is done by searching for pixels where there is a sharp change in intensity or color.
- It is very useful in reducing the amount of data to be processed by filtering out irrelevant data while keeping important structural features of an image.



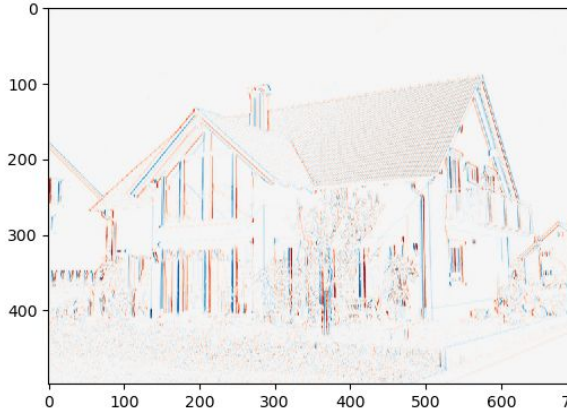
Sobel and Prewitt Filters:

- An edge detector is essentially a high-pass filter that allows high-frequency signals to go through and low-frequency signals to be suppressed.
- Edges of objects are the high-frequency components, so when we apply an edge detection filter to an image, we amplify the edges and suppress the rest.

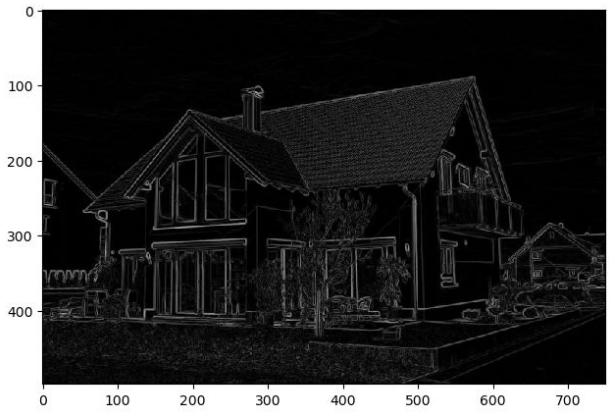
```
from skimage import filters
from skimage import feature
from skimage.filters import prewitt_h,prewitt_v
# prewitt kernel
pre_hor = prewitt_h(image2)
pre_ver = prewitt_v(image2)

# Sobel Kernel
ed_sobel = filters.sobel(image2)
```



prewitt_h



prewitt_v



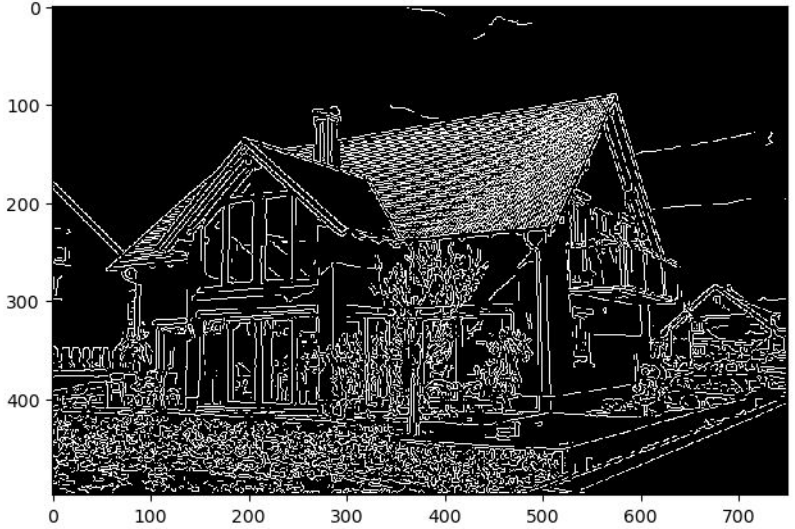
sobel

Canny Edge Detection

- Canny edge detection technique is considered to be the standard edge detection method in image processing
- Provides the highest accuracy in edge detection in shorter time than the other techniques.

