

PROCESSAMENTO DIGITAL DE IMAGENS PARA IDENTIFICAÇÃO DE VEGETAÇÃO SOB LINHAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

Acadêmico: Marco Aurélio Oliveira Rocha

Orientador: Dr. Anderson Diogo Spacek

Sumário

1. Introdução
2. Revisão bibliográfica
3. Procedimentos Experimentais
4. Resultados e Discussões
5. Conclusão
6. Referências Bibliográficas



1. Introdução

Definição do problema

- Interrupções no fornecimento de energia elétrica
- Contato da vegetação com os condutores de eletricidade



1. Introdução

Definição do problema



Fonte: GOVERNO DO ESTADO DE SANTA CATARINA (2018).



Fonte: HEDMOND (2017).

1. Introdução

Objetivos

Utilização de processamento digital de imagens como meio auxiliar para monitoramento da vegetação próxima a rede elétrica

- Implementação de um algoritmo capaz de realizar a identificação da rede elétrica e da vegetação presente no local



2. Revisão bibliográfica

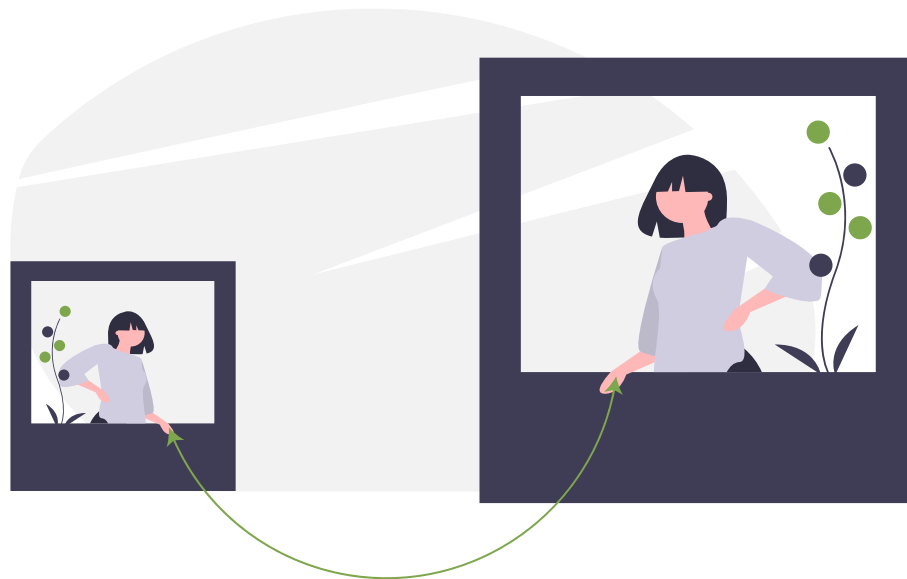
Meios utilizados para monitoramento da vegetação

- Inspeção de campo (visual)
- Inspeção aérea (visual/vídeo)
- Utilização de métodos de sensoriamento remoto



