



Industrialização, Digitalização,
Desempenho

5º Simpósio Brasileiro de Tecnologia da Informação
e Comunicação na Construção e 5º Workshop de
Tecnologia de Processos e Sistemas Construtivos

FLORIANÓPOLIS-SC | 20 a 22 de agosto

1º ESTUDO COMPARATIVO DAS VERSÕES DA YOLO 11 VISANDO A AUTOMATIZAÇÃO DE INSPEÇÕES DE TELHADOS PARA FINS DE MANUTENÇÃO.

Comparative study of yolo 11 versions for automating roof inspections for maintenance purposes

Walisson Santos Oliveira

Universidade Federal da Bahia | Salvador, BA | walissonoliveira@ufba.br

Alisson Souza Silva

Universidade Federal da Bahia | Salvador, BA | so_alisson@hotmail.com

Pedro Afonso Vieira Fernandes Braga

Universidade Federal da Bahia | Salvador, BA | pedrofernandes@ufba.br

Dayana Bastos Costa

Universidade Federal da Bahia | Salvador, BA | dayanabcosta@ufba.br

RESUMO

O principal problema abordado neste artigo é a carência de estudos focados na aplicação de algoritmos avançados de detecção de objetos, como o YOLOv11, para inspeções automatizadas de telhados e coberturas de edificações, especialmente em cenários com características visuais complexas. Nesse contexto, a Construção 4.0 (C4.0) tem revolucionado os processos construtivos, com destaque para as inspeções automatizadas, que substituem métodos manuais por tecnologias como drones e Inteligência Artificial (IA). No entanto, há uma falta de análises detalhadas sobre a eficiência de variantes de arquiteturas de IA, como a YOLOv11, nesse cenário. O objetivo deste trabalho é avaliar comparativamente o desempenho dos modelos variantes da YOLOv11 (11n, 11s, 11m, 11l e 11x) na detecção e classificação de defeitos em telhados e coberturas. Para isso, foram criados cinco modelos baseados na YOLOv11, treinados com imagens coletadas por drones em 25 edificações públicas. Os resultados indicam que, ao ponderar desempenho e complexidade, a variante m da YOLOv11 apresenta melhor performance com 53,44% de mAP50 na validação. Conclui-se que a escolha da variante ideal não depende diretamente do indicador de desempenho, mas sim da capacidade de generalização do modelo associado à sua necessidade de recurso computacional devido a robustez.

Palavras-chave: Coberturas de edificações; Inspeções Automatizadas; Inteligência Artificial; Aprendizado de máquina.

ABSTRACT

The main problem addressed in this article is the lack of studies focused on the application of advanced object detection algorithms, such as YOLOv11, for automated inspections of building roofs and coverings, especially in scenarios with complex visual characteristics. In this context, Construction 4.0 (C4.0) has revolutionized construction processes, with emphasis on automated inspections, which replace manual methods with technologies such as drones and Artificial Intelligence (AI). However, there is a lack of reported analyses on the efficiency of AI architecture variants, such as YOLOv11, in this scenario. The objective of this work is to comparatively evaluate the performance of YOLOv11 model variants (11n, 11s, 11m, 11l and 11x) in the detection and classification of defects in roofs and coverings. For this, five models based on YOLOv11 were created, trained with images collected by drones in 25 public buildings. The results indicate that, when weighing performance and complexity, the m variant of YOLOv11 performs better with 53.44% of mAP50 in the validation. It is concluded that the choice of the ideal variant does not depend directly on the performance indicator, but rather on the generalization capacity of the model associated with its need for computational resources due to robustness.

Keywords: Building roofs; Automated Inspections; YOLOv11; Artificial intelligence; Machine learning.

¹OLIVEIRA, W. S., SILVA, A. S., BRAGA, P. A. V. F., COSTA, D. B. Estudo Comparativo das Versões da YOLO 11 Visando a Automatização de Inspeções de Telhados para Fins de Manutenção. In: 5º SIMPÓSIO BRASILEIRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO, 4., 2025, Florianópolis. *Anais [...]*. Porto Alegre: ANTAC, 2025.

1 INTRODUÇÃO

A Construção 4.0 (C4.0) representa um avanço significativo a respeito da integração de tecnologias digitais aos processos construtivos, possibilitando otimização, eficiência, redução de custos (Ferreira; Correa, 2019). Nesse cenário, um dos pilares dessa transformação é a utilização de processos automatizados, possibilitando o controle em tempo real, através do uso da Internet das Coisas (IoT) e Inteligência Artificial (IA), abrangendo desde o planejamento e design de projetos com BIM (Modelagem da Informação da Construção) até a execução (Aghabayli, 2021; Da Silva et al., 2018).

Dentre as diversas aplicações da C4.0, as inspeções automatizadas surgem com a finalidade de substituir os métodos tradicionais de inspeção que dependem do trabalho manual exaustivo para identificar problemas e defeitos construtivos (Dang et al., 2021). Tal solução, destaca-se pelo uso de drones equipados com câmeras de alta definição capazes de inspecionar ativos de infraestrutura de difícil acesso (Fan; Saadeghvaziri, 2019; Zahariadis et al., 2017), telhados e coberturas (Staffa et al., 2023) e fachadas (Chen et al., 2023). Alguns algoritmos de Inteligência Artificial (IA), em especial as redes neurais convolucionais (CNNs), tem alcançado conquistas surpreendentes (Lecun et al., 2015; Long et al., 2015), contribuindo com o processo de tomada de decisão, com a análise de dados e na execução de tarefas de inferência (Long et al., 2015; Liu, 2023; Zhang et al., 2024).

No campo da visão computacional (VC), sobressaem-se os avanços sobre os algoritmos de detecção de objetos, principalmente as arquiteturas da família *You Only Look Once* (YOLO) (Hui, 2023; Khanam; Hussain, 2024; Terven et al., 2023). Nesse contexto, a abordagem YOLO se destaca das arquiteturas tradicionais de detecção de objetos ao propor uma estrutura unificada capaz de prever simultaneamente caixas delimitadoras e probabilidades de classe, atendendo aos requisitos de processamento em tempo real (Koubaa et al., 2021). Hodiernamente, o desenvolvimento do algoritmo YOLO com a introdução do YOLOv11 representa um avanço significativo na tecnologia de detecção de objetos em tempo real, uma vez que se baseia nos pontos fortes de versões anteriores da YOLO, além de introduzir recursos que expandem sua utilidade dentro da VC (Sharma et al., 2024). Ademais, a YOLOv11 apresenta em sua arquitetura tais variantes: 11n, 11s, 11m, 11l e 11x, respectivamente, em ordem crescente de quantidade de parâmetros.

