# edge-detection

November 2, 2023

[29]:
```
import cv2
import matplotlib.pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

[30]:
```
img = cv2.imread('Dataset/cat6.jpg')
```

[31]:
```
def imageMain(imgtitle,image):
    imgVer = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    plt.figure(figsize=(5,3))
    plt.imshow((imgVer).astype(np.uint8))
    plt.title(imgtitle)
    plt.axis('off')
    plt.grid(False)
    plt.show()
```

Real Images

[32]:
```
imageMain("PussyCat", img)
```

Gray Image

[33]:
```
gray = cv2.cvtColor(img ,cv2.COLOR_RGB2GRAY)
```

[34]:
```
imageMain("Gray", gray)
```


Edge Detection — CANNY Edge Detection

[35]:
```
canny = cv2.Canny(img, 150, 250)
```

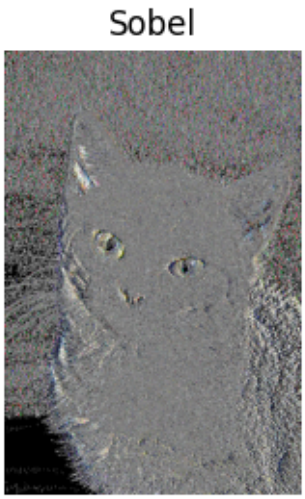
[36]:
```
imageMain("Canny 1", canny)
```

[37]:
```
canny = cv2.Canny(img, 100, 150)
imageMain("Canny 2", canny)
```


Edge Detection — SOBEL Edge Detection

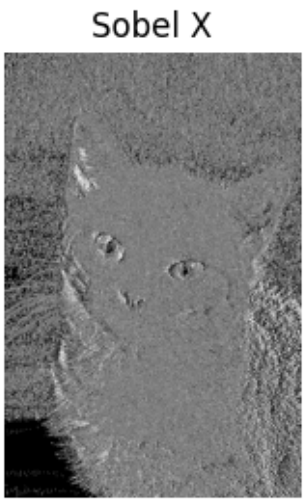
[38]:
```
sobel = cv2.Sobel(img, cv2.CV_32F, 1, 0)
```

[39]:
```
imageMain("Sobel", sobel)
```




[40]:
```
sobelX = cv2.Sobel(gray, cv2.CV_32F, 1, 0, ksize=3)
```

[41]:
```
imageMain("Sobel X", sobelX)
```



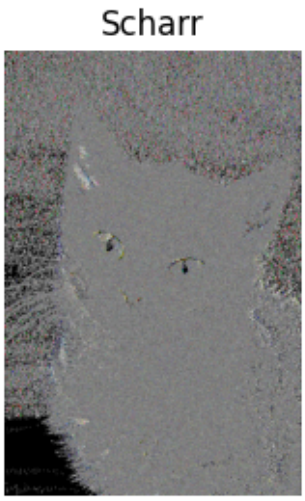

[42]:
```
sobelY = cv2.Sobel(gray, cv2.CV_32F, 0, 1, ksize=3)
```

[43]:
```
imageMain("SobelY", sobelY)
```


Edge Detection — SCHARR Edge Detection

[44]:
```
scharr = cv2.Scharr(img, cv2.CV_32F, 1, 0)
```

[45]:
```
imageMain("Scharr", scharr)
```




Edge Detection — LAPLACIAN Edge Detection

[46]:
```
laplacian = cv2.Laplacian(img, cv2.CV_32F)
```

[47]:
```
imageMain("Laplacian", laplacian)
```


Edge Detection —ZERO CROSSING

[48]:
```
img = cv2.imread('Dataset/cat6.jpg',0) # 0 indicates that the image should be␣
↪loaded in grayscale mode.
```

[49]:
```
blur = cv2.GaussianBlur(img, (3, 3), 0)
```

[50]:
```
laplacian = cv2.Laplacian(blur, cv2.CV_64F)
```

[51]:
```
edges = cv2.threshold(np.abs(laplacian), 30, 250, cv2.THRESH_BINARY)[1]
```

[52]:
```
plt.imshow(edges, cmap='gray')
plt.title('Zero Crossing')
plt.axis('off')
plt.show()
```


Edge Detection —CANNY-DERICHE

[53]:
```
blurred_img = cv2.GaussianBlur(img, (5, 5), 0)
```

[54]:
```
low_threshold = 50  # Adjust this threshold based on the image and noise level
high_threshold = 150  # Typically set to a ratio of 1:3 or 1:2 to low_threshold
```

[55]:
```
edgess = cv2.Canny(blurred_img, low_threshold, high_threshold)
```

[56]:
```
imageMain("Canny-Deriche", edgess)
```


[ ]:

Github: https://github.com/mdnuruzzamanKALLOL

Kaggle: https://www.kaggle.com/nuruzzamankallol