

# Detecção e Análise de Tamanho de Copas de Árvores Urbanas com YOLOv11 e K-Means: Uma Ferramenta para Planejamento Urbano

Breno Braga Galvão

Universidade de Brasília, Brasília – DF

**PPGI**

Programa de Pós-graduação  
em Informática



# Introdução:

- **Importância da vegetação urbana**
  - Melhora da qualidade do ar
  - Regulação térmica e equilíbrio ecológico
- **Desafios no monitoramento e gestão das áreas verdes**
  - Necessidade de métodos precisos e escaláveis
- **Objetivo do estudo**
  - Desenvolver um pipeline automatizado para análise das copas de árvores urbanas
- **Metodologia**
  - Combinação do YOLOv11 (detecção de objetos) com K-Means (classificação das copas)
  - Estimativa da área das copas e análise das características estruturais



# Contextualização:

Modelos como YOLO e DETR são amplamente usados na detecção de objetos, mas enfrentam limitações ao equilibrar precisão e velocidade em tempo real. O YOLOv11, uma versão otimizada do YOLO, destaca-se por sua:

- alta precisão e
- eficiência computacional.

Nesse trabalho iremos explorar o desempenho da **YOLOv11** na detecção de copas de árvores urbanas

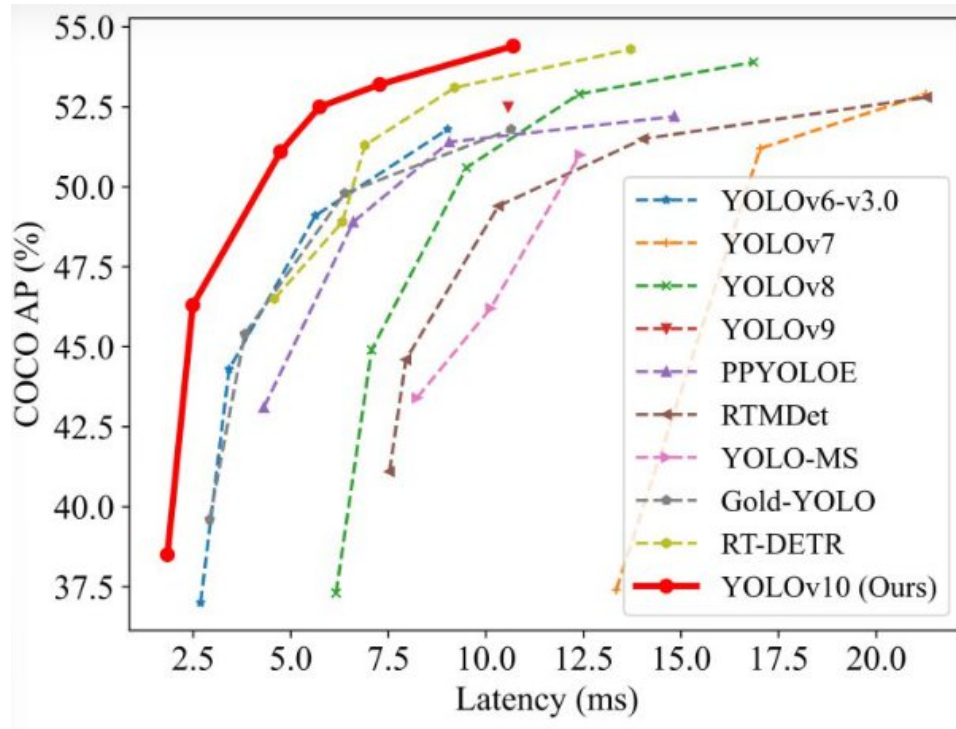


Gráfico 1: Comparação entre os detectores de objetos em tempo real.