No contexto das aplicações da YOLO para inspeções automatizadas, apesar de avanços significativos, ainda existe uma carência de estudos focados na aplicação específica em cenários relacionados a inspeção de telhados e coberturas de edifícios. Em Wang et al. (2022) foi desenvolvido uma arquitetura baseada em YOLOx para detectar anomalias em estruturas de telhados de grandes vãos, enquanto que em Hou et al. (2023) o algoritmo YOLOv3 é utilizado para detectar peças decorativas em patrimônio histórico. Igualmente, Qiu et al. (2024) observa a utilização do algoritmo YOLOv8 para avaliar telhas faltantes e quebradas. No entanto, a literatura carece de trabalhos que utilizem o YOLOv11 no cenário das inspeções automatizadas de telhados. Além disso, a respeito dos estudos encontrados, poucos abordam analiticamente as arquiteturas e variantes envolvidas, na tentativa de se encontrar um modelo mais eficiente para detecção de anomalias estruturais em ambientes com características visuais complexas, como variações de iluminação, texturas irregulares, sobreposição e uma maior quantidade de rótulos.

Diante deste contexto, este trabalho visa avaliar quais variantes da YOLOv11 apresentam melhor desempenho na detecção e classificação de defeitos em telhados e coberturas de edificações. Para isso, foram desenvolvidos cinco modelos baseados na arquitetura da YOLOv11, sendo YOLOv11 n, s, m, l, e x. Esses modelos foram criados com base em imagens coletadas por drone em 25 telhados e coberturas de edificações públicas.

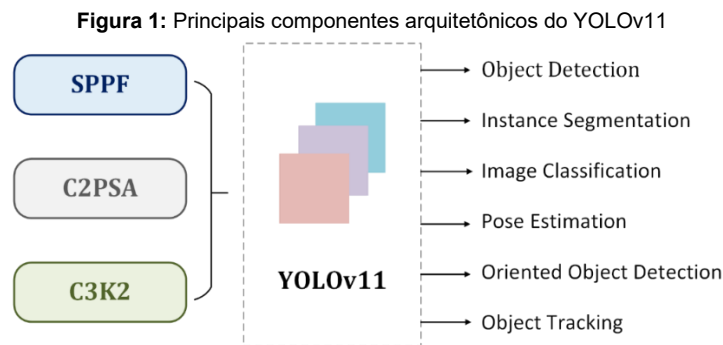
2 REVISÃO BIBLIOGRÁFICA

2.1 Arquitetura da Yolov11

Desenvolvida pela Ultralytics, a YOLOv11 é uma evolução da arquitetura YOLO, pois está baseada nos avanços de versões anteriores. Ademais, oferece cinco modelos escalonados, são eles n, s, m, l e x (respectivamente “nano”, “small”, “medium”, “large” e “extra-large”), podendo ser utilizada em aplicações como detecção de objetos, segmentação de instâncias, classificação de imagens, estimativa de pose e detecção de objetos orientados (Jegum et al., 2024), com eficiência computacional adaptável conforme

variante utilizada. Desta forma, quando comparada a outras versões, apresenta desempenho superior em tarefas complexas, uma vez que se introduziu melhorias com um poder de extração mais preciso.

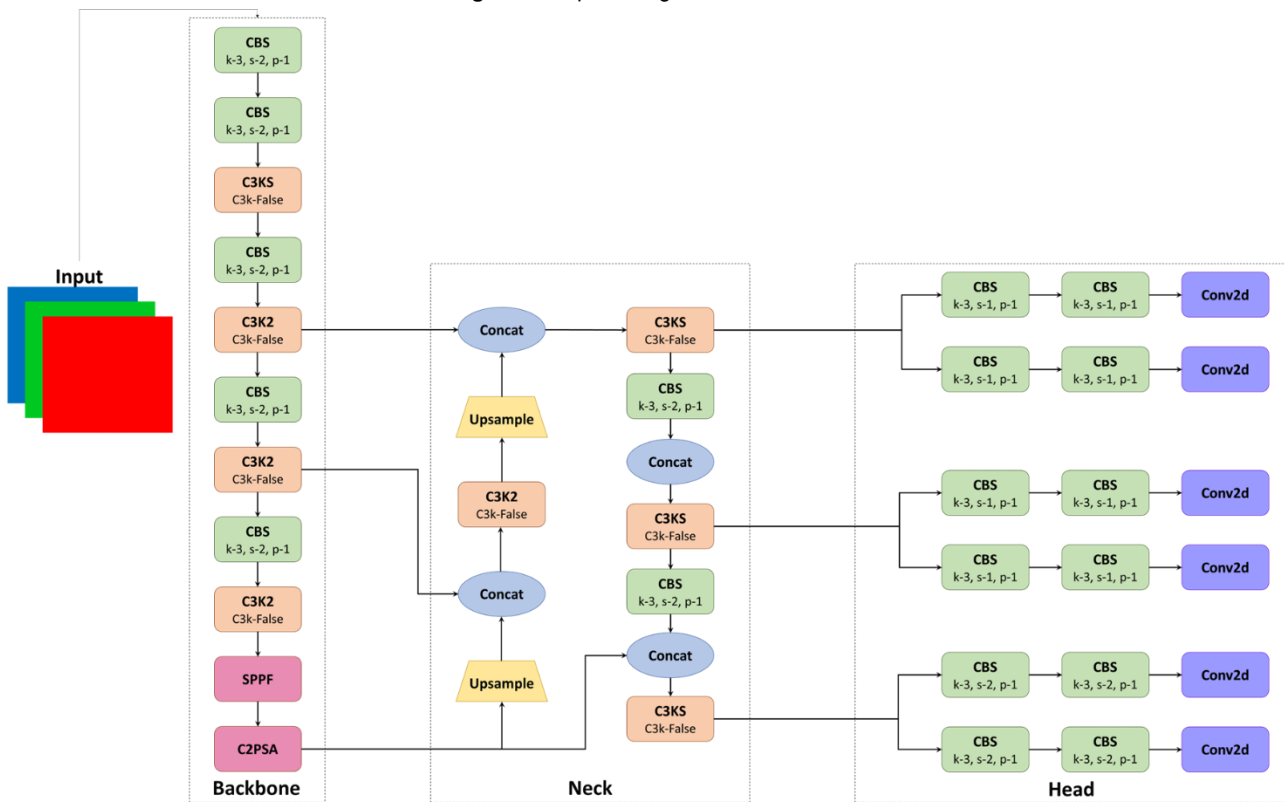
O YOLOv11 centrou-se na introdução da estrutura dos blocos C2PSA (*Cross-Stage Partial with Spatial Attention*), possibilitando que o modelo se concentrasse em regiões críticas na imagem. Ademais, trouxe um *backbone* reestruturado com kernel menores e camadas otimizadas, melhorando a velocidade de processamento sem sacrificar o desempenho. Outrossim, a inclusão do Spatial Pyramid Pooling-Fast (SPPF) permitiu uma combinação mais rápida de características extraídas de regiões distintas (Ali; Zhang, 2024), conforme figura 1.



