

DETECÇÃO E CÁLCULO DA VELOCIDADE DE DESCARGAS ELÉTRICAS ATMOSFÉRICAS USANDO DEEP LEARNING

Giovane Beck Suss - 163777
Departamento de Ciência e Tecnologia
Universidade Federal de São Paulo
São José dos Campos, São Paulo
Email: gbsuss@unifesp.br

Resumo—Este artigo apresenta o desenvolvimento e implementação de um modelo de deep learning voltado para a detecção e análise de descargas elétricas atmosféricas, especificamente raios que se propagam da nuvem em direção ao solo. O modelo YOLOv8 foi utilizado para identificar e monitorar o canal principal dessas descargas em imagens sequenciais, permitindo o cálculo da velocidade média dos raios durante o seu percurso até o solo. A metodologia proposta visa otimizar e acelerar o processo de análise dessas descargas, que anteriormente era realizado manualmente, proporcionando uma maior precisão na determinação da velocidade dos raios. A base de dados foi construída e rotulada manualmente, seguida pelo treinamento e validação do modelo YOLOv8. Os resultados demonstram a eficácia do modelo em identificar corretamente o canal principal e calcular a velocidade dos raios com precisão.

1. Introdução/motivação

Os raios são fenômenos presentes na Terra desde muito antes da evolução da vida em nosso planeta. Todas as civilizações antigas incorporaram raios e trovões em suas crenças religiosas como a ira dos deuses. Hoje sabemos que a todo o instante, existem aproximadamente 2.000 tempestades pela

atmosfera da terra, atingindo o solo cerca de 100 vezes por segundo (COORAY, 2015).

Os primeiros estudos na área de eletricidade atmosférica eram realizados através do campo elétrico, mas com o avanço da tecnologia e com o uso de câmeras de alta velocidade, foi possível adquirir imagens que permitiam o estudo da propagação destas descargas.

Com os vídeos capturados com estas câmeras, é possível observar características destes raios que não eram possíveis antigamente, como a sua velocidade média, que pode estar relacionada a outras características, e assim, explicar fenômenos que antigamente não podiam ser explicados.

A motivação deste trabalho surge da análise de raios descendentes positivos, ou nuvem-solo positivos, que são descargas elétricas que se originam na nuvem e se propagam em direção ao solo, transferindo cargas positivas ao fazerem contato com o mesmo (NAG; RAKOV, 2009). Esses raios possuem uma característica específica chamada Líder de Recuo, que são canais condutores flutuantes cujas extremidades têm polaridades opostas (MAZUR et al., 2013) e se originam em canais decaídos dos raios positivos (MAZUR et al., 2016).

Segundo dados do INPE, esta característica tem influência direta na velocidade dos raios

descendentes positivos. Como o processo de calcular a velocidade desses raios em 2D é lento, surge a necessidade de otimizá-lo. Portanto, propomos o uso de modelos de Deep Learning para a análise de vídeos e imagens, visando acelerar e aprimorar a precisão do cálculo da velocidade dessas descargas elétricas atmosféricas.

2. Conceitos fundamentais

Para que haja o entendimento do que é a proposta do trabalho, faz-se necessária a aplicação dos conceitos que envolvem o trabalho, tais como:

- Raios Descendentes Positivos: São descargas elétricas que se originam na nuvem e se propagam em direção ao solo, transferindo cargas positivas ao fazerem contato com o mesmo (NAG; RAKOV, 2009).
- Líder de Recuo: São canais condutores flutuantes cujas extremidades têm polaridades opostas (MAZUR et al., 2013) e se originam em canais decaídos dos raios positivos (MAZUR et al., 2016).
- Canal principal: É o canal condutor principal que conecta o raio no solo e assim passa a corrente para o mesmo (FERRO, 2022)..
- Câmeras de Alta Velocidade: São câmeras capazes de registrar de 1.000 a 100.000 imagens por segundo (ips) (SCHUMANN, 2016).
- Rede Neural Convolucional (CNN): é projetada para processar dados com estrutura de grade, como imagens, utilizando camadas de convolução para extrair características relevantes e camadas de pooling para reduzir a dimensionalidade.
- Camada de convolução: aplica filtros às imagens para extrair características importantes, como bordas e texturas.

- Camada de pooling: reduz a quantidade de informações, simplificando os dados e mantendo apenas as características mais relevantes, ajudando na eficiência do processamento.

- Deep Learning: É um subconjunto do aprendizado de máquina, que é essencialmente uma rede neural com três ou mais camadas.

- Modelo para Análise de Imagens: Extrai características de imagens estáticas usando CNNs, identificando padrões espaciais como formas e texturas para reconhecer e classificar elementos específicos.