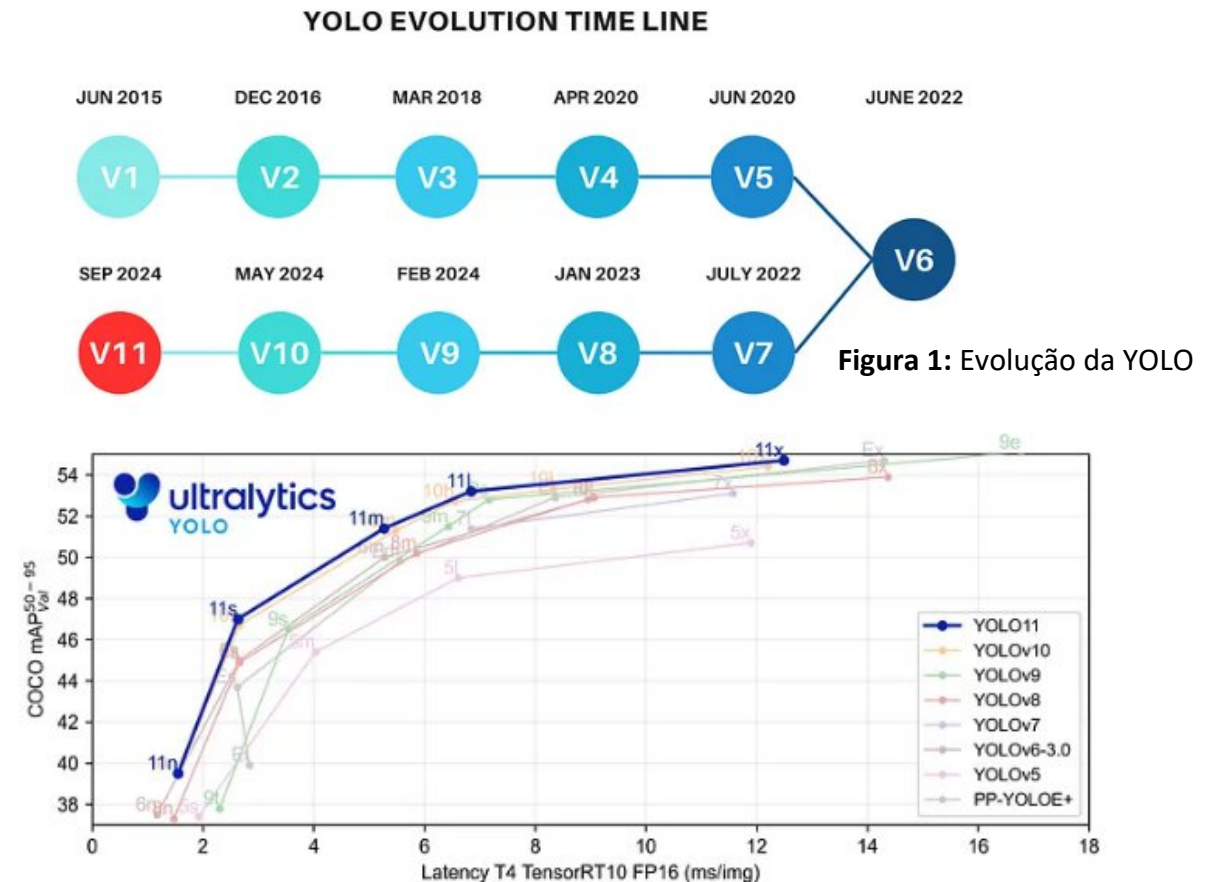


Gráfico 2: Comparação entre os detectores de objetos em tempo real.

# Contextualização:

## ETAPAS:

**1ª: Conversão para DataFrame:** Os arquivos de texto foram convertidos em um DataFrame estruturado, contendo as colunas: *image\_filename* (nome da imagem), *xmin*, *xmax*, *ymin*, *ymax* (coordenadas das caixas delimitadoras). O DataFrame final continha 3.382 caixas delimitadoras associadas a 220 imagens.

	image_filename	xmin	xmax	ymin	ymax
0	106.png	46	176.0	0.0	39.0
1	106.png	196	366.0	42.0	182.0
2	106.png	420	512.0	7.0	137.0
3	106.png	507	512.0	0.0	59.0
4	106.png	509	512.0	182.0	215.0

**2ª: Divisão dos Dados:** O conjunto de dados foi dividido em conjuntos de treinamento e validação, com 90% das imagens (198 imagens e 3.080 caixas delimitadoras) destinadas ao treinamento e 10% (22 imagens e 302 caixas delimitadoras) reservadas para validação.

**3ª: Formatação dos Rótulos:** Para compatibilidade com o modelo YOLO, as coordenadas das caixas delimitadoras foram convertidas para o formato YOLO (*x\_center*, *y\_center*, *width*, *height*), normalizadas em relação ao tamanho da imagem.