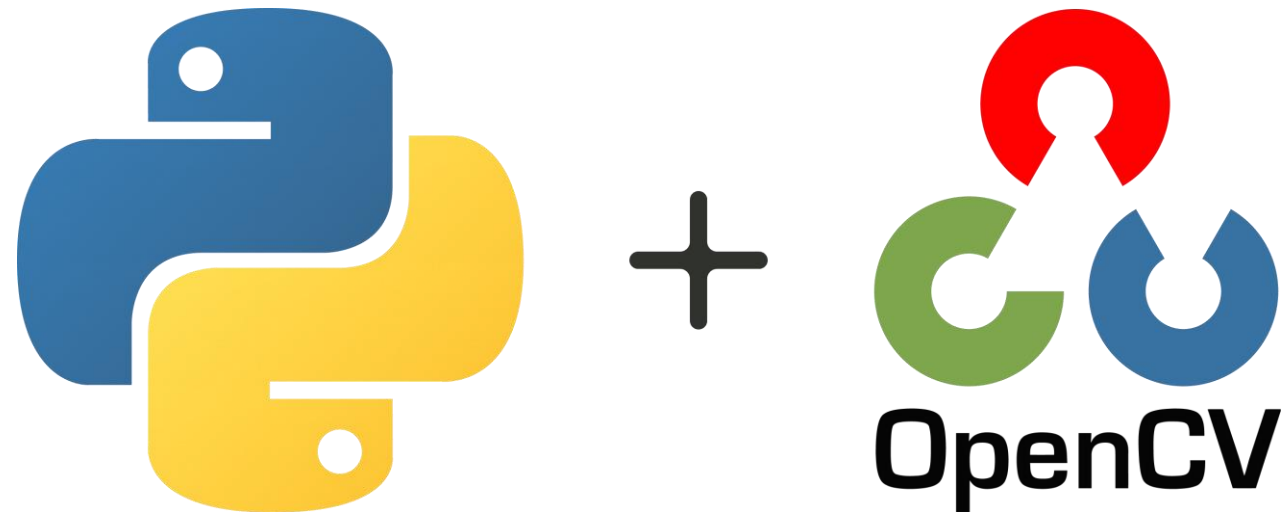
2. Revisão bibliográfica

Processamento Digital de Imagens



2. Revisão bibliográfica

Python e OpenCV



2. Revisão bibliográfica

Algoritmos de convolução

- Detectores de borda
 - Detector Canny
 - Detector Laplaciano
 - Detector Sobel
- Transformada probabilística de Hough



3. Procedimentos Experimentais

Captura da imagens



Fonte: GOOGLE (2020).

3. Procedimentos Experimentais

Meio utilizado para captura das imagens



Fonte: DJI (2017).

3. Procedimentos Experimentais

Características do voo realizado para obtenção das imagens

- **Altitude:** 10 metros
- **Distância da linha:** 15 metros
- **Angulação da câmera:** 0°
- **Resolução utilizada:** 4000 x 3000 pixels
- **Balanco de branco:** *Sunny*