3. Trabalhos Relacionados

A utilização de técnicas de Inteligência Artificial (IA) para o reconhecimento e cálculo da velocidade de raios descendentes positivos não é uma área muito estudada, pois os dados para fazer tal processo são muito escassos, sendo que apenas algumas instituições no mundo possuem vídeos de raios tirados por câmeras de alta velocidade, sendo uma delas o INPE.

Este trabalho se baseou no artigo de Campos et al. (2014) com o título “High-speed video observations of natural cloud-to-ground lightning leaders – A statistical analysis”, que fez a mesma análise deste trabalho, que foi relacionar a velocidade do raio positivo com a presença de líder de recuo.

4. OBJETIVOS

No presente trabalho, os principais objetivos foram:

- Usar o modelo de Deep Learning YOLOv8 (You Only Look Once, versão 8) da Ultralytics para a detecção de ramificações de raios positivos que se propagam em direção ao solo.
- Detectar o canal principal em sequências de raios que se propagam em direção ao solo.

- Calcular a velocidade média de raios positivos a partir de sequências de imagens nas quais o canal principal foi detectado.

5. Metodologia Experimental

5.1 Resumo do processo

Uma base de dados rotulada foi utilizada para o treinamento do modelo YOLOv8, com as imagens sendo separadas manualmente entre treino e teste. O modelo foi escolhido pela sua eficiência em tarefas de detecção de objetos, sendo treinado para identificar raios em imagens.

5.2 Pré-processamento

5.2.1 Base de dados.

O primeiro passo para a montagem do modelo foi a construção da base de dados, onde foram separadas 290 imagens para treinamento e 72 imagens para a validação do modelo, onde as imagens variam entre imagens de partes do canal principal indo em direção ao chão e imagens do canal principal fazendo contato com o solo, a fim de treinar o modelo em todas as possibilidades do percurso do raio.

Fig. 1. Representação de uma imagem que mostra parte do canal principal.

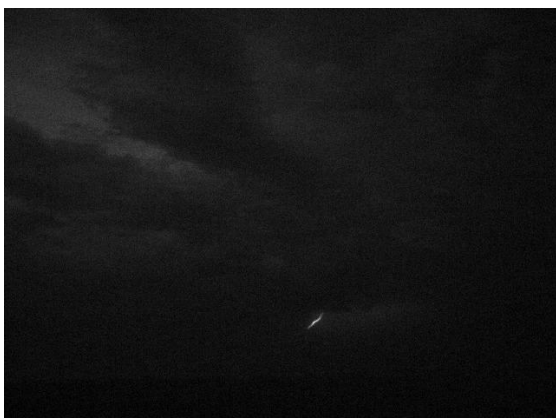


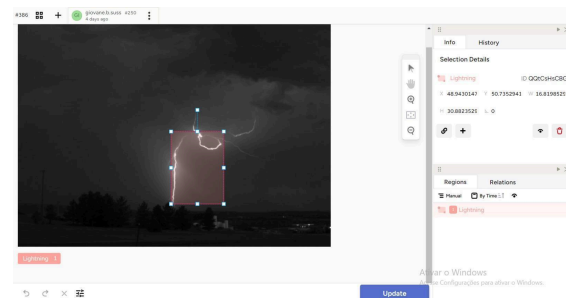
Fig. 2. Representação de outra imagem que mostra o canal fazendo contato com o solo.



Para treinar o modelo YOLOv8 com as imagens selecionadas, foi necessário rotular as imagens da base de dados para uma única classe chamada *Lightning*. A rotulação envolve fornecer ao modelo coordenadas específicas que indicam onde a caixa delimitadora deve ser posicionada, destacando a localização do raio nas imagens.

Este processo foi feito a partir da biblioteca label-studio no python, que abre um servidor no próprio computador e permite então fazer a rotulação de imagens.

Fig. 3. Exemplo de uso do Label Studio.



Após a rotulação das imagens, a base de dados foi separada em duas pastas, uma para os rótulos (*Labels*), onde foi separado entre treino e teste, e outra para as imagens, que também foram separadas entre treino e teste.

5.3 Treinamento do modelo

5.3.1 Modelo utilizado

O modelo utilizado foi o YOLOv8 (You Only Look Once) da ultralytics, onde duas versões dele foram utilizadas, sendo o YOLOv8s e o YOLOv8m. O YOLO é uma técnica de detecção de objetos em tempo real que processa a imagem inteira de uma vez, dividindo-a em uma grade e fazendo previsões simultâneas para múltiplas caixas delimitadoras e classes.

O YOLOv8s é a versão leve do YOLOv8, projetada para ser eficiente e rápida, com menos parâmetros e complexidade. Já o YOLOv8m é uma versão intermediária do YOLOv8, que equilibra precisão e velocidade. Ele possui mais parâmetros que o YOLOv8s, oferecendo maior precisão na detecção de objetos.