As coordenadas no formato YOLO são calculadas da seguinte forma:

class	x-center	y-center	x_extend	y_extend	image_filename
0	277.0	111.0	172.0	143.0	106
0	408.0	193.0	67.0	85.0	106
0	227.0	250.0	78.0	59.0	106
0	338.0	213.0	85.0	91.0	106
0	465.0	78.0	92.0	152.0	106

$$x_{center} = \frac{x_{min} + x_{max}}{2 \times largura_{imagem}}$$

$$y_{center} = \frac{y_{min} + y_{max}}{2 \times altura_{imagem}}$$

$$width = \frac{x_{max} - x_{min}}{largura_{imagem}}$$

$$height = \frac{y_{max} - y_{min}}{altura_{imagem}}$$

Uma coluna adicional *class* foi adicionada ao DataFrame, contendo o valor 0 para todas as caixas delimitadoras, indicando a classe única "árvore"

# Contextualização:

## ETAPAS:

4ª: **Escolha do modelo para detecção das árvores:** Utilizou-se a **YOLOv11** para a detecção das copas de árvores urbanas

5ª: **Aplicação do K-Means:** Para analisar a distribuição dos tamanhos das copas das árvores detectadas, foi aplicado o algoritmo K-Means na coluna *area* (calculada como o produto de *x\_extend* e *y\_extend* das caixas delimitadoras).

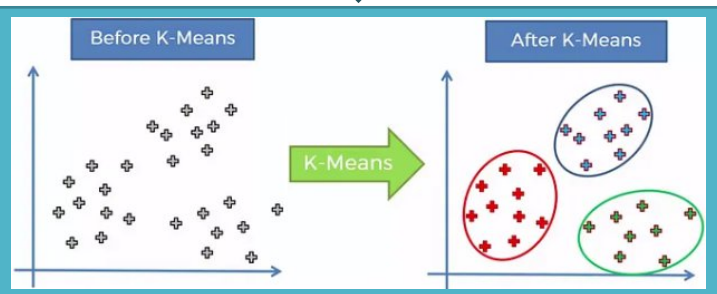
class	x-center	y-center	x_extend	y_extend	image_filename	area
0	277.0	111.0	172.0	143.0	106	24596.0
0	408.0	193.0	67.0	85.0	106	5695.0
0	227.0	250.0	78.0	59.0	106	4602.0
0	338.0	213.0	85.0	91.0	106	7735.0
0	465.0	78.0	92.0	152.0	106	13984.0

**x-center, y-center:** Coordenadas do centro da caixa delimitadora (bounding box) em relação à imagem.

**x\_extend, y\_extend:** Esses valores representam a largura (*x\_extend*) e a altura (*y\_extend*) da caixa delimitadora.

**area:** A área é calculada multiplicando *x\_extend* por *y\_extend*, resultando em **pixels quadrados** ( $\text{px}^2$ ). Por exemplo, na primeira linha, a área é  $172.0 * 143.0 = 24596.0 \text{ px}^2$

1 pixel = 5 cm = 0,05 m



A aplicação do algoritmo K-Means permitiu a categorização das árvores detectadas em três clusters de tamanho: pequeno, médio e grande

	class	x-center	y-center	x_extend	y_extend	image_filename	area	cluster
0	0	277.0	111.0	172.0	143.0	106.jpg	24596.0	2
1	0	408.0	193.0	67.0	85.0	106.jpg	5695.0	1
2	0	227.0	250.0	78.0	59.0	106.jpg	4602.0	0
3	0	338.0	213.0	85.0	91.0	106.jpg	7735.0	1
4	0	465.0	78.0	92.0	152.0	106.jpg	13984.0	1



# Resultados:

## Resultados:

**1º: Resultados da aplicação do YOLOv11:** O modelo apresentou 221 verdadeiros positivos (TP), 70 falsos negativos (FN) e 81 falsos positivos (FP).