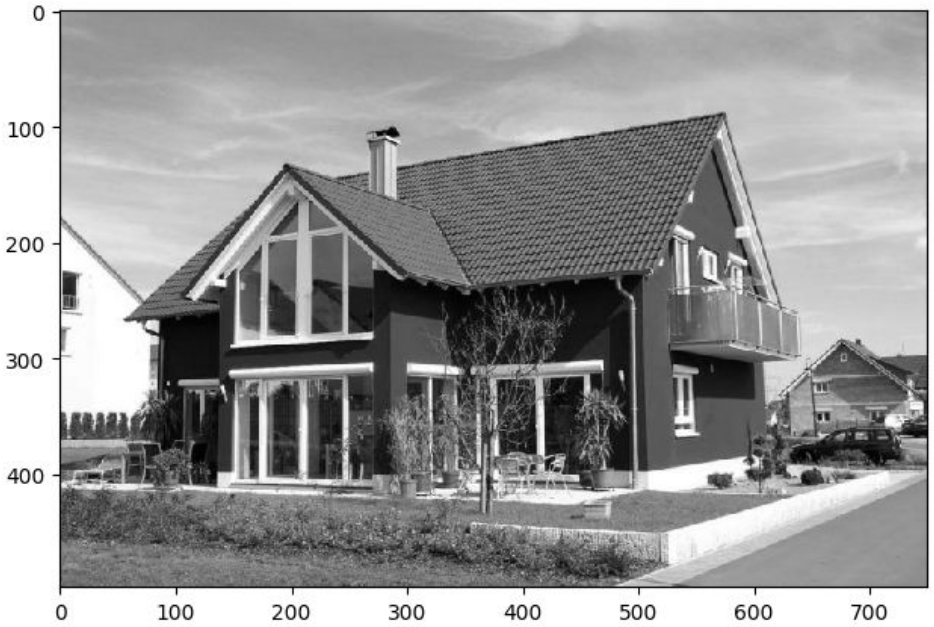
```
#canny filter
from skimage import feature
can = feature.canny(image2)
imshow(can)
```


0
100
200
300
400
0
100
200
300
400
500
600
700


0
100
200
300
400
0
100
200
300
400
500
600
700

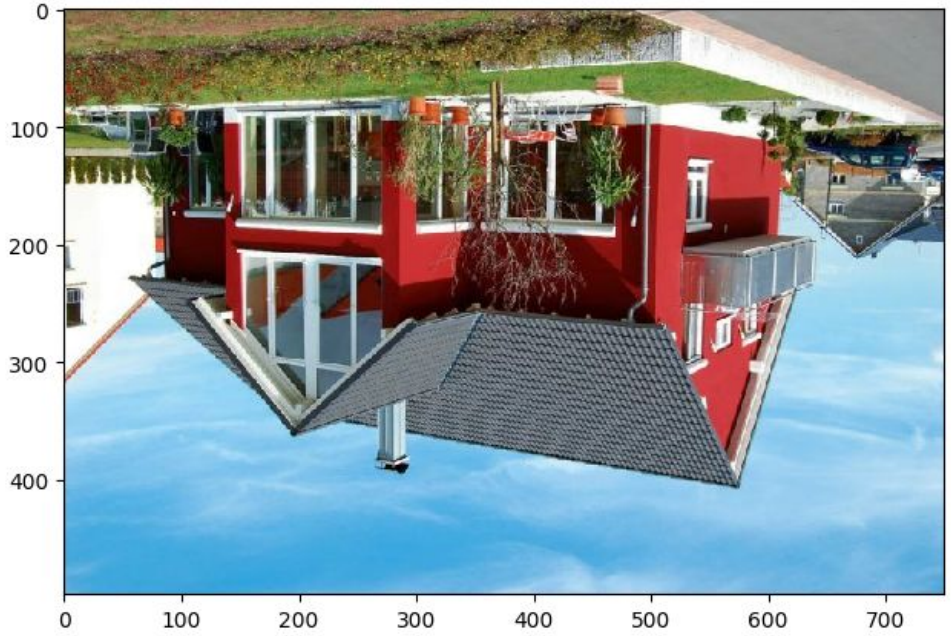
RGB to GRAY

```
from skimage import color
grayscale = color.rgb2gray(image1)

imshow(grayscale)
plt.show()
```



Flip Image

```
# Flip the image in up direction
verticalflip = np.flipud(image1)
imshow(verticalflip)
plt.show()
```



OpenCV

```
import cv2
from google.colab.patches import cv2_imshow
image = cv2.imread('house.jpg')
cv2_imshow(image)
```

Grayscale

```
# Display the grayscale image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2_imshow( gray_image)
```