5.3.2 Escolha de parâmetros

Como o modelo YOLOv8 já redimensionaliza as imagens, foi escolhida uma resolução de 640 pixels por 640 pixels, que é importante para manter a consistência nos dados e evitar que o modelo encontre dificuldades ao processar imagens de diferentes tamanhos.

Também foi utilizado *batch sizes* de tamanho 32, 16 e 8, pois não temos uma base de dados muito grande.

Todos os modelos foram treinados por um total de 25 épocas para evitar o overfitting do modelo.

5.3.3 Treinamento

Seis modelos foram treinados para a análise, sendo três deles baseados no YOLOv8s: um foi treinado por 8 épocas, outro por 16 épocas e o terceiro por 32 épocas. Os outros três modelos utilizaram o YOLOv8m, com a mesma divisão de épocas: um por 8, outro por 16 e o último por 32.

Durante o treinamento, o desempenho do modelo foi monitorado utilizando métricas como mAP (*mean Average Precision*), precisão, e *recall*. Essas métricas ajudam a entender como o modelo está aprendendo e a detectar possíveis problemas, como o overfitting, ao comparar o desempenho entre os conjuntos de treinamento e validação.

5.4 Pós-processamento

Após o treinamento, o modelo foi utilizado para realizar inferências em novas imagens ou sequências de imagens, identificando e desenhando caixas delimitadoras ao redor dos raios detectados. Essas inferências foram realizadas com o modelo treinado, e os resultados foram salvos para análise posterior.

5.4.1 Cálculo da velocidade média do raio

Para calcular a velocidade dos raios, o modelo foi aplicado a sequências de 8 imagens consecutivas, onde a posição das caixas delimitadoras foi rastreada ao longo do tempo.

Foi calculada então a distância entre as bordas inferiores na sequência de imagens, ou seja, entre a imagem 1 e 2, entre a 2 e 3, e assim por diante. Como os raios onde o modelo calcula a velocidade já tem a sua velocidade calculada em um arquivo excel, foi então extraído a constante que converte o valor de pixels para metros.

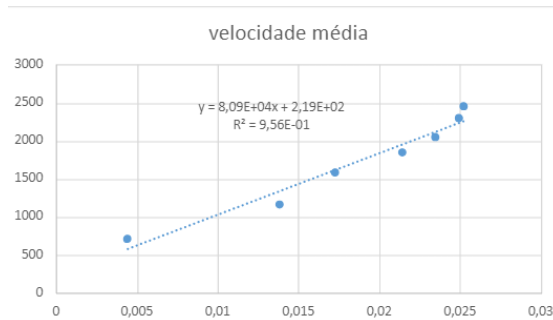
Fig. 4. Exemplo da planilha de cálculo da velocidade.

Tempo (s)	Distância (m)
0,004444	710,4208598
0,013809	1165,635986
0,017301	1581,963617
0,021428	1852,381667
0,023492	2044,317217
0,02492	2297,8611
0,025238	2449,599475

A partir desta constante, é feito o produto das distâncias em pixels entre as bordas pela constante, transformando as distâncias em

metros. Tendo então estas distâncias, é também extraído do arquivo excel o intervalo de tempo entre os frames, nos dando a possibilidade de montar um gráfico da distância percorrida pelo tempo.

Fig. 5. Gráfico feito a partir das distâncias e intervalos de tempo obtidos



A distância entre as bordas das caixas em pixels foi convertida em metros, e, utilizando o tempo entre as imagens, foi calculada a velocidade média do raio.

5.4.2 Análise dos resultados

Os resultados da detecção e cálculo de velocidade foram analisados para avaliar a precisão e a eficácia do modelo. Gráficos da precisão, recall e Mean average Precision foram gerados para visualizar a evolução do modelo ao longo das épocas de treinamento e para mostrar a velocidade dos raios em diferentes condições.

5.4.3 Ajustes finais

Com base na análise dos resultados, foram feitos ajustes finais no modelo e nas configurações de treinamento para otimizar ainda mais o desempenho. Isso inclui ajustes na taxa de aprendizado, aumento do conjunto de dados, e possíveis melhorias na arquitetura do modelo.

6. Resultados e Discussões

Agora, vamos comparar os seis modelos treinados anteriormente. Para a avaliação dos modelos, foi usada as métricas mAP (Mean Average Precision), precision e recall.

6.1 mAP (Mean Average Precision)

O desempenho do modelo foi primeiramente avaliado utilizando as métricas de mAP50 (mean Average Precision com IoU fixado em 0.50) e mAP50-95 (média de precisões calculadas para diferentes thresholds de IoU, de 0.50 a 0.95).

6.1.1 modelos de *batch size* = 8 mAP

Para os modelos de *batch size* = 8, vamos comparar os modelos YOLOv8s e YOLOv8m.