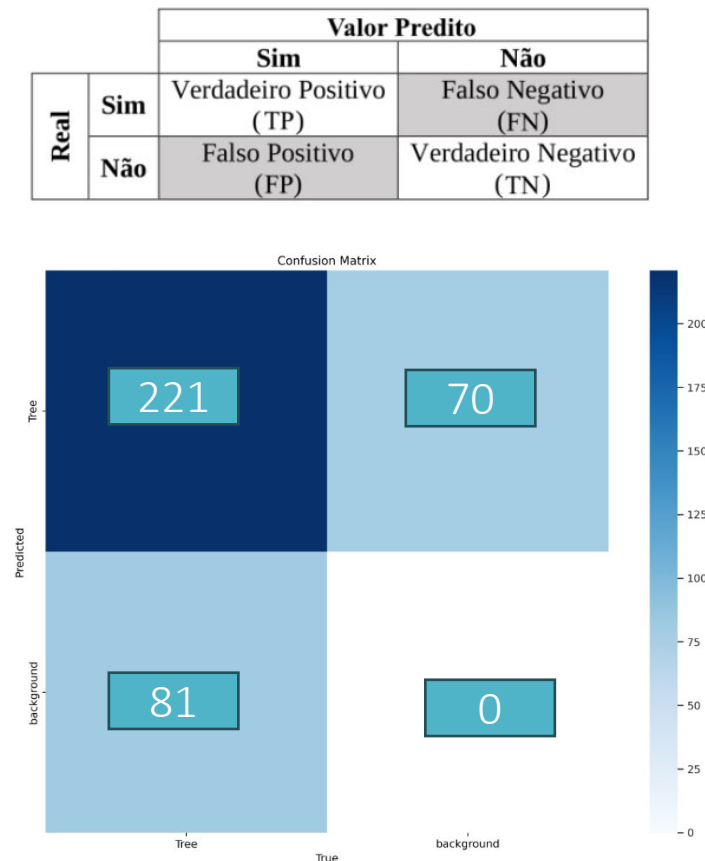


Figura 2: Matriz de Confusão

**Precisão:**

Dentre os previstos como positivos, qual a proporção de acertos.

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Precisão} = \frac{221}{221 + 81} = \frac{221}{302} \approx 0,7318 \quad (\text{ou } 73,18\%)$$

**Recall:**

Dentre os positivos reais, qual a proporção de acertos.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Revocação} = \frac{221}{221 + 70} = \frac{221}{291} \approx 0,7595 \quad (\text{ou } 75,95\%)$$

**F1-score:**

Combina precisão e recall.

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$$

$$F1\text{-Score} = 2 \times \frac{0,7318 \times 0,7595}{0,7318 + 0,7595} = 2 \times \frac{0,556}{1,4913} \approx 0,7456 \quad (\text{ou } 74,56\%)$$



# Resultados:

Resultados:

2º: Resultados da aplicação do K-Means: O K-Means subdividiu as árvores em três Clusters (0 = pequeno, 1 = Médio e 2 = Grande)

Cluster	Área em px <sup>2</sup>
Pequeno (0)	Área < 5000,0 px <sup>2</sup>
Médio (1)	5000,0 ≤ Área < 20000,0 px <sup>2</sup>
Grande (2)	Área ≥ 20000,0 px <sup>2</sup>

