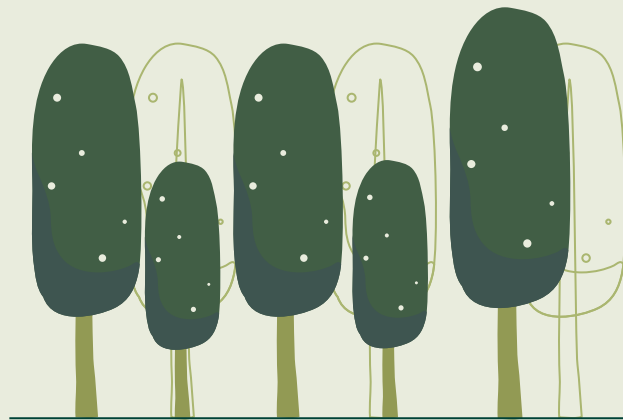


Desafio P&D

Resultados

Mateus Grota Nishimura Ferro



Seções

01 Etapa 1

Pré-processamento e
segmentação

02 Etapa 2

Vetorização de polígonos

03 Etapa 3

Contagem de árvores

04 Etapa 4

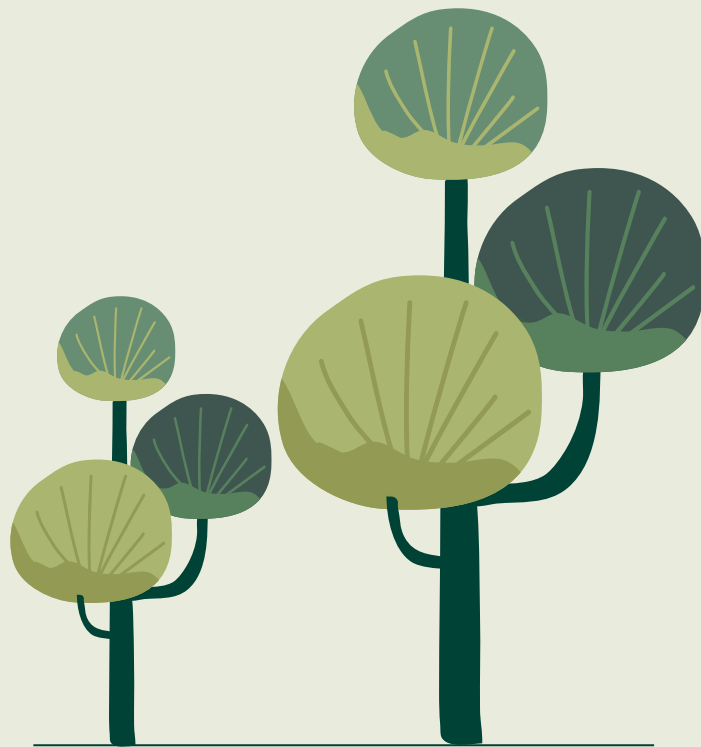
Heterogeneidade das árvores



01

Etapa 1

Segmentação da imagem



Etapa 1



Etapa mais complicada do qual **todas** as outras **etapas dependiam**. Para esta etapa usou-se: ***técnicas de pré-processamento; segmentação; detecção de bordas; aplicação de máscara.***

Citando todas as técnicas de pré-processamento testadas para garantir qual teria melhor resultado:

Aplicação de filtros:

- filtro da média
- filtro da mediana
- filtro gaussiano
- filtro de Sobel

Operações morfológicas:

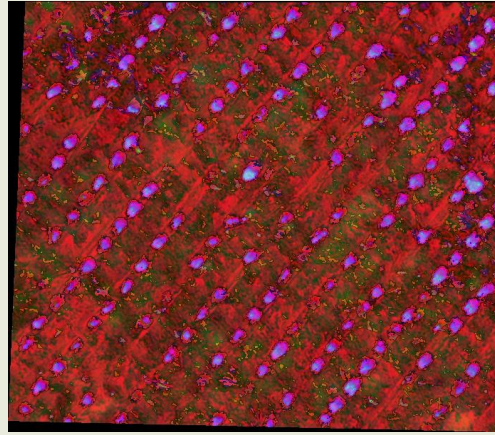
- dilatação,
- erosão
- abertura
- fechamento