Fig. 6. Comparação entre o mAP50 dos dois modelos treinados.

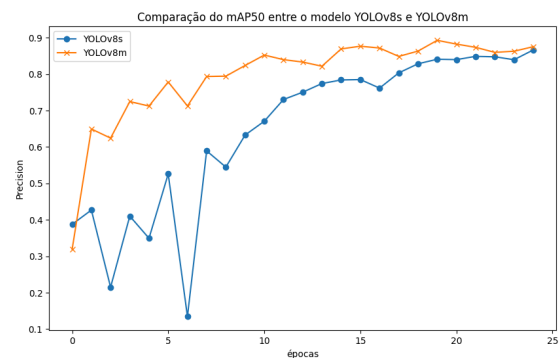
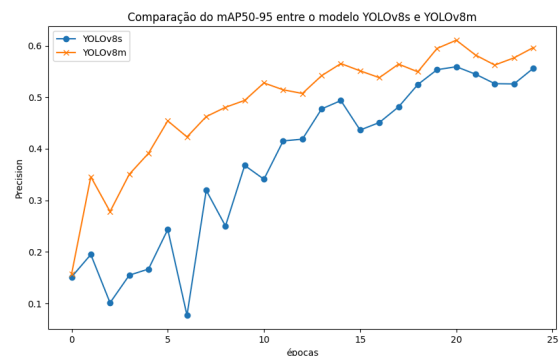


Fig. 7. Comparação entre o mAP50-95 dos dois modelos treinados.



Para o modelo YOLOv8s, os valores de mAP50 e mAP50-95 foram de 86.65% e 55.60%, respectivamente. O modelo YOLOv8m, por sua vez, apresentou um leve aumento na precisão, com valores de mAP50 atingindo 89.29% e mAP50-95 chegando a 61.05%.

Este resultado indica que, ao usar uma versão do modelo com mais capacidade (YOLOv8m), há uma melhoria na detecção de raios, especialmente quando critérios mais rigorosos de sobreposição são aplicados.

6.1.2 modelos de *batch size* = 16 mAP

Para os modelos de *batch size* = 16, vamos comparar os modelos YOLOv8s e YOLOv8m.

Fig. 8. Comparação entre o mAP50 dos dois modelos treinados.

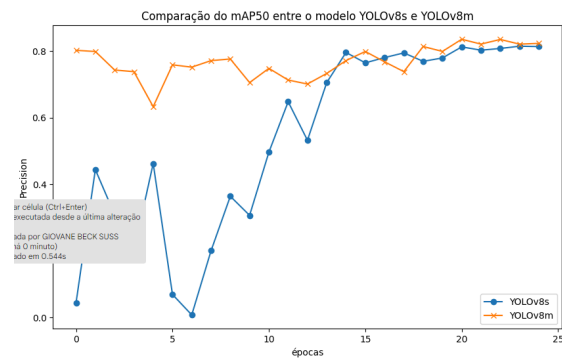
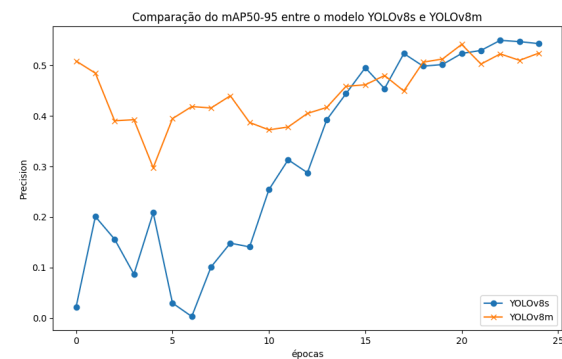


Fig. 9. Comparação entre o mAP50-95 dos dois modelos treinados.



Para o modelo YOLOv8s, os valores de mAP50 e mAP50-95 foram de 81.46% e 54.97%, respectivamente. O modelo

YOLOv8m, por sua vez, apresentou um leve aumento no valor de mAP50, atingindo 83.49%, e uma leve diminuição no mAP50-95, atingindo 54.16%.

6.1.3 modelos de *batch size* = 32 mAP

Para os modelos de *batch size* = 32, vamos comparar os modelos YOLOv8s e YOLOv8m.

Fig. 10. Comparação entre o mAP50 dos dois modelos treinados.