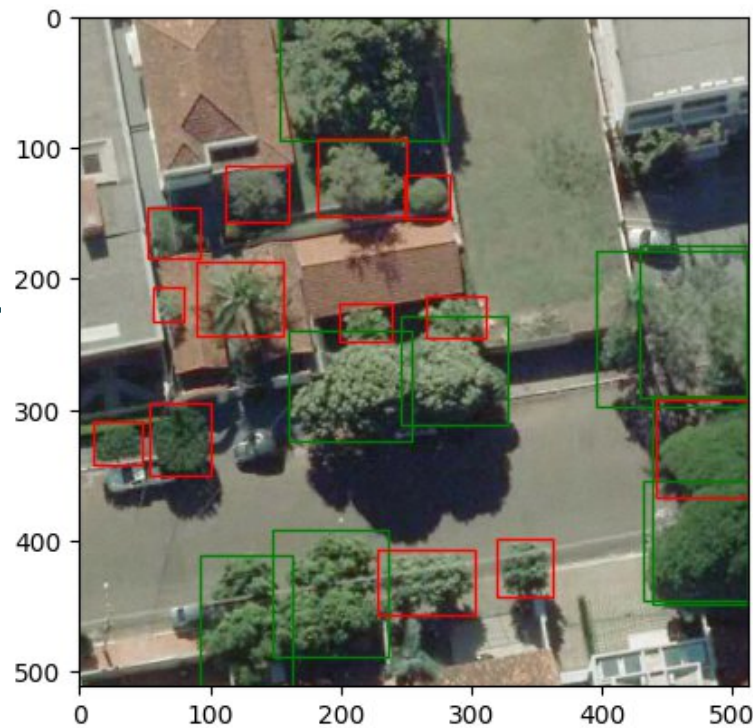


Figura 3: Exemplo dos dados finais com áreas das copas e classificação em clusters.

# Resultados:

## Resultados:

3º: **Estimando o tamanho da copa da árvore:** Para imagens de drones com baixa altitude, 1 pixel pode representar 5 cm, dessa forma, temos que:

Cluster	Área em px <sup>2</sup>	Área em m <sup>2</sup>
Pequeno (0)	Área < 5000,0 px <sup>2</sup>	Área < 5000 × 0,0025 = 12,5 m <sup>2</sup>
Médio (1)	5000,0 ≤ Área < 20000,0 px <sup>2</sup>	12,5 m <sup>2</sup> ≤ Área < 50,0 m <sup>2</sup>
Grande (2)	Área ≥ 20000,0 px <sup>2</sup>	Área ≥ 20000 × 0,0025 = 50,0 m <sup>2</sup>

1 pixel = 5 cm = 0,05 m.

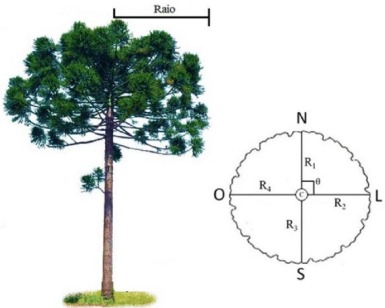
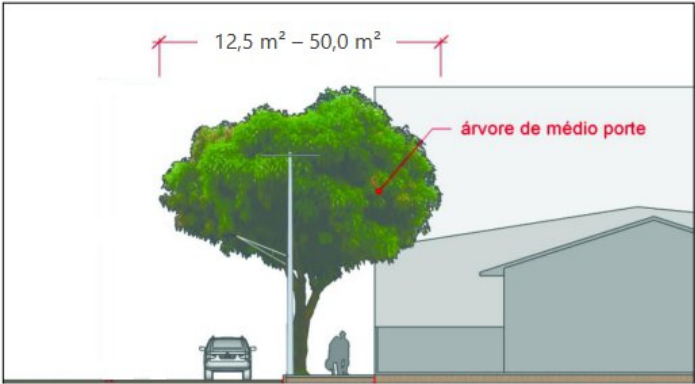
1 pixel<sup>2</sup> = 0,05 m × 0,05 m = 0,0025 m<sup>2</sup>

	class	x-center	y-center	x_extend	y_extend	image_filename	area	cluster
0	0	277.0	111.0	172.0	143.0	106.jpg	24596.0	2



Tabela Resumida para Alta Resolução (1 pixel = 5 cm):

Descrição	Pixels	Centímetros (cm)	Metros (m)	Área (cm <sup>2</sup> )	Área (m <sup>2</sup> )
Largura (x_extend)	172,0 px	860 cm	8,6 m	-	-
Altura (y_extend)	143,0 px	715 cm	7,15 m	-	-
Área da copa	24.596 px <sup>2</sup>	-	-	614.900 cm <sup>2</sup>	61,49 m <sup>2</sup>