Pré-processamento

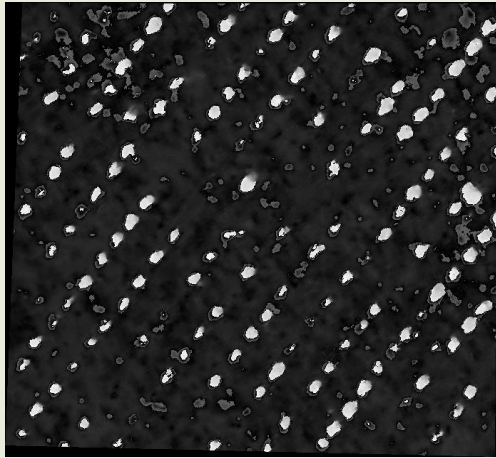
As imagens foram também transformadas para o espaço de cor HSV, para tornar a detecção **menos sensível** às **variações de iluminação**, que podem prejudicar a precisão da segmentação.

Ao analisar o canal de **matiz** (Hue), que representa a cor dominante em uma imagem, já podemos **discernir** claramente as características das **árvores**, mesmo em condições de iluminação variáveis

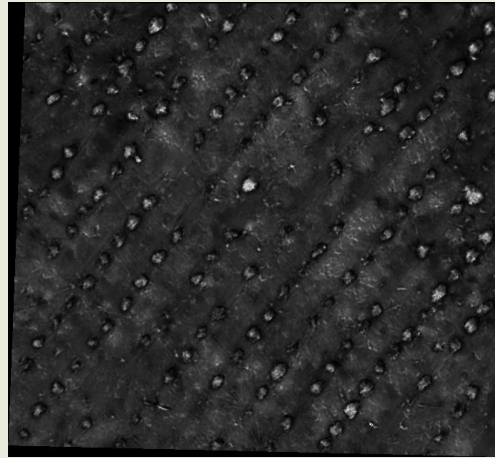




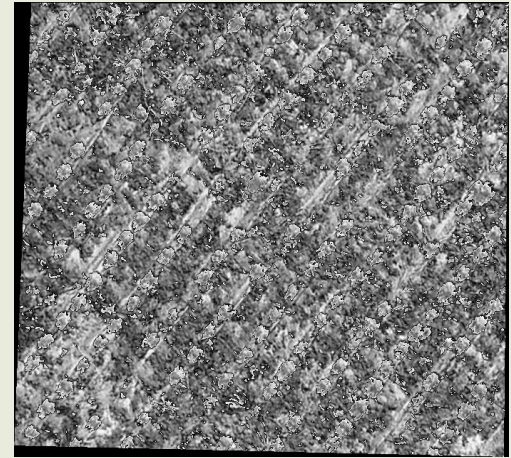
=



+



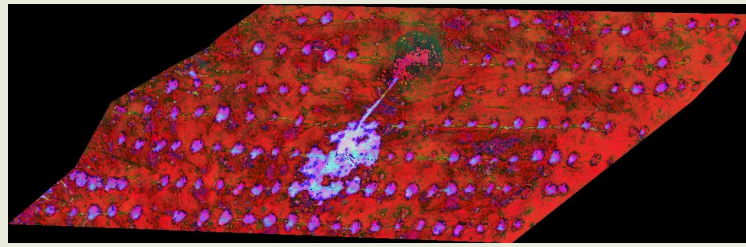
+



Matiz

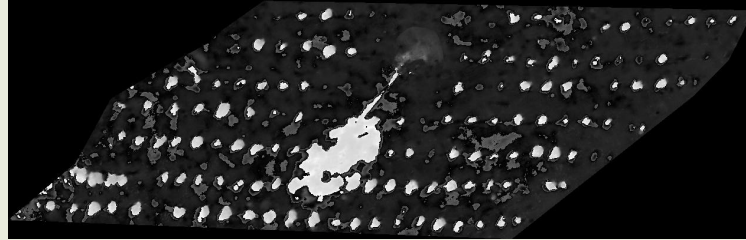
Saturação

Valor



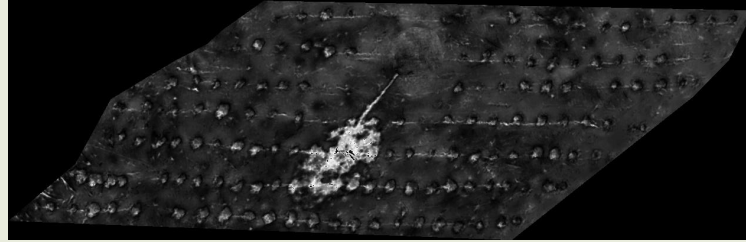
=

Matiz



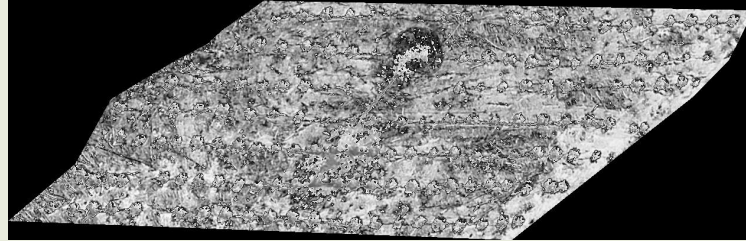
+

Saturação



+

Valor



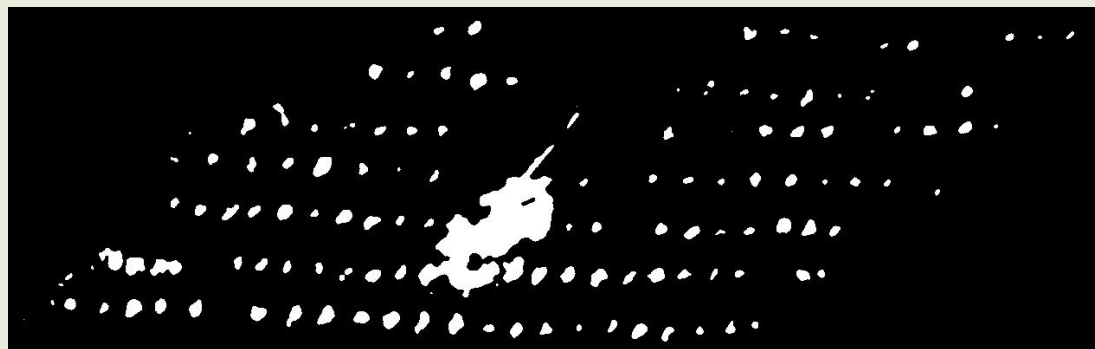
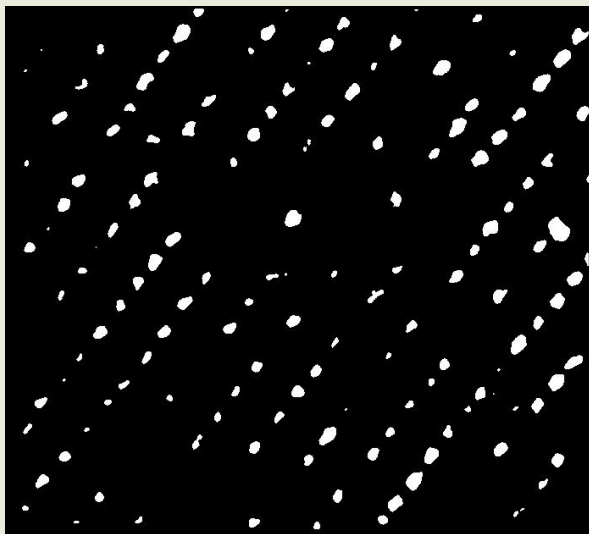
Segmentação