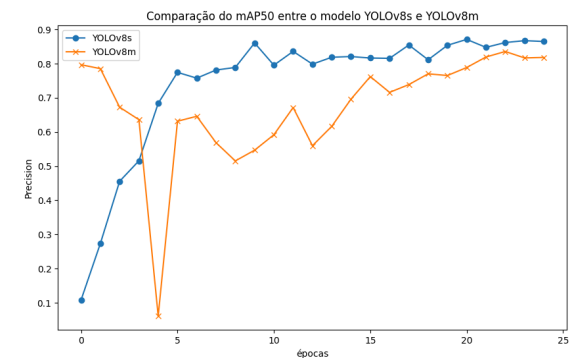
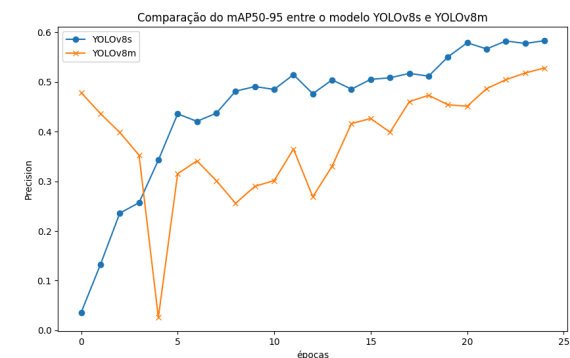


Fig. 11. Comparação entre o mAP50-95 dos dois modelos treinados.



Para o modelo YOLOv8s, os valores de mAP50 e mAP50-95 foram de 87.08% e 58.27%, respectivamente. O modelo YOLOv8m, por sua vez, apresentou uma leve diminuição na precisão, com valores de mAP50 atingindo 83.55% e mAP50-95 chegando a 52.79%.

6.2 Precisão e Recall.

Além do mAP, foram calculadas as métricas de precisão e recall. A precisão mede a proporção de predições corretas em relação ao total de predições feitas, indicando a exatidão do modelo. Já o recall avalia a proporção de predições corretas em relação ao total de instâncias positivas reais, refletindo a capacidade do modelo em identificar todas as ocorrências relevantes.

6.2.1 modelos de *batch size* = 8 *precision* e *recall*

Para os modelos de *batch size* = 8, vamos comparar os modelos YOLOv8s e YOLOv8m.

Fig. 12. Comparação entre o *precision* dos dois modelos.

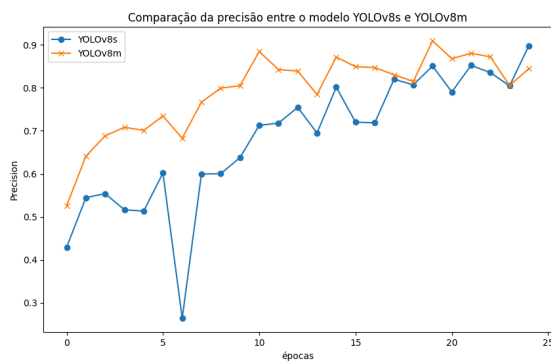
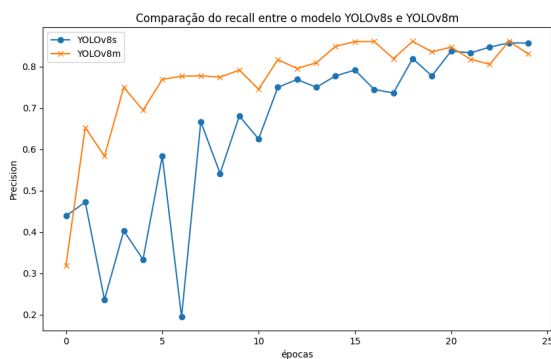


Fig. 13. Comparação entre o *recall* dos dois modelos.



Para o modelo YOLOv8s, os valores de *precision* e *recall* foram de 89.81% e 85.74%, respectivamente. O modelo YOLOv8m, por sua vez, apresentou um leve aumento, com

valores de *precision* atingindo 90.93% e *recall* chegando a 86.13%.

6.2.2 modelos de *batch size* = 16 *precision* e *recall*

Para os modelos de *batch size* = 16, vamos comparar os modelos YOLOv8s e YOLOv8m.

Fig. 14. Comparação entre o *precision* dos dois modelos.

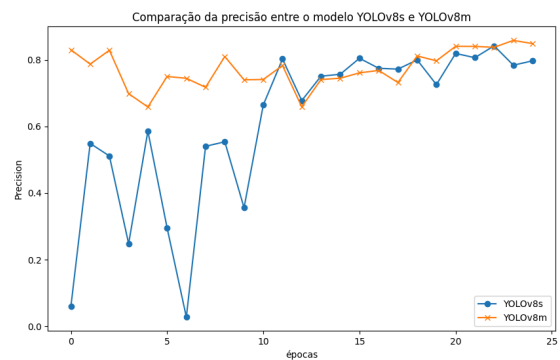
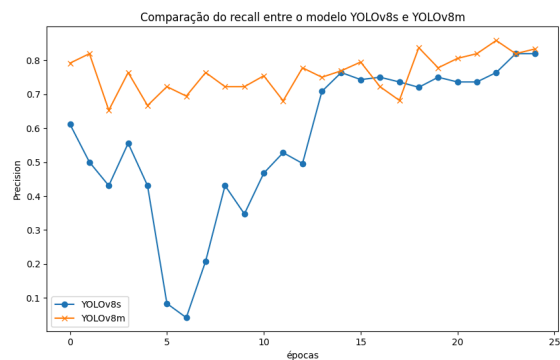


Fig. 15. Comparação entre o *recall* dos dois modelos.