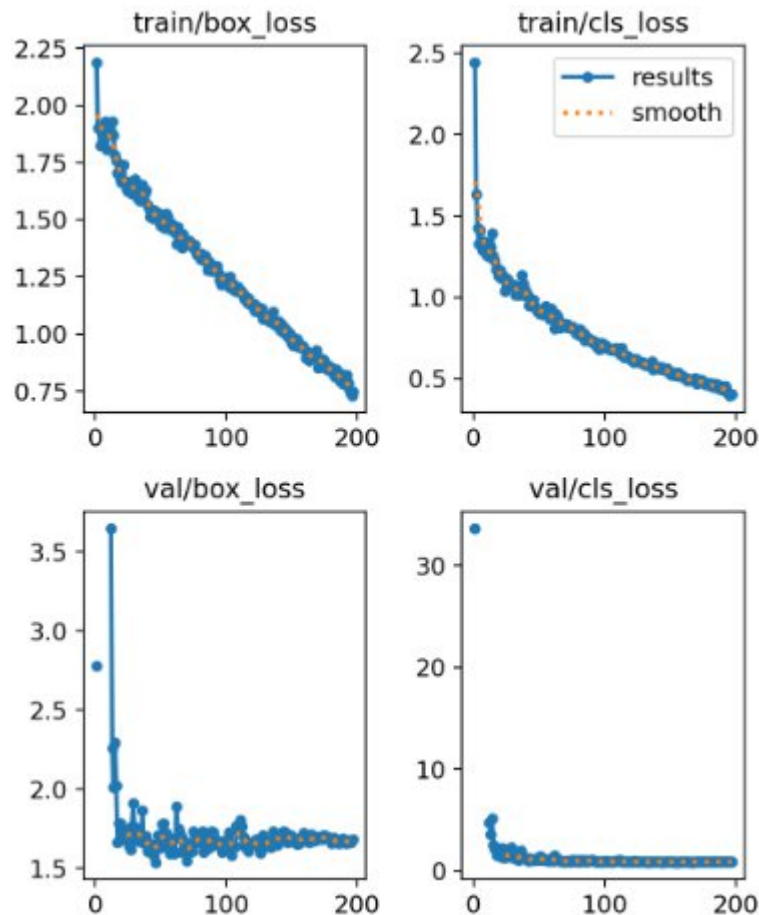




# Resultados:

## Resultados:

4º: **Desempenho do modelo:** Apresentou boa eficiência e precisão, com um mAP50 (mean Average Precision com IoU de 50%) de 0.75-0.80. Foram analisadas 200 épocas:



### Gráficos de Perda (Loss):

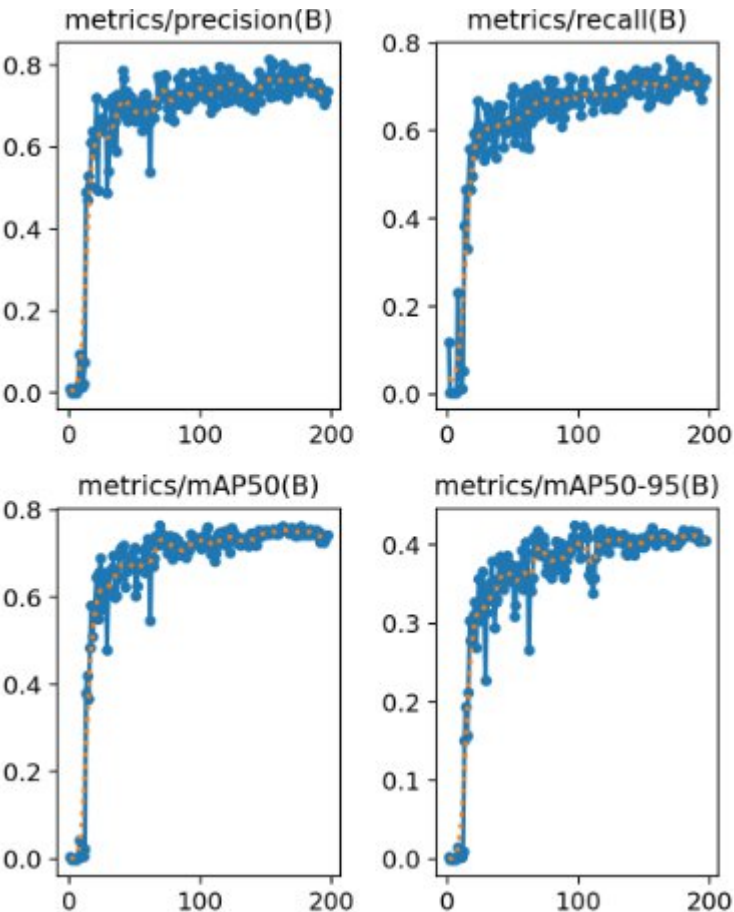
**Box Loss:** A perda associada à regressão das caixas delimitadoras (bounding boxes) diminuiu de forma consistente ao longo das épocas, indicando que o modelo aprendeu a prever com precisão as coordenadas das caixas delimitadoras.