Fonte: Khanam e Hussain (2024)

A arquitetura YOLOv11 é entendida em 3 subdivisões: *Backbone*, *Neck* e *Head* (figura 2). O componente *Backbone* é responsável por extrair recursos da imagem de entrada em múltiplas escalas. Esse processo envolve empilhar camadas convolucionais e blocos especializados, gerando mapas de recursos em várias resoluções. Posteriormente, o componente *Neck* é o estágio de processamento intermediário, empregando camadas especializadas para agregar e aprimorar representações de recursos em diferentes escalas, envolvendo *upsampling* e concatenação de mapas de características de diferentes níveis. Finalmente, o componente *Head*, responsável por gerar as previsões finais e processar os mapas de recursos passados pelo *Neck*, produzindo caixas delimitadoras e rótulos de classe para objetos dentro da imagem (Khanam; Hussain, 2024).

Figura 2: Arquitetura geral do YOLOv11

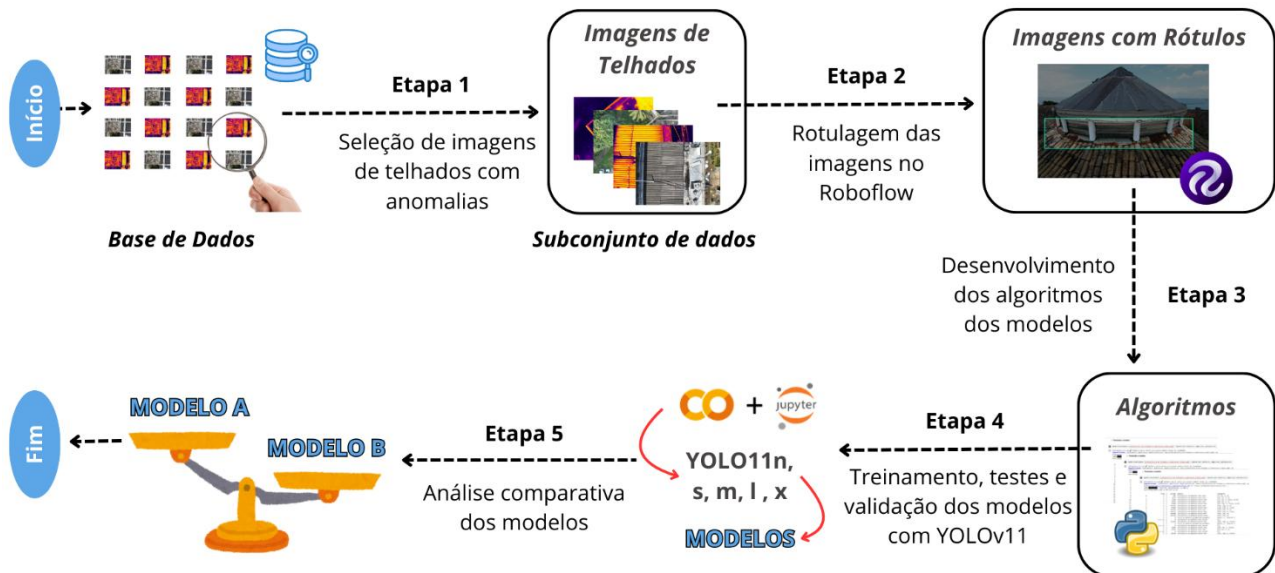


Fonte: Adaptado de Jegham et al. (2024)

3 MÉTODO DE PESQUISA

Este trabalho adotou uma estratégia de pesquisa de caráter exploratório, desenvolvida em cinco etapas (Figura 3), sendo: (1) Refinamento da base de dados, em que foram selecionadas, de um conjunto de dados maior, apenas imagens de telhados e coberturas de 25 edifícios públicos; (2) Rotulagem dos defeitos presentes nas imagens de telhados e coberturas; (3) Desenvolvimento dos algoritmos dos modelos em Python; (4) Treinamento, validação e testes dos modelos (70%, 20% e 10% do subconjunto de dados, respectivamente) utilizando as variantes da YOLOv11; e (5) Análise comparativa dos cinco modelos desenvolvidos, por meio de métricas com base no trabalho de Silva et al. (2025).

Figura 3: Delineamento de pesquisa

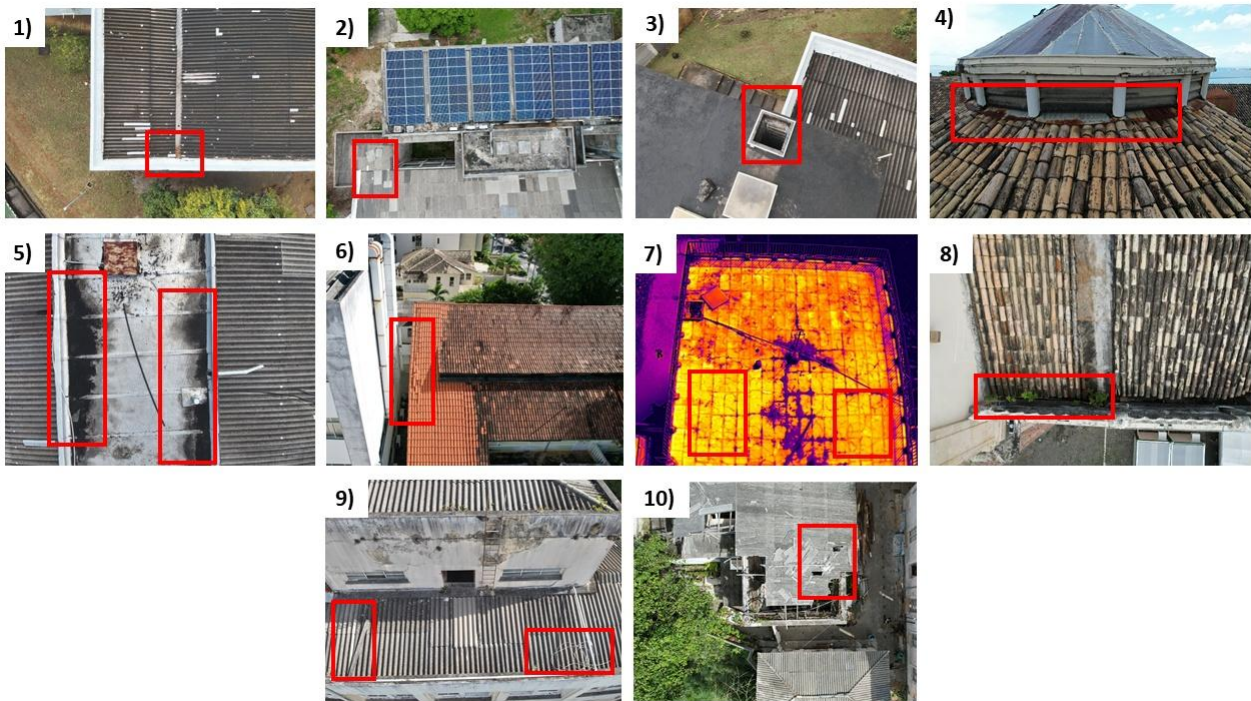


Fonte: Os autores

3.1 Base de dados

Esta etapa contemplou a estruturação da base de dados utilizada nesta pesquisa. As imagens foram coletadas ao longo de 25 inspeções, totalizando mais de 42.000 m² de telhados inspecionados, coletando um total de 7.937 imagens. A partir dessa base de dados, foi realizado um refinamento minucioso nas imagens, incluindo apenas imagens com defeitos em telhados e coberturas das edificações inspecionadas. Dessa forma, foi selecionado um subconjunto de dados composto por 7.210 imagens em dois formatos, RGB e termais. Após essa seleção foram definidas 10 classes de defeitos presentes nos telhados e coberturas (Figura 4), sendo: 1) Presença de sujeira na cobertura; 2) Presença de telha extra no telhado; 3) Presença de alçapões abertos; 4) Integridade de calhas e rufos; 5) Presença de biofilme; 6) Deslocamento das telhas; 7) Fissura em laje; 8) Presença de vegetação; 9) Presença de resíduo; e 10) Integridade das telhas.