Para o modelo YOLOv8s, os valores de *precision* e *recall* foram de 84.17% e 81.94%, respectivamente. O modelo YOLOv8m, por sua vez, apresentou um leve aumento, com valores de *precision* atingindo 85.81% e *recall* chegando a 85.83%.

6.2.3 modelos de *batch size* = 32 *precision* e *recall*

Para os modelos de *batch size* = 32, vamos comparar os modelos YOLOv8s e YOLOv8m.

Fig. 16. Comparação entre o *precision* dos dois modelos.

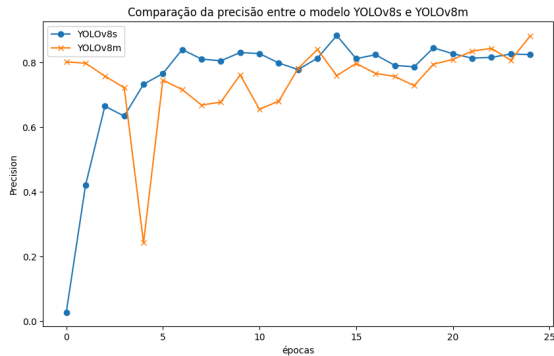
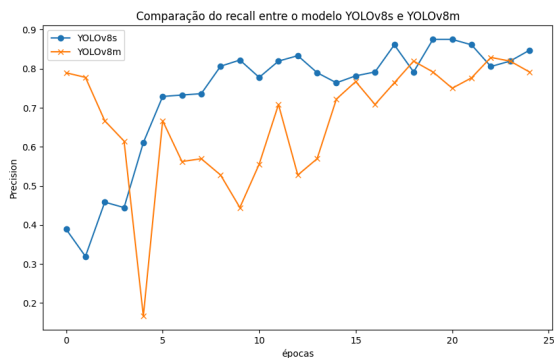


Fig. 17. Comparação entre o *recall* dos dois modelos.



Para o modelo YOLOv8s, os valores de *precision* e *recall* foram de 88.41% e 87.50%, respectivamente. O modelo YOLOv8m, por sua vez, apresentou uma leve diminuição, com valores de *precision* atingindo 88.22% e *recall* chegando a 82.88%.

6.3 Conclusão

Comparando o YOLOv8s e YOLOv8m, vemos que para um *batch size* de 8 e 16 o YOLOv8m é mais preciso, mas para um *batch size* de 32 o YOLOv8s é mais preciso, como podemos ver nos resultados acima.

Logo, fazendo uma comparação geral entre as versões YOLOv8s e YOLOv8m revelou que, embora ambas as versões apresentem um bom desempenho, o YOLOv8m oferece uma vantagem em termos de precisão e capacidade de detecção. Essa melhoria pode ser atribuída ao maior número de parâmetros e à maior

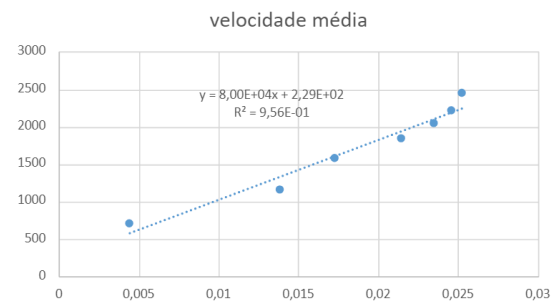
profundidade da rede, que permitem ao YOLOv8m capturar características mais detalhadas nas imagens.

No entanto, essa melhoria em precisão vem com o custo de maior tempo de inferência e maior uso de recursos computacionais. Para aplicações onde a velocidade e a eficiência computacional são críticas, o YOLOv8s ainda pode ser a melhor escolha, enquanto o YOLOv8m é mais adequado para cenários onde a máxima precisão é necessária e os recursos de processamento não são um limite.

6.4 Cálculo da velocidade

O cálculo da velocidade foi feita anteriormente de forma manual para uma sequência de imagens para um raio, que será usado agora para avaliar os modelos. A velocidade real do raio em 2D é igual a 8.00×10^4 m/s com o seguinte gráfico:

Fig. 18. Gráfico das velocidades reais do raio.