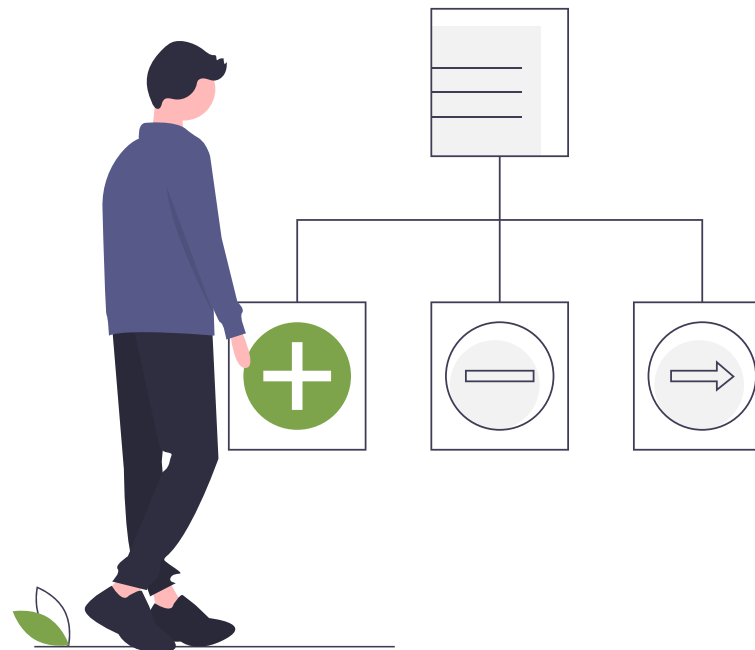
3. Procedimentos Experimentais

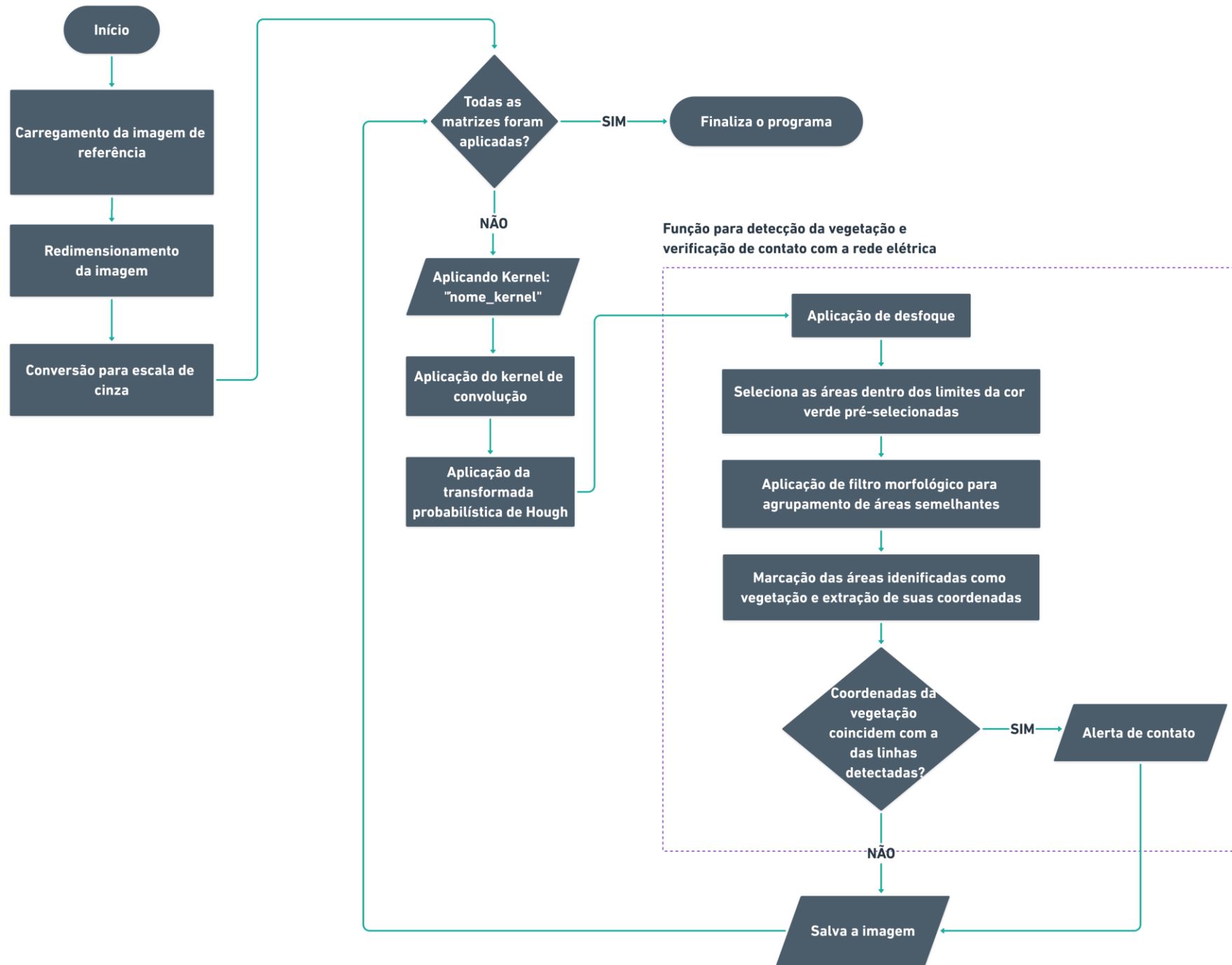
Escolha da imagem usada como “referência”



3. Procedimentos Experimentais

Fluxograma geral para desenvolvimento do algoritmo





3. Procedimentos Experimentais

Implementação das matrizes convolucionais

```
# Argumentos necessários para inicialização
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--imagem", required=True,
                help="Caminho até a imagem de referencia")
args = vars(ap.parse_args())

# Carregamento da imagem
image = cv2.imread(args["imagem"])
# Conversão para escala de cinza
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

for (kernelName, kernel) in kernelBank:
    print("Aplicando {} kernel".format(kernelName))
    opencvOutput = cv2.filter2D(gray, -1, kernel)
```

```
# Kernel Laplaciano
laplacian = np.array([
    [1, 1, 1],
    [1, 8, 1],
    [1, 1, 1]], dtype="int")

# Kernel Sobel Y
sobelY = np.array([
    [-1, -2, -1],
    [0, 0, 0],
    [1, 2, 1]], dtype="int")

# Kernel Canny
canny = detector_canny()

kernelBank = (
    ("sobel_y", sobelY),
    ("canny", canny),
    ("laplacian", laplacian)
)
```