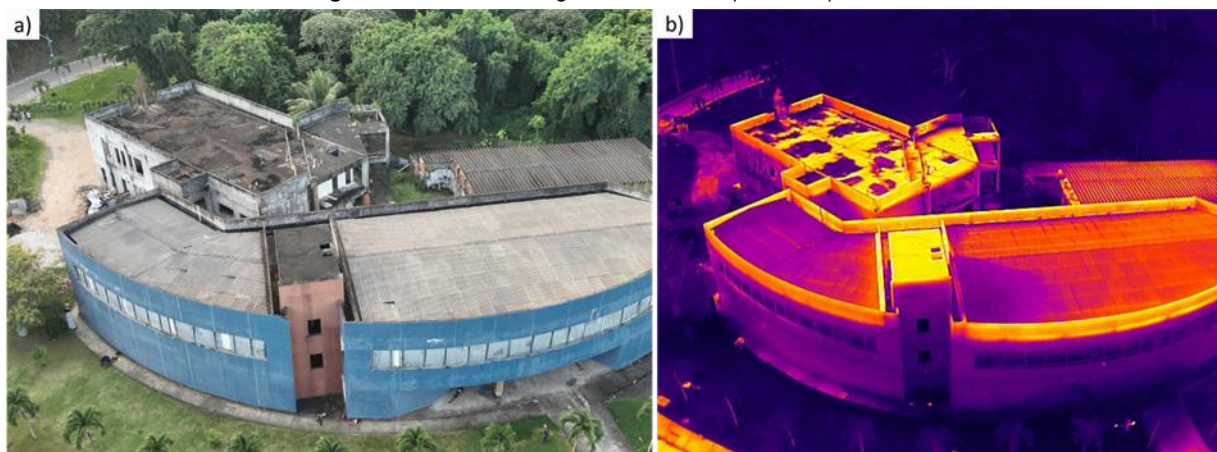
Figura 4: Exemplos das classes nos telhados e coberturas



Fonte: Os autores

Os dados utilizados nesse estudo contemplam imagens RGB e Termiais (Figura 5a e 5b), que foram divididas em 70% para etapa de treinamento, 20% validação e 10% testes. Segundo Politi et al. (2021), uma imagem termográfica é uma representação da radiação infravermelha emitida por um determinado objeto. Essa radiação, por sua vez, varia conforme a temperatura, mapeando as diferenças de calor. De mesmo modo, uma imagem RGB é uma representação digital das cores, onde cada pixel da imagem é composto basicamente por valores de intensidades de três cores primárias: vermelho, verde e azul (Jayaraman et al., 2009).

Figura 5: Formato de imagens utilizadas – a) RGB e b) termiais



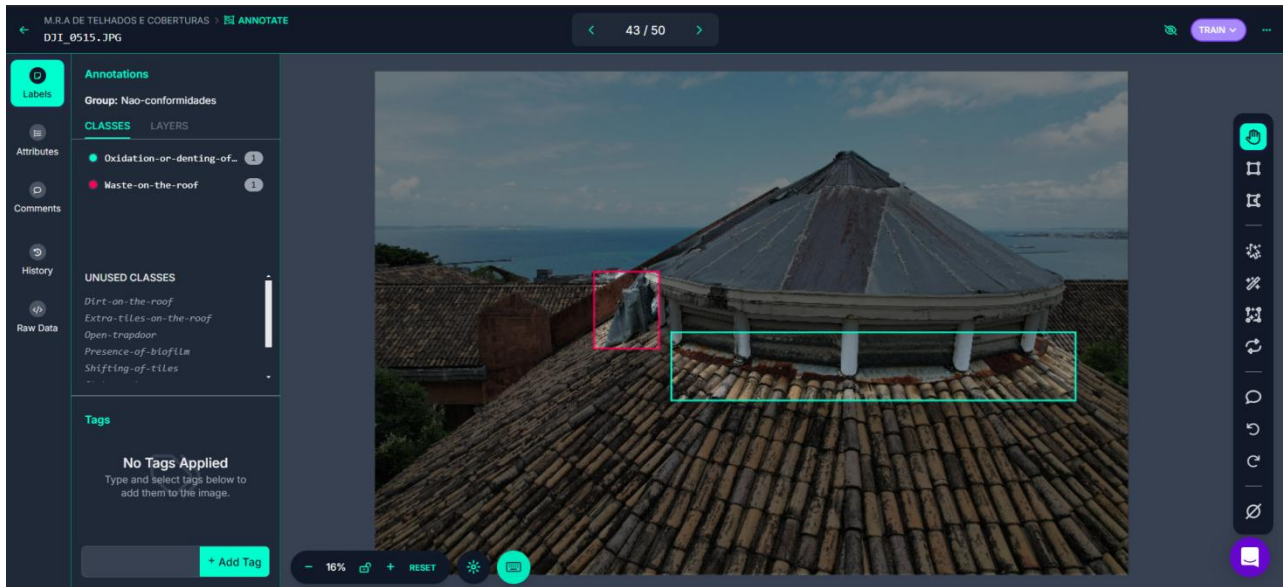
Fonte: Os autores

3.2 Desenvolvimento dos modelos

Após a criação do subconjunto de dados, a etapa de rotulagem das 7210 imagens foi feita no Roboflow de forma manual (Figura 6). Esse processo foi realizado entre 12 de julho a 19 de agosto de 2024, em uma média de 5 horas diárias de segunda a sexta, totalizando uma média de 135 horas. Observou-se durante a rotulagem um caráter de atividade de cunho subjetivo, onde a seleção dos rótulos dependia fortemente do

entendimento do indivíduo acerca da identificação da não conformidade, bem como um trabalho extenuante e cansativo, pois exigia do rotulador muitas horas diante da tela e realização de movimentos relativamente repetitivos.

Figura 6: Ambiente de rotulagem no Roboflow



Fonte: Os autores

Posteriormente, os algoritmos dos modelos foram desenvolvidos com arquitetura YOLOv11 nas plataformas do Google Colab e Jupyter Notebook, em Python. Consequentemente foram treinados, validados e testados 5 modelos, conforme tabela 1. Para realização deste processo, utilizou-se uma máquina com processador Intel core i7- 12700KF, 64 GB RAM DDR4 e placa de vídeo NVIDIA GeForce RTX 3080. O algoritmo base utilizado pode ser visualizado na tabela 2.