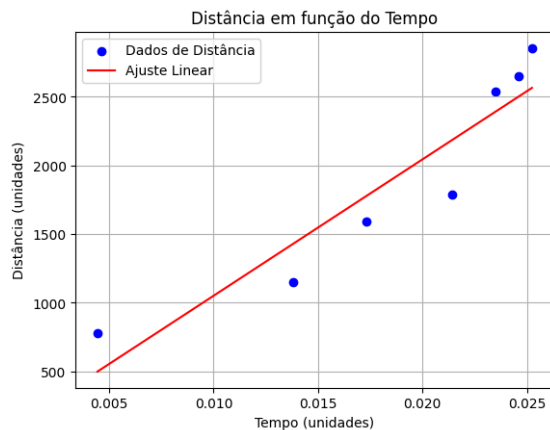


Para o cálculo da velocidade, foi feita a detecção para uma sequência de imagens de um raio que vai em direção ao chão. Após a detecção, é calculada a distância entre as bordas inferiores das caixas delimitadoras em pixels, e a partir do fator de conversão pixel/metro, que está na planilha onde já foi calculada a velocidade do raio anteriormente, é convertida esta distância entre as caixas para metros.

A partir das distâncias em metros e dos intervalos de tempos, também retirados da planilha onde já foi calculada a velocidade

anteriormente, é possível fazer um gráfico das velocidades do raio ao longo do seu caminho em direção ao solo calculadas a partir dos dados fornecidos pelo modelo, como mostra a imagem a seguir.

Fig. 19. Exemplo do gráfico das velocidades do raio.



A partir desse gráfico é calculado o coeficiente angular da reta a partir de um ajuste linear do gráfico, como é visto acima, nos dando a velocidade média dos pontos.

O modelo que apresentou melhor desempenho foi o YOLOv8s com *batch size* = 32, calculando uma velocidade de $9,94e+04$ m/s, que em comparação com a velocidade real, está relativamente próximo. O segundo melhor modelo foi o YOLOv8m com *batch size* = 32 que calculou uma velocidade média de $1,07e+05$ m/s.

Comparando o gráfico da figura 19, que seria o gráfico feito pelo modelo YOLOv8s com *batch size* = 32, com a figura 18, que é o gráfico da velocidade real do raio, vemos que não existem muitas diferenças entre os dois, logo podemos concluir que o modelo é preciso.

6.5 Teste de Wilcoxon

Para verificar se as diferenças observadas entre os desempenhos dos modelos YOLOv8s e YOLOv8m eram estatisticamente

significativas, foi realizado o teste de Wilcoxon.

6.5.1 Modelos com *batch size* = 8

Fazendo o teste de wilcoxon para os modelos de *batch size* = 8 para o mAP50 e mAP50-95.

Para o mAP50, o teste deu um p-valor igual a $2.56e-06$, que é muito menor que 0.05, logo rejeitamos a hipótese de que não há diferença, logo os dois modelos tem uma diferença significativa em relação a mAP50.

Para o mAP50-95, o teste deu um p-valor igual a $5.96e-08$, que também é muito menor que 0.05, logo confirmamos que existe uma diferença significativa em relação ao mAP50-95.

Fazendo as médias dos mAP50 e mAP50-95 dos dois modelos, podemos comparar e concluir qual é o melhor. Para mAP50, o YOLOv8m teve uma média de 0.79, superior a 0.64 do YOLOv8s. Para o mAP50-95, o YOLOv8m teve uma média de 0.48, superior a 0.37 do YOLOv8s.

Logo, podemos concluir que o modelo YOLOv8m é mais preciso que o YOLOv8s quando treinado com *batch size* = 8

6.5.1 Modelos com *batch size* = 16

Fazendo o teste de wilcoxon para os modelos de *batch size* = 16 para o mAP50 e mAP50-95.

Para o mAP50, o teste deu um p-valor igual a $3.19e-05$, que é muito menor que 0.05, logo rejeitamos a hipótese de que não há diferença, logo os dois modelos tem uma diferença significativa em relação a mAP50.

Para o mAP50-95, o teste deu um p-valor igual a $3e-03$ que também é menor que 0.05, logo confirmamos que existe uma diferença significativa em relação ao mAP50-95.

Fazendo as médias dos mAP50 e mAP50-95 dos dois modelos, podemos comparar e

concluir qual é o melhor. Para mAP50, o YOLOv8m teve uma média de 0.76, superior a 0.54 do YOLOv8s. Para o mAP50-95, o YOLOv8m teve uma média de 0.44, superior a 0.31 do YOLOv8s.

Logo, podemos concluir que o modelo YOLOv8m é mais preciso que o YOLOv8s quando treinado com *batch size* = 16

6.5.1 Modelos com *batch size* = 32

Fazendo o teste de wilcoxon para os modelos de *batch size* = 32 para o mAP50 e mAP50-95.

Para o mAP50, o teste deu um p-valor igual a 2e-02, que é muito menor que 5e-02, logo rejeitamos a hipótese de que não há diferença, logo os dois modelos tem uma diferença significativa em relação a mAP50.

