

RGB Image

```
from PIL import Image, ImageFilter
```

```
image = Image.open("car.jpeg")
```

```
image
```



Edge Detection

```
image2 = image.convert("L")
```

```
image2 = image2.filter(ImageFilter.FIND_EDGES)
```

```
image2.save(r"Edge_Sample.png")
```

```
image2
```



RGB to GrayScale Conversion

```
from PIL import ImageOps
```

```
gray_image = ImageOps.grayscale(image)
```

```
gray_image
```



HOG - Histogram of Oriented Gradients

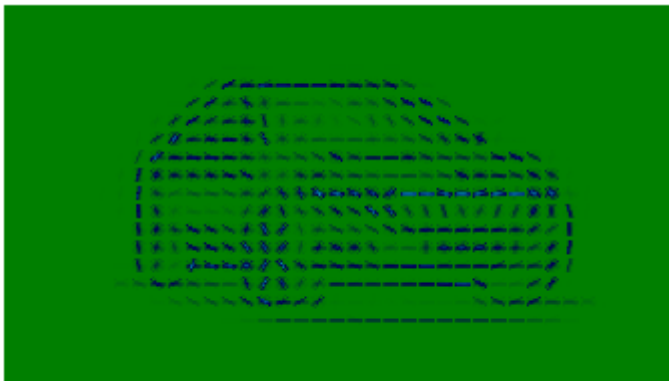
```
#importing required libraries
from skimage.io import imread
from skimage.transform import resize
from skimage.feature import hog
from skimage import exposure
import matplotlib.pyplot as plt

# reading the image
img = imread('car.jpeg')
plt.axis("off")
plt.imshow(img)
print(img.shape)

resized_img = img
# resized_img = resize(img, (64*4, 100*4))
# plt.axis("off")
# plt.imshow(resized_img)
# print(resized_img.shape)

#creating hog features
fd, hog_image = hog(resized_img, orientations=9, pixels_per_cell=(8, 8),
                    cells_per_block=(2, 2), visualize=True, multichannel=True)
plt.axis("off")
plt.imshow(hog_image, cmap="ocean")
```

```
(169, 299, 3)
<matplotlib.image.AxesImage at 0x7f2c58af9e90>
```



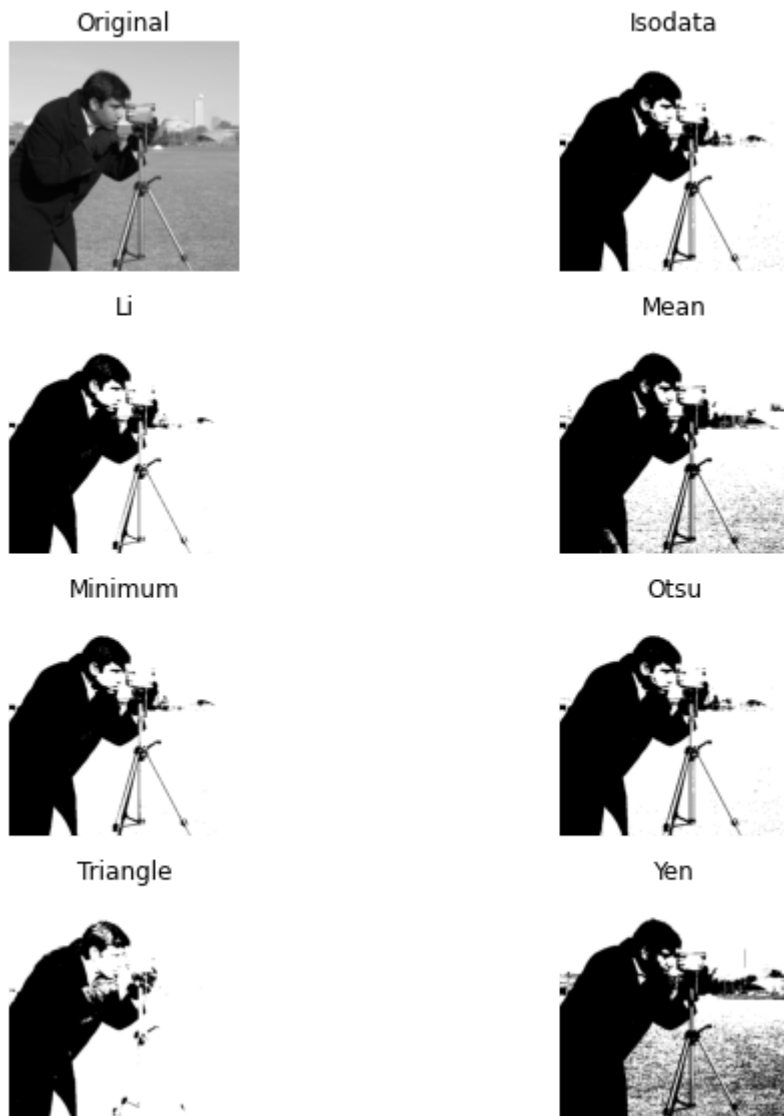
Treshholds

```
import matplotlib
import matplotlib.pyplot as plt

from skimage import data
from skimage.filters import try_all_threshold

img = data.camera()

fig, ax = try_all_threshold(img, figsize=(10, 8), verbose=False)
plt.show()
```



✓ 2s completed at 4:29 PM

● ✕