3. Procedimentos Experimentais

Implementação da Transformada Probabilística de Hough

```
# Parâmetros para detecção das linhas
minLineLength = 150
maxLineGap = 30
threshold = 100

# Aplicação da TPH
lines = cv2.HoughLinesP(kernel, 1, np.pi/180, threshold, minLineLength, maxLineGap)
for line in lines:
    for x1,y1,x2,y2 in line:
        cv2.line(gray, (x1 + 1000, y1), (x2 - 5000, y2), (0,0,255), 1)
```

3. Procedimentos Experimentais

Detecção da vegetação e contato com a rede elétrica

```
def parametros_padrao(self):  
    default_args = {  
        'blur': 15,  
        'morph': 6,  
        'iterations': 4,  
  
        'HSV_min': [self.params.parameters['H'][48],  
                   self.params.parameters['S'][10],  
                   self.params.parameters['V'][25]],  
  
        'HSV_max': [self.params.parameters['H'][33],  
                   self.params.parameters['S'][166],  
                   self.params.parameters['V'][111]],  
    }
```

3. Procedimentos Experimentais

Detecção da vegetação e contato com a rede elétrica

```
def _morph(self):
    processes = self.params.array
    for process in processes:
        morph_amount = process['size']
        kernel_type = self.params.cv2_kt[process['kernel']]
        morph_type = self.params.cv2_mt[process['type']]
        iterations = process['iters']
        kernel = cv2.getStructuringElement( kernel_type, (morph_amount, morph_amount))
        if morph_type == 'erode':
            self.images['morphed'] = cv2.erode(
                self.images['morphed'], kernel, iterations=iterations)

        self.images['current'] = self.images['morphed']

self.images['current'] = cv2.drawContours( self.images['morphed'], 0, (0, 255, 0), 3)
coord = self.find(self.images['current'])
```