Tabela 1: Relação dos modelos e arquiteturas utilizadas

Modelos	Imagens	Arquitetura	Camadas	Nº de Parâmetros	Tempo de Treinamento
Modelo 1	RGB + Termais	Yolov11n	319	~2,6 milhões	17h:52min:52seg
Modelo 2	RGB + Termais	Yolov11s	319	~9,4 milhões	3h:39min:24seg
Modelo 3	RGB + Termais	Yolov11m	409	~20,1 milhões	5h:21min:52seg
Modelo 4	RGB + Termais	Yolov11l	631	~25,3 milhões	48h:25min:56seg
Modelo 5	RGB + Termais	Yolov11x	631	~56,9 milhões	58h:47min:23seg

Fonte: Os autores

Tabela 2: Algoritmo de treinamento dos modelos com YOLOv11

```

1 # Instalando bibliotecas necessárias
2 !pip install roboflow
3 !pip install ultralytics
4 # Importando bibliotecas
5 from roboflow import Roboflow
6 import ultralytics
7 ultralytics.checks() # Verifica instalação e compatibilidade das dependências
8 # Configuração do Roboflow para download do dataset
9 rf = Roboflow(api_key="SUA_CHAVE_DE_API_AQUI")
10 project = rf.workspace("getec-ufba").project("m.r.a-de-telhados-e-coberturas")
11 # Instanciamento do modelo YOLO pré-treinado
12 from ultralytics import YOLO
13 model = YOLO("yolov11n.pt") # Pode-se escolher as versões n, s, m, l e x
14 #Treinamento do modelo
15 model.train( data='/content/M.R.A-de-Telhados-e-Coberturas-1/data.yaml', epochs=300, batch=16, imgsz=640,
patience=10)

```

Fonte: Os autores

Os parâmetros utilizados na etapa de treinamento dos modelos são apresentados na tabela 3. Dentre eles, é o possível encontrar o “*Learning rate*”, correspondente a velocidade em que o algoritmo se ajusta aos dados; o “*Batch*”, tamanho do lote em cada iteração; as “*Epochs*”, iterações de treinamento; o “*Momentum*”, fator que influencia a incorporação de gradientes anteriores na atualização do atual; “*Patience*”, número de iterações a aguardar sem melhoria nas métricas de validação antes de parar o treino antecipadamente; e “*imgsz*”, que diz respeito a resolução de entrada das imagens no modelo.

Tabela 3: Valores dos parâmetros utilizados no treinamento dos modelos

Parâmetro	Valor utilizado
<i>Learning rate</i>	0.01
<i>Batch</i>	16
<i>Epochs</i>	300
<i>Momentum</i>	0.937
<i>Patience</i>	10
<i>imgsz</i>	640

Fonte: Os autores

3.3 Indicador de desempenho utilizado para avaliação dos modelos

A métrica utilizada foi o *mean Average Precision* (mAP), média da precisão média de todas as classes. Ela fornece uma medida quantitativa da precisão do modelo ao detectar e classificar objetos em uma imagem, levando em conta tanto a Precisão, proporção de casos previstos como positivos que são realmente positivos, e o Recall, número de casos positivos que foram corretamente identificados pelo modelo. Portanto:

$$mAP = \frac{1}{n} \sum_{k=1}^n AP_k \quad (1)$$

Onde n se refere a quantidade total de classes, enquanto que AP_k se refere a precisão média de cada classe k .

Adicionalmente, existem variantes do mAP, como mAP50 e mAP50-95, que diferenciam conforme o *IOU*, métrica calculada para quantificar a área de sobreposição entre a caixa delimitadora de previsão e a caixa delimitadora da verdade fundamental, dividido pela área de união entre eles. Essa métrica varia de 0, não há sobreposição entre as caixas, a 1, onde duas caixas preditas são idênticas.

O *AP* de uma classe k é determinado pela área sob a curva de Precisão x Recall, que avalia o desempenho do modelo em diferentes limiares de confiança. Assim, calcula-se o mAP seguindo tais passos: 1) O modelo gera uma lista de detecções ordenadas de acordo com o valor de confiança correspondente a cada predição; 2) O *IOU* de cada predição é analisado, se for maior ou igual ao limiar, determinam-se as

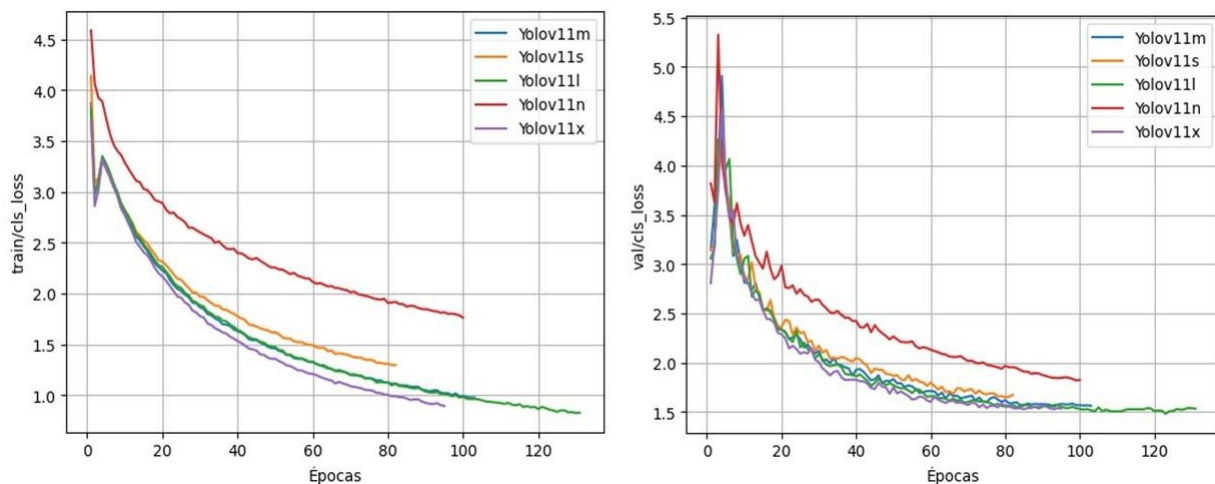
detecções corretas, caso contrário indica uma previsão incorreta; 3) Gera-se a curva de Precisão x Recall com base nas previsões; 4) Calcula-se o **AP**; 5) Determina-se o mAP, conforme (1).

Um mAP alto indica que o modelo tem boa precisão e recall, detectando corretamente os objetos e evitando previsões erradas. Todavia, um mAP baixo sugere que o modelo não está detectando os objetos corretamente ou está gerando muitas previsões incorretas.

4 RESULTADOS E DISCUSSÃO

Durante a fase de treinamento e validação, as perdas de classificação (cls_loss) dos modelos podem fornecer importantes indicadores sobre como os modelos estão se comportando durante o treinamento e como generalizam para dados não visualizados. O cls_loss no treinamento indica quão bem o modelo está aprendendo a classificar as imagens nos rótulos corretos, enquanto que na validação essa perda se refere ao desempenho de generalização dos modelos. Os dados do cls_loss podem ser visualizados na figura 7.