A segmentação foi feita com **limiarização binária**, com o seguinte trecho de código:

```
_, segmentada = cv2.threshold(imagem, 150, 255, cv2.THRESH_BINARY)
```

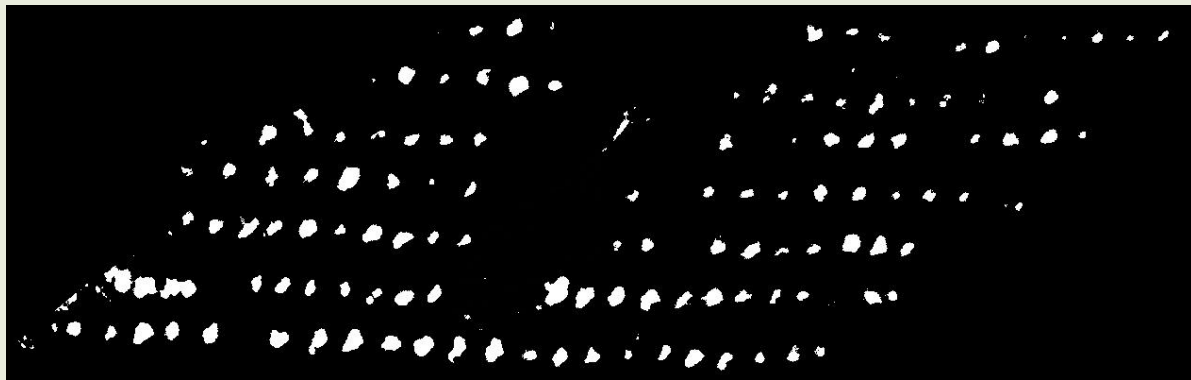
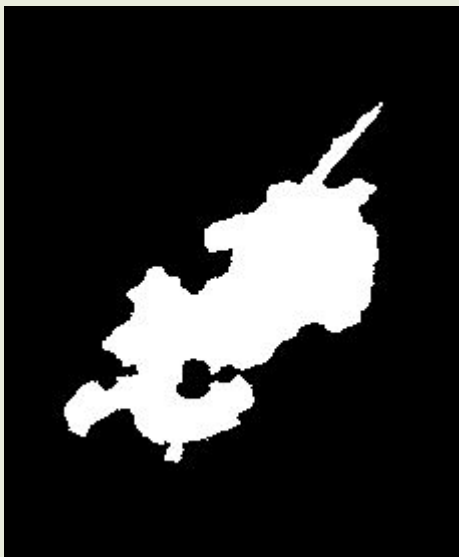
Onde os píxeis com **intensidade maior** que **150** são definidos como 255 (**branco**) e os píxeis com **intensidade menor** ou **igual** a **150** são definidos como 0 (**preto**), resultando em uma imagem binária segmentada.

É possível notar que a **sombra** causada na imagem *sample2.tif*, **atrapalhou** a segmentação, tornando toda a sombra como uma grande árvore, isto **impactaria** no **cálculo** da **heterogeneidade**.



Remoção sombra árvore

Para a remoção da sombra da árvore, foi definido um **limiar** para a **área de cada contorno** de árvore de **1000**. Assim, foi possível **segmentar a sombra** e aplicar uma **máscara inversa**, **eliminando-a** da imagem.