Para o mAP50-95, o teste deu um p-valor igual a 1e-02 que também é menor que 5e-02, logo confirmamos que existe uma diferença significativa em relação ao mAP50-95.

Fazendo as médias dos mAP50 e mAP50-95 dos dois modelos, podemos comparar e concluir qual é o melhor. Para mAP50, o YOLOv8m teve uma média de 0.67, inferior a 0.74 do YOLOv8s. Para o mAP50-95, o YOLOv8m teve uma média de 0.38, inferior a 0.44 do YOLOv8s.

Logo, podemos concluir que o modelo YOLOv8s é mais preciso que o YOLOv8m quando treinado com *batch size* = 32

6.4 Pontos fortes

- **Precisão de Detecção:** Todos os 6 modelos, especialmente o YOLOv8s com *batch size*=32, demonstraram uma boa acurácia na detecção de raios.
- **Eficiência no Cálculo da Velocidade:** O processo de cálculo da velocidade foi significativamente acelerado pelo uso do YOLOv8, permitindo a análise de grandes

volumes de dados em menos tempo, mas a sua precisão ainda não é exata.

- **Flexibilidade do Modelo:** A possibilidade de ajustar o modelo para diferentes tamanhos e complexidades de rede (YOLOv8s vs YOLOv8m) e *batch size* permite adaptar a solução a diferentes cenários e necessidades.

6.5 Limitações

- **Dependência de Dados Anotados:** A precisão do modelo depende fortemente da qualidade e quantidade dos dados anotados, o que pode ser um gargalo em projetos com recursos limitados para anotação.
- **Recursos Computacionais:** O treinamento e a inferência do YOLOv8m requerem mais recursos computacionais, o que pode ser uma limitação em dispositivos com menor capacidade de processamento.
- **Precisão baixa da velocidade:** a precisão do cálculo da velocidade ainda é baixa, necessitando melhorias na base de dados e parâmetros.

7. Conclusão

Podemos concluir que em geral o modelo YOLOv8m demonstrou uma eficiência superior em comparação ao YOLOv8s, devido ao uso de um maior número de parâmetros, o que se refletiu em uma precisão mais elevada, embora com a necessidade de recursos computacionais mais robustos. Entretanto, o modelo mais preciso foi o YOLOv8s com *batch size* = 32, dando uma boa aproximação da velocidade média do raio. Os resultados obtidos com o YOLOv8m alcançaram uma precisão próxima a 85%, indicando que o modelo é eficaz na detecção de raios. Os resultados do YOLOv8s alcançaram uma precisão de 85% em média também.

No entanto, para aprimorar ainda mais a capacidade de análise, especialmente em casos de raios distantes ou com canais muito fracos,

é essencial expandir e diversificar a base de dados utilizada para o treinamento.

Apesar dos resultados promissores, há um amplo espaço para melhorias. Futuramente, pretende-se otimizar o modelo para aumentar a precisão na detecção de canais de raios descendentes e no cálculo de sua velocidade, explorando novas técnicas e ampliando o conjunto de dados. Esse trabalho contínuo será fundamental para avançar na análise de descargas elétricas atmosféricas, permitindo aplicações ainda mais precisas e robustas.

8. Referências

MAZUR. **Principles of Lightning Physics: Principles of Lightning Physics.** Ed. 20161201. National Severe Storms Laboratory, Norman, Oklahoma, USA, 2016. 183 p. ISBN 978-0-7503-1152-6.

MAZUR, V. et al. Recoil leader formation and development. **Journal of Electrostatics**, v. 71, n. 4, p. 763–768, ago. 2013.

COORAY, V. **An Introduction to Lightning.** Uppsala, Sweden: Springer, 2015. 391 p. ISBN(9789401789387)

NAG, A.; RAKOV, V. A. Some inferences on the role of lower positive charge region in facilitating different types of lightning. **Geophysical Research Letters**, v. 36, n. 5, 16 mar.2009.

SCHUMANN, C. **ESTUDO DOS RAIOS ASCENDENTES A PARTIR DE OBSERVAÇÕES DE CÂMERAS DE ALTA RESOLUÇÃO TEMPORAL E DE MEDIDAS DE CAMPO ELÉTRICO.** 2016. 180 p. (INPE - 05.04.19.06-TDI). São José dos Campos, SP: Dissertação (Doutorado em Geofísica Espacial) - Instituto Nacional de Pesquisas Espaciais – INPE.

CAMPOS, L. et al. **High-speed video observations of natural cloud-to-ground lightning leaders – A statistical analysis.**

2014. 21p. Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, P. O. Box 515, 12227-010, Brazil.

FERRO, M. A. S. **Afinal, o que, Raios, é isso.** Ed. Do Autor. São José dos Campos, SP: INPE, 2022. 210 p. ISBN 978-65-00-55553-0.