Figura 7: Perdas de classificação no treinamento e validação



Fonte: Os autores

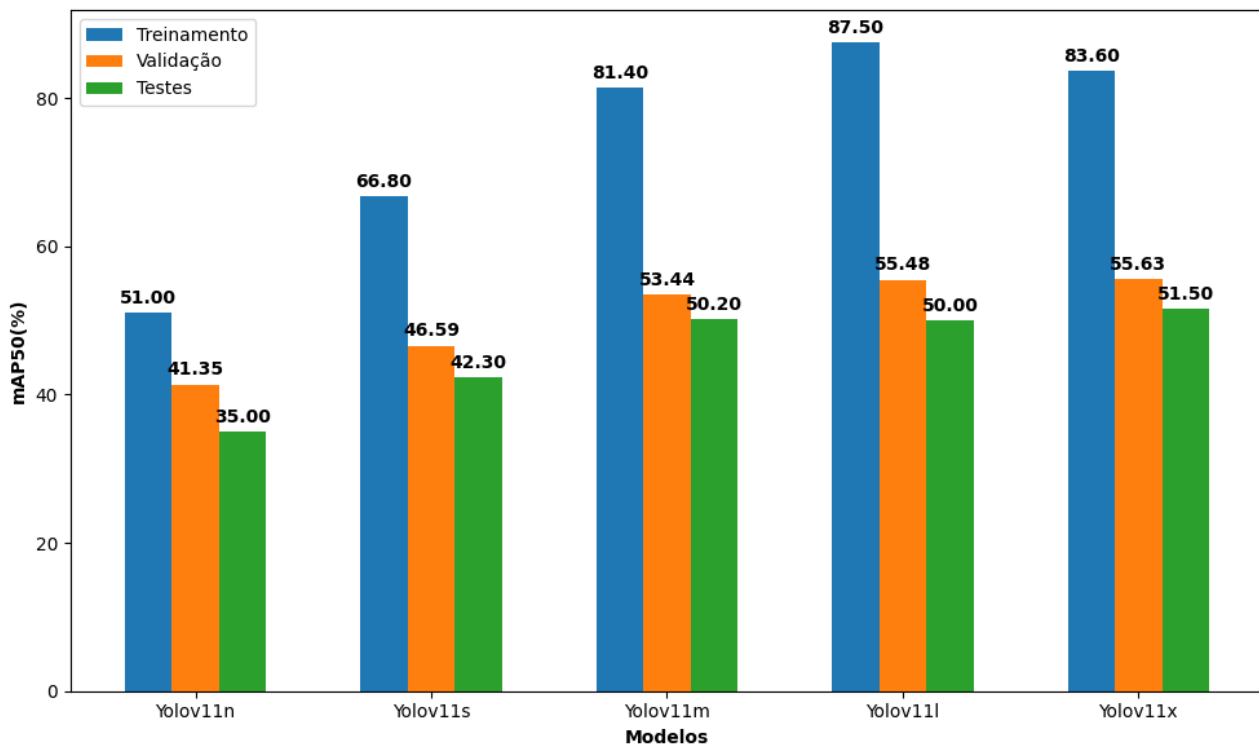
A figura 7 indica que o modelo YOLOv11n apresenta perda alta de classificação tanto no treinamento, quanto na validação. Tal fator, pode ser explicado pelo fato da variante “n” ter uma quantidade menor parâmetros. Todavia, o modelo YOLOv11x se apresenta com os menores valores de perda no decorrer das épocas, indicando que está aprendendo bem a identificar e generalizar as classes, quando comparado com as outras versões da YOLOv11. Além disso, a curva de cls_loss, tanto na validação quanto no treinamento, diminuem ao longo das épocas, indicando que o modelo pode não apresentar “sobreajuste”.

Analisando o mAP50 (figura 8), a variante n da YOLOv11 tem o menor desempenho em todos os conjuntos de dados. Ademais, o modelo YOLOv11s melhora em comparação com a n, mas somente a YOLOv11m consegue apresentar um salto significativo no desempenho com 81,4% de mAP50 no treinamento, 53,44% de mAP50 na validação e 50,20% de mAP50 nos testes. No entanto, são os modelos mais robustos que apresentam maiores valores de mAP50, em especial o modelo YOLOv11l com desempenho de 87,50% de mAP50, resultado ligeiramente melhor que os demais modelos na fase treinamento. Todavia, o modelo YOLOv11x apresentou um valor de mAP50 igual a 55,6% na etapa de validação, superando o YOLOv11l, que ficou entre 55,4% do mesmo indicador. Esse resultado mostra que embora a variante x apresente mais parâmetros, para o respectivo conjunto de dados, em comparação com a variante l, não há ganhos significativos desse mesmo indicador. Entretanto, pode se observar que a variante x da YOLOv11, para o problema de classificação em estudo, apresenta uma qualidade média de previsão de 55,6%, ou seja, o modelo em questão acerta em média 55,6% quando o *IOU* é maior ou igual a 50%.

Contrastando a figura 7, a figura 8 demonstra que os modelos ao apresentarem uma diferença relativa do treinamento para a validação e testes, principalmente o modelo YOLOv11l com discrepância do treinamento

para a validação de 32,02% de mAP50 e do treinamento para os testes de 37,5% mAP50, o *overfitting* está presente em todos os modelos, indicando má generalização para novos dados. Comparando os modelos, o *overfitting* se pronuncia na variante l de maneira mais acentuada. Contudo, o modelo YOLOv11m configura como o melhor modelo por apresentar similar capacidade de generalização que o modelo YOLOv11x e YOLOv11l, além de superar todos os demais. Outrossim, apresenta menor grau de *overfitting*, sendo 0,01% melhor que a variante x ao verificar a discrepância relativa do treinamento para validação, mesmo possuindo uma quantidade menor de parâmetros e exigindo menos recursos computacionais.