02

Etapa 2

Vetorização de polígonos



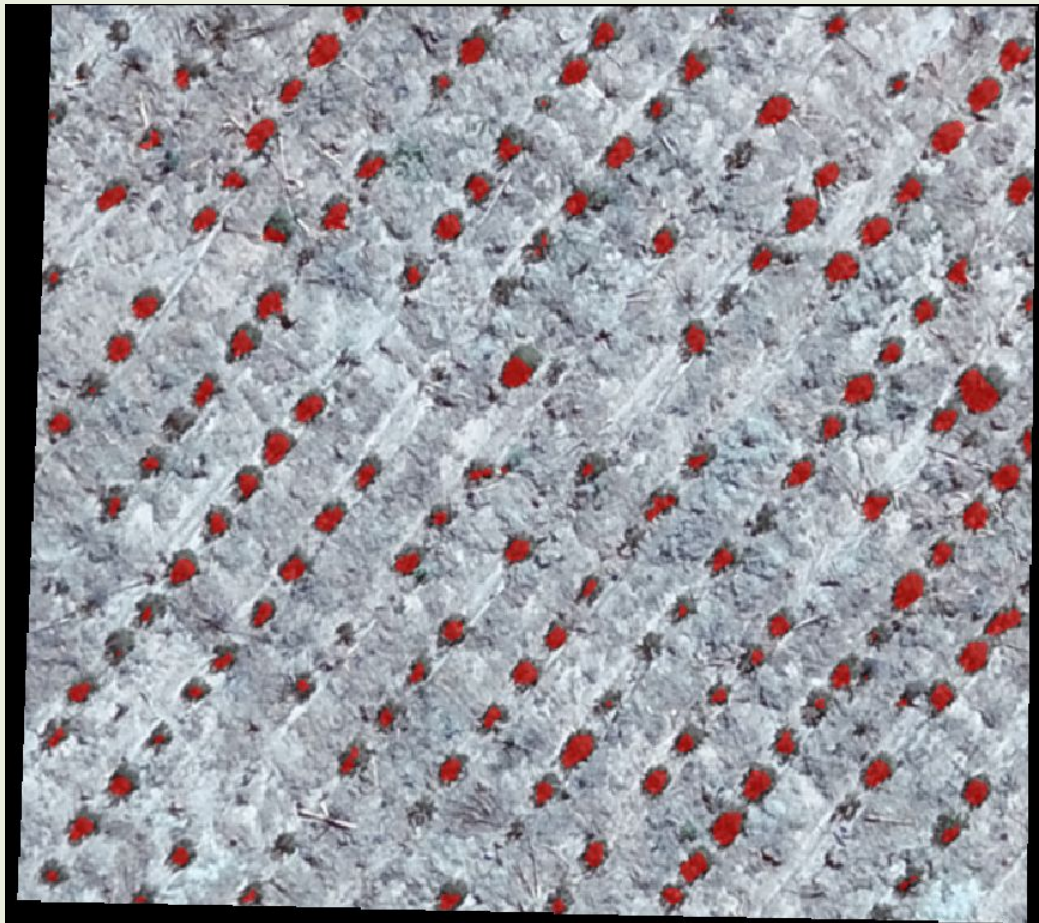
Detecção de contornos



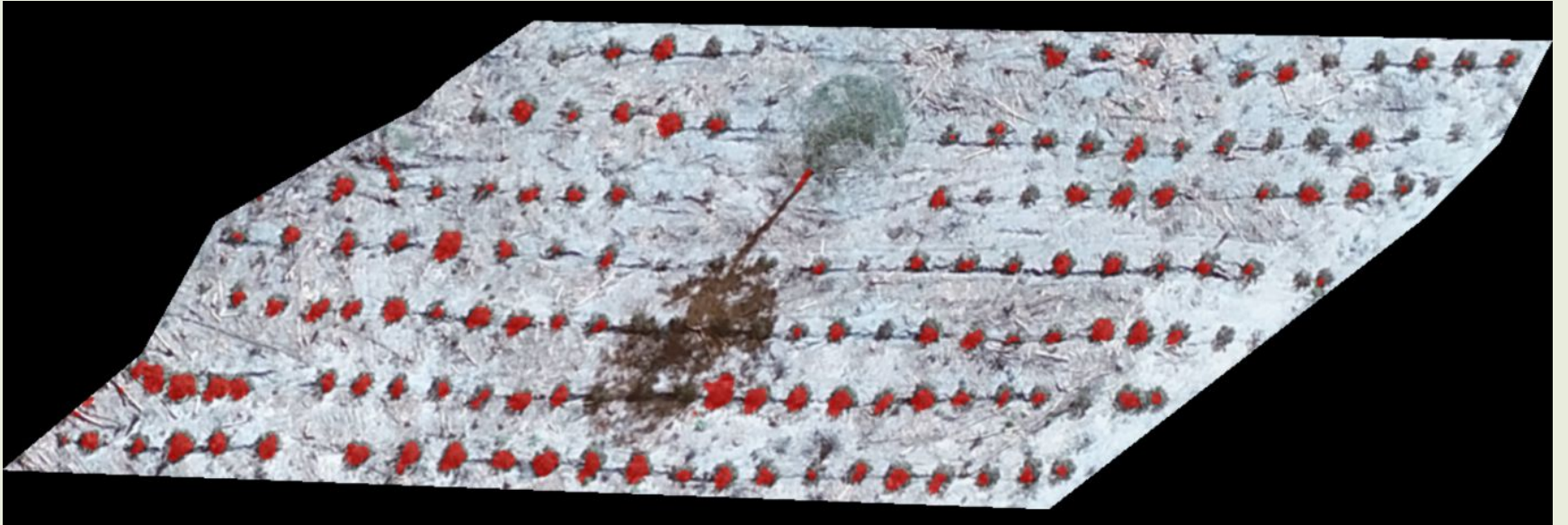
Para esta etapa, a **detecção de contornos** foi essencial, sendo possível através dele, pegar as coordenadas dos pontos de cada polígono segmentado.

Através da função `cv2.findContours()`, foram identificados os **contornos dos polígonos** e a partir disso, eles foram transformados em **objetos Polygon**.

Esses polígonos são então armazenados em um **formato de dados geoespaciais** usando um **GeoDataFrame**, **exportado** como um arquivo **GeoJSON**. Em seguida, a função **generate_image** usa esses contornos para sobrepor visualmente a imagem original, destacando as regiões onde os contornos foram encontrados.



sample1.tif



sample2.tif

03

Etapa 3



Contagem de árvores



A partir do GeoJSON gerado, a quantidade de polígonos foi contada, tendo como saída o número de árvores.

```
def count_trees():  
    gdf = gpd.read_file("./shapefile.geojson")  
    print(f"Árvores encontradas: {len(gdf)}")  
    return len(gdf)
```

sample1.tif - Árvores encontradas: 142

sample2.tif - Árvores encontradas: 133

04

Etapa 4

Heterogeneidade das árvores



Heterogeneidade