3. Procedimentos Experimentais

Detecção da vegetação e contato com a rede elétrica

```
cordXLinhas = []
cordYLinhas = []
cordXVegetacao = []
cordYVegetacao = []

# Separação das coordenadas X da rede e da vegetação
for line in lines:
    cordXLinhas.append(line[0])
for line in coord:
    cordXVegetacao.append(line[0])

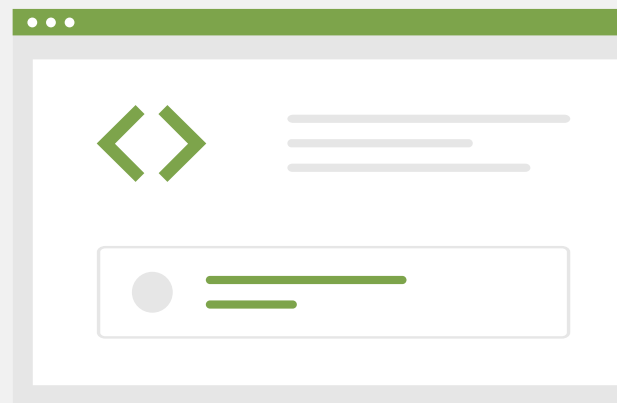
# Separação das coordenadas Y da rede e da vegetação
for col in lines:
    cordYLinhas.append(col[1])
for col in coord:
    cordYVegetacao.append(col[1])

# Verifica se a cruzamento entre as coordenadas
for line in cordXLinhas:
    if line in cordXVegetacao:
        print("Possível Contato Detectado")

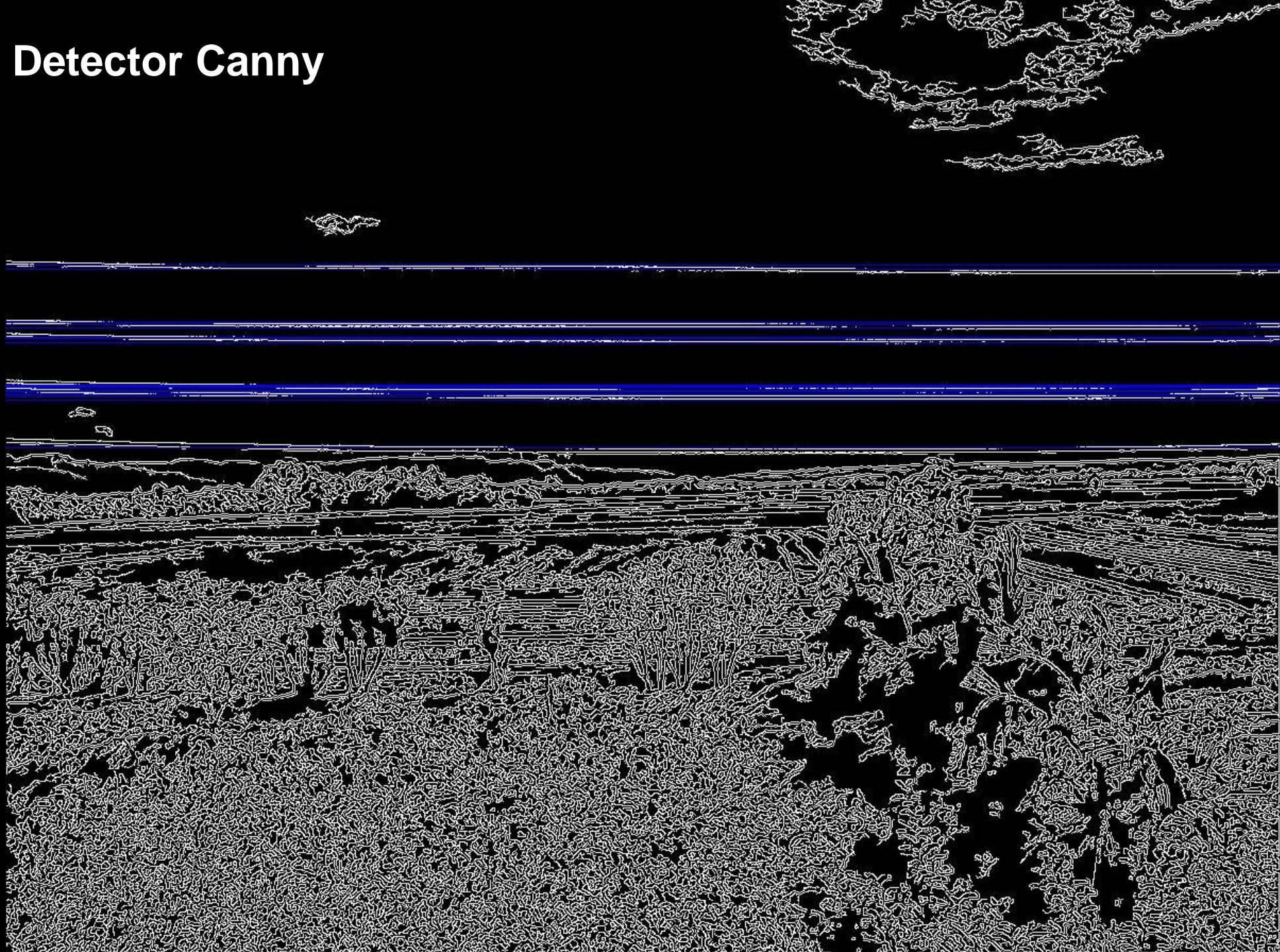
for col in cordYLinhas:
    if col in cordYVegetacao:
        print("Possível Contato Detectado")
```