Figura 8: Comparativo entre o mAP50 dos modelos

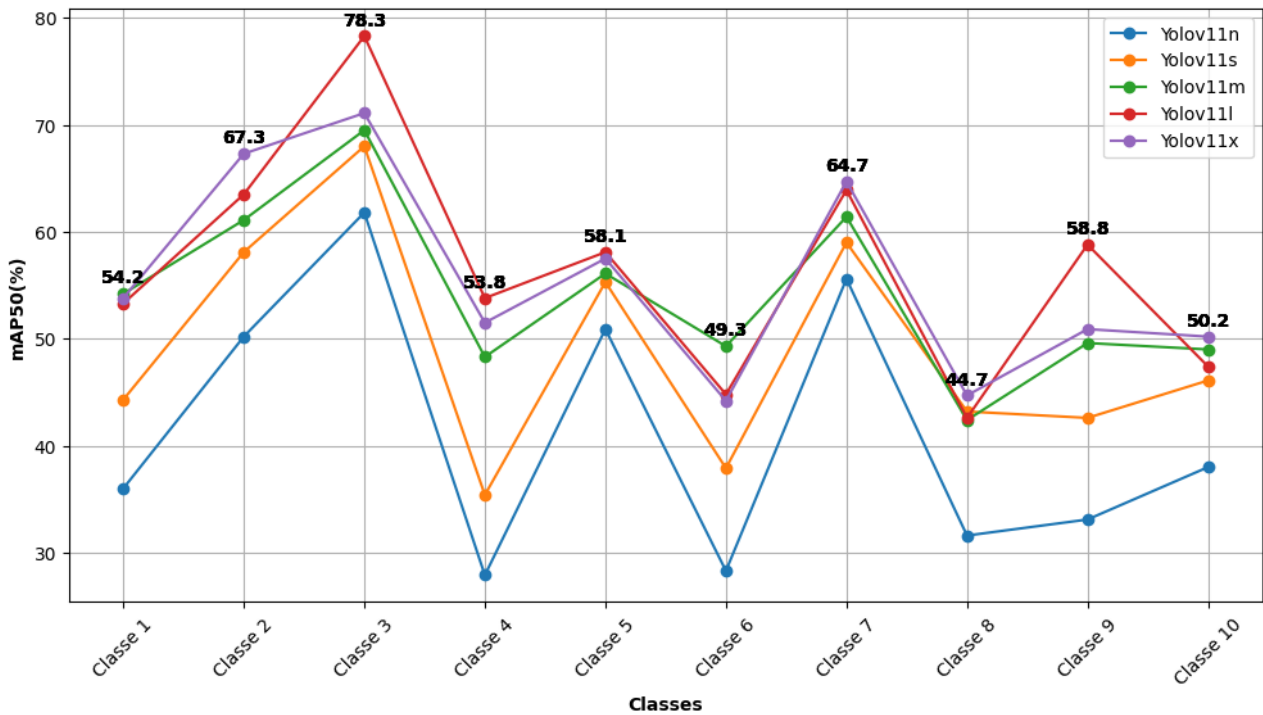


Fonte: Os autores

Analisando o desempenho dos modelos com relação às classes (figura 9), verifica-se que o modelo YOLOv11n apresentou menor indicador de mAP50 em todas as classes, seguido da variante s da YOLOv11 que conseguiu ser melhor que a n ao ter melhores indicadores por classe, superando às variantes m e l para a classe 8. Similarmente, a YOLOv11m, teve performance por classe melhor que as variantes s e m, entretanto destacou-se por apresentar melhor indicador nos rótulos 1 e 6, sendo 54,2% e 49,3% de mAP50, respectivamente. Além disso, obteve 49% de mAP50 para a classe 10, apenas perdendo para a variante x. Ademais, a variante l performou melhor nas classes 3, 4, 5 e 8, enquanto que a x só apresentou melhor desempenho nos rótulos 2, 7 e 10. Entretanto, o destaque maior do modelo YOLOv11l acontece nas classes 3 e 9, em que apresenta 78,2% e 85,8% de mAP, respectivamente, significando que a maioria das previsões para essas classes são corretas e bem localizadas. Ou seja, nas demais classes, a variante l apresenta desempenho similar às variantes x e m.

Ainda sobre a figura 10, somente as classes 2,3 e 7 obtiveram modelos com performance acima dos 60%. De forma mais crítica, a classe 6 não apresentou nenhum modelo com mAP50 maior ou igual a 50%, indicando que a maioria das previsões para essa classe não estão corretas ou que, possivelmente, os dados não estão sendo suficientemente representativos para os modelos criarem uma hipótese consistente.

Figura 9: Desempenho dos Modelos por Classe.



Fonte: Os autores

Selecionaram-se 20 imagens de forma aleatória no conjunto de testes para averiguar o desempenho dos modelos de variantes m, l e s conforme as predições correspondentes. Logo, verificou-se que os resultados das predições demonstraram que os modelos conseguiram inferir corretamente a maioria dos rótulos apresentados, mas pecaram de forma individualizada para algumas situações. Dentre elas, o modelo YOLOv11m e YOLOv11l apresentou dificuldades em prever corretamente a classe 6, deslocamento nas telhas, para o recorte do conjunto de testes realizado, enquanto a variante x conseguiu se sair melhor. Todavia, em situações de presença da classe 1, presença de sujeira, o modelo YOLOv11x não identificou. Com relação à classe 5, presença de biofilme, todos os modelos fizeram uma confusão com regiões mais escuras na imagem. As figuras 10, 11 e 12 ilustram melhor essas situações.

Figura 10: Exemplos de predições do modelo YOLOv11m. Nota: A classe 5 corresponde a presença de biofilme, e a classe 1 presença de sujeira na cobertura.



Fonte: Os autores

Figura 11: Exemplos de predições do modelo Yolov11l. Nota: A classe 5 corresponde a presença de biofilme, e a classe 1 presença de sujeira na cobertura.



Fonte: Os autores

Figura 12: Exemplos de predições do modelo Yolov11x. Nota: A classe 5 corresponde a presença de biofilme, e a classe 1 presença de sujeira na cobertura.



Fonte: Os autores

Ao analisar os estudos anteriores, observa-se que esse trabalho apresenta contribuição com relação a detecção e classificação de defeitos em telhados e coberturas de edificações, sobretudo quando os objetos nas imagens não são fáceis de serem localizados. Com relação ao estudo de Hou et al. (2023), que atingiu desempenho de 86,48% de mAP ao apresentar o algoritmo Yolov3 para utilização na detecção de peças decorativas em patrimônio histórico, o modelo com YOLOv11m apresentou indicador consideravelmente menor, de cerca de 33,04% de mAP50 de discrepância. Todavia, alguns fatores podem elucidar melhor respectivos resultados: Os dados utilizados por Hou et al. (2023) apresentam os objetos de forma mais evidente nas imagens, além do conjunto de dados maior e mais variado, permitindo que a rede convolucional generalize de uma forma mais precisa.

Outrossim, Wang et al. (2022) desenvolveu uma arquitetura baseada em YOLOx para detectar anomalias em estruturas de telhados de grandes vãos, atingindo 75,28% de mAP, à medida que Qiu et al. (2024) utilizou algoritmo YOLOv8 para avaliar telhas faltantes e quebradas, alcançando 89,4% de mAP. Mesmo o modelo com YOLOv11m apresentando diferença relativa de 21,84% e 35,96% de mAP, respectivamente, observou-se que tais trabalhos utilizaram uma quantidade de rótulos bem menor do que a utilizada no modelo m. Em Wang et al. (2022) utilizaram-se 4 classes, enquanto em Qiu et al. (2024) valeu-se de apenas 6 rótulos, justificando o melhor desempenho apresentado, bem como algumas confusões realizadas entre as classes pelo modelo com YOLOv11m.