Para descobrir a heterogeneidade das árvores, a área de todos os polígonos foram calculadas e normalizadas, para o desvio padrão ser calculado logo em seguida.

sample1.tif - Heterogeneidade: 0.15196476056804528

sample2.tif - Heterogeneidade: 0.18409182718573724



Conclusão

Conclusão



Através da contagem manual, foi possível tirar as métricas de precisão e recall, destaco que não foi possível ter certeza se alguns pontos eram árvores, ou apenas solo.

sample1.tif - 143 árvores

sample2.tif - 135 árvores

Contando as árvores detectadas(TP), as árvores não detectadas(FN) e os contornos detectados que não eram árvores(FP), obtivemos as seguintes métricas:

sample1.tif

TP: 138

FN: 1

FP: 3

Precisão: 0,97

Revocação: 0,99

sample2.tif

TP: 122

FN: 7

FP: 4

Precisão: 0,94

Revocação: 0,99

Conclusão



Embora as métricas de precisão e recall indiquem um desempenho satisfatório do modelo de segmentação atual, a análise visual revela que as segmentações das copas das árvores não foram precisas o suficiente.

Diante das limitações de tempo e recursos de hardware, não foi possível realizar o treinamento de uma rede neural, como a U-Net, que poderia oferecer maior robustez e capacidade de segmentações mais precisas e confiáveis das copas das árvores.