**Class Loss:** A perda de classificação, que mede o erro na identificação correta das classes (neste caso, a classe única "árvore"). A redução confirma que o modelo é capaz de distinguir corretamente as árvores de outros elementos nas imagens.

# Resultados:

## Resultados:

4º: **Desempenho do modelo:** Apresentou boa eficiência e precisão, com um mAP50 (mean Average Precision com IoU de 50%) de 0.75-0.80. Foram analisadas 200 épocas:



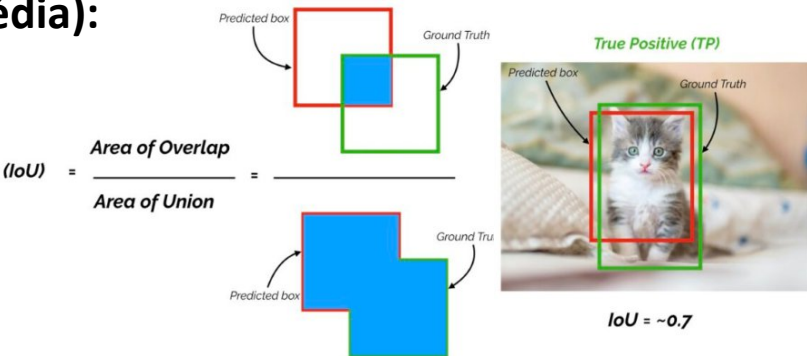
### Gráficos de Métricas:

**Precision e Recall:** Os gráficos de precisão e revocação mostraram um aumento rápido nas primeiras épocas, estabilizando-se em valores de **0,73** e **0,76**, respectivamente.

**mAP50 e mAP50-95:** As métricas de mAP50 e mAP50-95 apresentaram uma tendência de crescimento ao longo do treinamento, atingindo valores de **0,75-0,80** e **0,40-0,45**, respectivamente. Esses resultados confirmam a robustez do modelo em diferentes cenários

- **mAP50 e mAP50-95 (Precisão média):**

Métrica	Limiar de IoU	Rigor
mAP50	IoU $\geq$ 50%	Menos rigorosa
mAP50-95	IoU de 50% a 95% (em 5%)	Mais rigorosa



# Conclusões:

## Viabilidade do Modelo YOLOv11:

- O modelo YOLOv11 demonstrou ser eficaz para a detecção de copas de árvores urbanas, com **precisão de 73%, revocação de 76% e F1-Score de 0.745**.
- As métricas **mAP50 (0.75-0.80)** e **mAP50-95 (0.40-0.45)** confirmaram a robustez e capacidade de generalização do modelo.

## Análise de Clusters com K-Means:

- A categorização das copas das árvores em três grupos (**pequeno, médio e grande**) forneceu insights valiosos sobre a distribuição dos tamanhos das árvores.
- Essa análise pode ser útil para **planejamento urbano, gestão de áreas verdes e estudos ecológicos**.

## Análise do tamanho das copas:

- Foi possível estimar com precisão o tamanho das copas, através de uma análise dos pixels.

## Sugestões de trabalhos futuros:

- **Expansão do Conjunto de Dados:** Incluir imagens de diferentes ambientes e condições climáticas para melhorar a generalização do modelo.
- **Aplicações Práticas:** Integrar o modelo em sistemas de **monitoramento em tempo real** e desenvolver ferramentas para **estimativa de biomassa** ou **sequestro de carbono**

# Referências:

P. Zamboni, J. M. Junior, J. de Andrade Silva, G. T. Miyoshi, E. T. Matsubara, K. Nogueira, and W. N. Gonçalves. "Benchmarking Anchor-Based and Anchor-Free State-of-the-Art Deep Learning Methods for Individual Tree Detection in RGB High-Resolution Images." Remote Sensing, 13(13):2482, 2021. <https://doi.org/10.3390/rs13132482>.

Ultralytics. "YOLOv11: State-of-the-Art Object Detection Model." Ultralytics Documentation, 2023. Available: [YOLOv11 Documentation](#).

# Obrigado!