5 CONSIDERAÇÕES FINAIS

A automatização de inspeções de telhados para fins de gestão da manutenção nem sempre é uma tarefa simples de ser realizada, mesmo com avanços significativos de algoritmos de detecção de objetos, como a YOLOv11. Modelos de detecção e classificação de não conformidades nos telhados e cobertura de edifícios baseados em visão computacional usando a arquitetura da YOLOv11 foram estabelecidos, treinados, avaliados e testados. Sendo assim, neste estudo uma análise comparativa de diferentes modelos a partir das variantes n, s, m, l e x da YOLOv11 foi feita, equilibrando desempenho e complexidade. Atingiu-se o resultado de 53,44% de mAP50 na validação com o modelo YOLOv11m. Portanto, as principais conclusões são:

1. O estudo comparativo entre as variantes da YOLOv11 para o conjunto de dados e problema apresentado demonstrou que as diferenças em termos de indicador de desempenho apresentadas entre o modelo YOLOv11m, YOLOv11l e YOLOv11x são pequenas, mesmo a variante x apresentado uma quantidade de parâmetros maior.
2. A etapa de pré processamento e análise dos dados, bem como uma seleção mais detalhada dos rótulos a serem identificados nas imagens podem contribuir significativamente com a performance final dos modelos a serem desenvolvidos.
3. Algoritmos com arquiteturas menos sofisticadas são capazes de se saírem melhor em determinados cenários do aprendizado de máquina.
4. A minimização de erros na etapa de rotulagem das imagens pode permitir que a máquina construa uma melhor hipótese dos dados apresentados.
5. Indicadores de desempenho analisados isoladamente nem sempre refletem a capacidade dos modelos. Deve-se analisar o seu poder de generalização, bem como o seu custo computacional.

Em relação às limitações, o tempo excessivo dedicado à etapa de rotulagem das imagens comprometeu outras fases do estudo, como a análise dos dados. Ademais, houve dificuldades com o pré-processamento das imagens, devido à complexidade dos cenários, presença de múltiplos rótulos, sobreposição de objetos e à dificuldade de identificação de elementos. Assim, estudos futuros podem aprimorar os modelos reduzindo o número de classes, utilizando uma base de dados mais robusta e detalhada e propondo estratégias para mitigar o *overfitting*. Além disso, a inclusão de ajustes nos hiperparâmetros pode contribuir para o aprimoramento do modelo, permitindo identificar variações que melhorem os indicadores de reconhecimento.

6 AGRADECIMENTOS

Os autores agradecem a CAPES e ao CNPq por meio do número 443942/2024-2, pelo suporte financeiro para a pesquisa e pela equipe da Superintendência de Meio Ambiente e Infraestrutura da UFBA (SUMAI-UFBA).

REFERÊNCIAS

- AGHABAYLI, Aydan. Machine Learning Applied to Building Information Models. 2021. Dissertação de Mestrado. Universidade do Minho (Portugal).
- ALI, Momina Liaqat; ZHANG, Zhou. The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers*, v. 13, n. 12, p. 336, 2024.
- ALZARRAD, Ammar et al. Automatic assessment of roofs conditions using artificial intelligence (AI) and unmanned aerial vehicles (UAVs). *Frontiers in Built Environment*, v. 8, p. 1026225, 2022
- CHEN, Kaiwen et al. GIS-based information system for automated building façade assessment based on unmanned aerial vehicles and artificial intelligence. *Journal of Architectural Engineering*, v. 29, n. 4, p. 04023032, 2023.
- DA SILVA, Alice Duarte; SIMÃO, A. dos S.; MENEZES, C. A. G. Impactos da Indústria 4.0 na Construção Civil brasileira. *Simpósio de Excelência em Gestão e Tecnologia*, v. 15, 2018.
- DANG, L. Minh et al. Deep learning-based sewer defect classification for highly imbalanced dataset. *Computers & Industrial Engineering*, v. 161, p. 107630, 2021.
- FAN, Jin; SAADEGHVAZIRI, M. Ala. Applications of drones in infrastructures: Challenges and opportunities. *International Journal of Mechanical and Mechatronics Engineering*, v. 13, n. 10, p. 649-655, 2019.

- FERREIRA JR, Cláudio Barboza; CORREA, Fabiano Rogerio. Construção 4.0–virtualização do processo construtivo. In: WORKSHOP DE TECNOLOGIA DE PROCESSOS E SISTEMAS CONSTRUTIVOS. 2019. p. 1-5.
- HOU, Miaole et al. A detection method for the ridge beast based on improved YOLOv3 algorithm. *Heritage Science*, v. 11, n. 1, p. 167, 2023.
- HUSSAIN, M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, 11 (7), 677 [em linha]. 2023.
- JAYARAMAN, Subramania; ESAKKIRAJAN, S.; VEERAKUMAR, T. *Digital image processing*. New Delhi: Tata McGraw Hill Education, 2009.
- JEGHAM, Nidhal et al. Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors. *arXiv preprint arXiv:2411.00201*, 2024.
- KHANAM, Rahima; HUSSAIN, Muhammad. YOLOv11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.
- KOUBAA, Anis et al. Cloud versus edge deployment strategies of real-time face recognition inference. *IEEE Transactions on Network Science and Engineering*, v. 9, n. 1, p. 143-160, 2021.
- LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *nature*, v. 521, n. 7553, p. 436-444, 2015.
- LIU, Hongxia. Intelligent Operation and Maintenance Management Method and System of Engineering Project Based on BIM. In: 2023 4th IEEE Global Conference for Advancement in Technology (GCAT). IEEE, 2023. p. 1-6.
- LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. p. 3431-3440.
- POLITI, Stefano et al. Infrared thermography images acquisition for a technical perspective in screening and diagnostic processes: protocol standardized acquisition. *Cureus*, v. 13, n. 11, 2021.
- QIU, Haochen et al. Research on intelligent monitoring technology for roof damage of traditional Chinese residential buildings based on improved YOLOv8: taking ancient villages in southern Fujian as an example. *Heritage Science*, v. 12, n. 1, p. 231, 2024.
- SHARMA, Akhilesh; KUMAR, Vipin; LONGCHAMPS, Louis. Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species. *Smart Agricultural Technology*, v. 9, p. 100648, 2024.
- STAFFA, Luciano de Brito Junior et al. Web platform for building roof maintenance inspection using UAS and artificial intelligence. *International Journal of Building Pathology and Adaptation*, 2025.
- SILVA, Alisson Souza et al. Modelo baseado em CNN para reconhecimento automatizado de anomalias na execução de fachadas visando apoio à Gestão da Qualidade. *Ambiente Construído*, v. 25, p. e143288, 2025.
- TERVEN, Juan; CÓRDOVA-ESPARZA, Diana-Margarita; ROMERO-GONZÁLEZ, Julio-Alejandro. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, v. 5, n. 4, p. 1680-1716, 2023.
- WANG, Pujin et al. Automatic ceiling damage detection in large-span structures based on computer vision and deep learning. *Sustainability*, v. 14, n. 6, p. 3275, 2022.
- ZAHARIADIS, Theodore et al. Preventive maintenance of critical infrastructures using 5G networks & drones. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2017. p. 1-4.
- ZHANG, Yongcheng et al. An Integrated Method Using a Convolutional Autoencoder, Thresholding Techniques, and a Residual Network for Anomaly Detection on Heritage Roof Surfaces. *Buildings*, v. 14, n. 9, p. 2828, 2024.