4. Resultados e Discussões

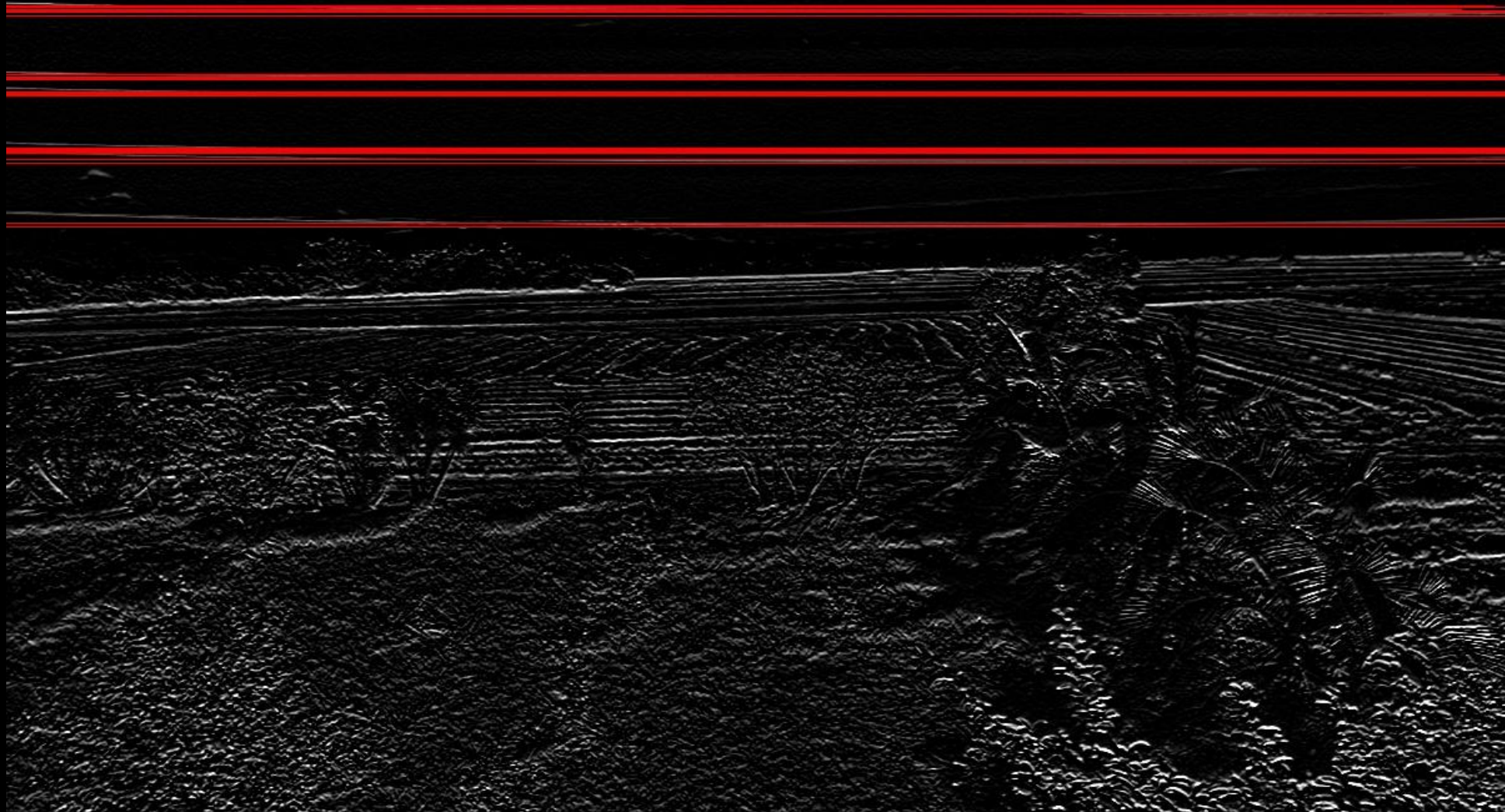
Reconhecimento da rede elétrica



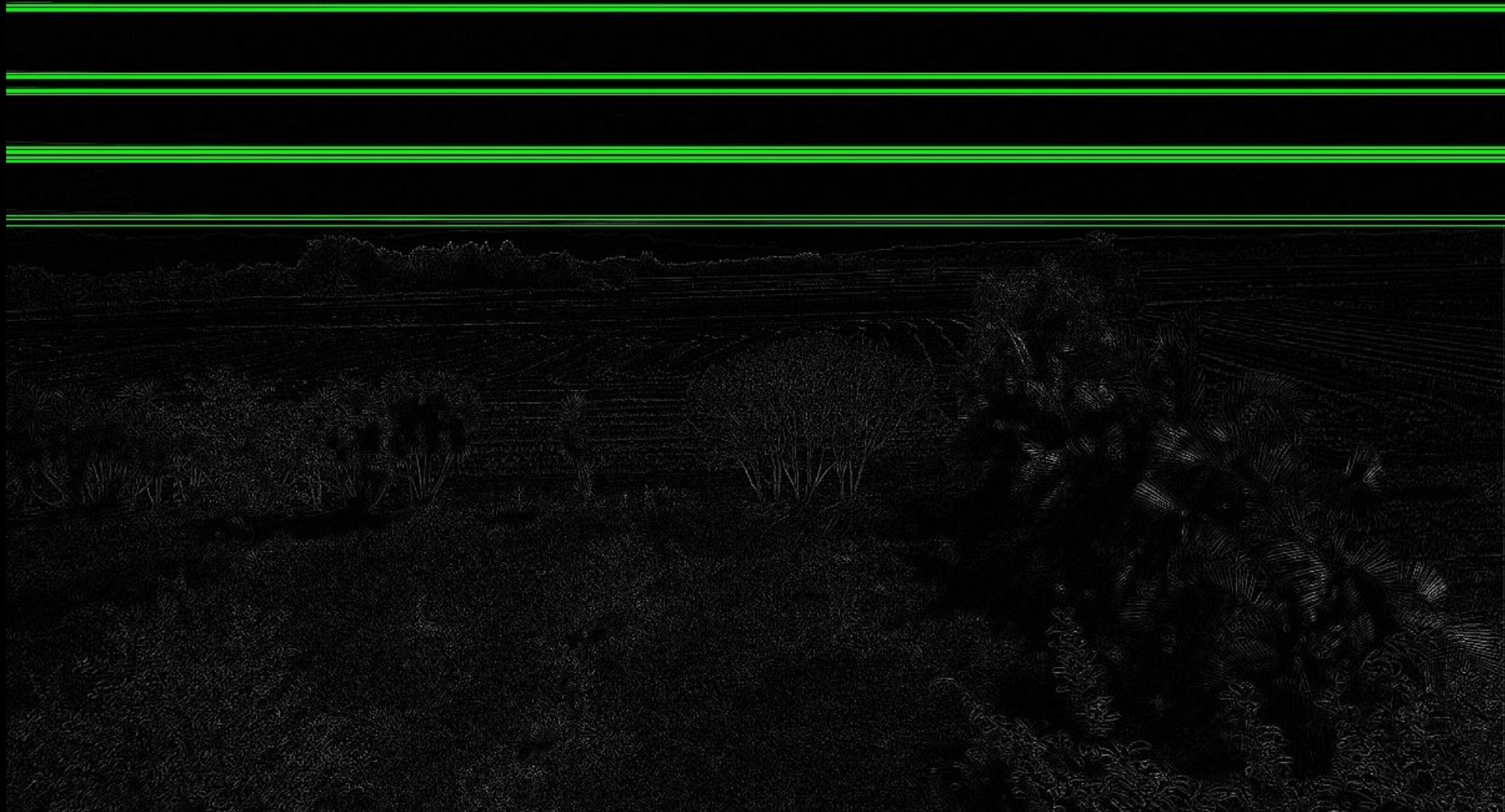
Detector Canny



Detector Sobel



Detector Laplaciano



4. Resultados e Discussões

Reconhecimento da vegetação



4. Resultados e Discussões

Contato da rede elétrica com a vegetação



4. Resultados e Discussões

Contato da rede elétrica com a vegetação



5. Conclusão

- Os três detectores de bordas analisados em conjunto com TPH mostraram-se eficazes para detecção da rede elétrica
- A identificação da vegetação através do método de classificação por nível de cor apresentou-se satisfatória
- A relação entre as coordenadas dos cabos contutores com as coordenadas da vegetação podem resultar em falsos positivos

5. Conclusão

Sugestões para trabalhos futuros

- Desenvolvimento de interface gráfica para o algoritmo proposto
- Utilização de bibliotecas de *machine learning* e/ou IA, para realização do monitoramento através de vídeos

6. Referências

DJI. **PHANTOM 3 Standard: User Manual**. 2017.

GOOGLE. **Google**. Disponível em: <<https://www.google.com.br/maps/@-28.7411274,-49.4786331,14z>>.

GOVERNO DO ESTADO DE SANTA CATARINA. **Vegetação na rede é a principal causa de falta de energia no Estado**. Disponível em: <<https://www.sc.gov.br/index.php/noticias/temas/energia/vegetacao-na-rede-e-a-principal-cao-de-falta-de-energia-no-estado>>. Acesso em: 13 jul. 2020.

HEDMOND, S. **Arcing Power Line Shows You Exactly Why To Stay a Safe Distance Away**. Disponível em: <<https://www.constructionjunkie.com/blog/2017/11/17/video-of-arc-ing-power-line-shows-you-exactly-why-to-stay-a-safe-distance-away>>. Acesso em: 13 jul. 2020.

PROCESSAMENTO DIGITAL DE IMAGENS PARA IDENTIFICAÇÃO DE VEGETAÇÃO SOB LINHAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

Acadêmico: Marco Aurélio Oliveira Rocha

Orientador: Dr. Anderson Diogo Spacek