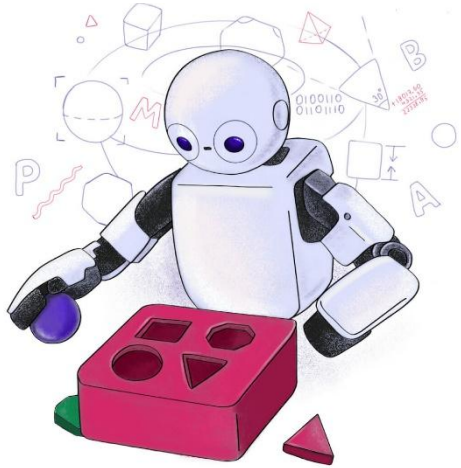
TP558 - Tópicos avançados em Machine Learning:

Small Object Detection Algorithm Based on Improved YOLOv8 for Remote Sensing



Inatel

Jonas Vilasboas Moreira
jonasmoreira@dtel.inatel.br

Introdução

O artigo propõe um **algoritmo de detecção de pequenos objetos** em **imagens de sensoriamento remoto**, chamado **LAR-YOLOv8**, que é uma versão melhorada do modelo YOLOv8.

O problema que ele resolve: **objetos pequenos**, como carros ou aeronaves vistos de satélites ou drones, são difíceis de identificar devido ao **tamanho reduzido**, ao **fundo confuso** e à **baixa resolução das imagens**.

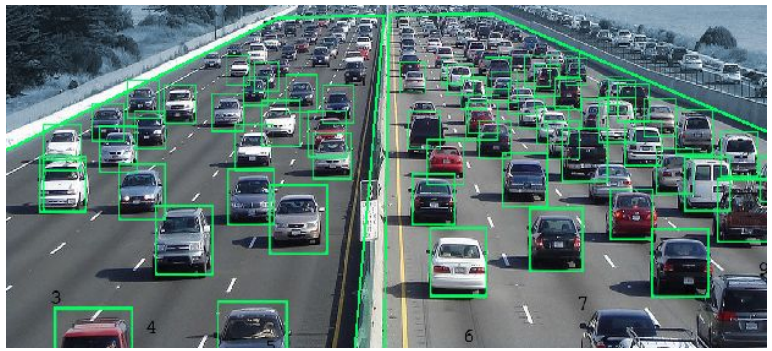
Introdução

Contexto

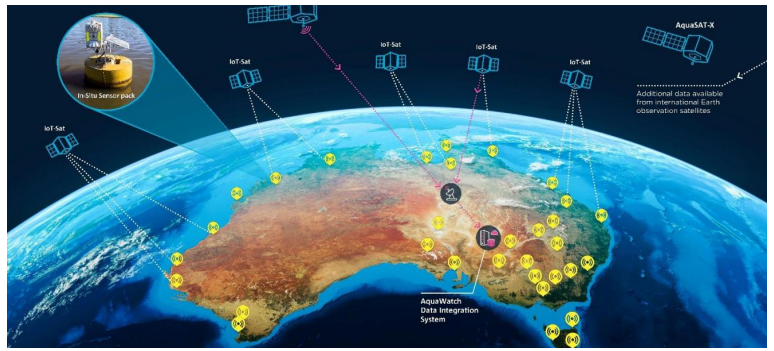
- Aumento da quantidade e qualidade de imagens geradas através de sensoriamento remoto.

Exemplo:

Monitoramento com Drones



Monitoramento de tráfego

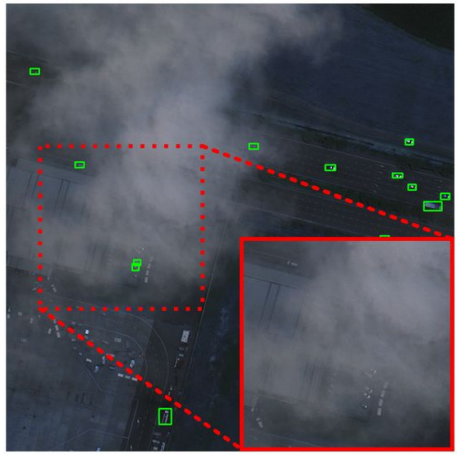


Satélites e aeroespacial

Introdução

Desafio

- Detecção de alvos pequenos.



Imagens com “ruído”

Objetos em contextos complexos

Objetos muito parecidos

Introdução

Motivação

Modelos atuais apresentam bom desempenho em imagens normais, porém melhorias são necessárias para imagens de sensoriamento remoto, pois elas apresentam grande variação na escala dos objetos, e são suscetíveis a fatores como iluminação e condições climáticas.

O objetivo do trabalho é desenvolver um modelo de detecção de objetos em sensoriamento remoto mais eficiente e robusto, baseado no YOLOv8.

Introdução

Contribuições do Trabalho

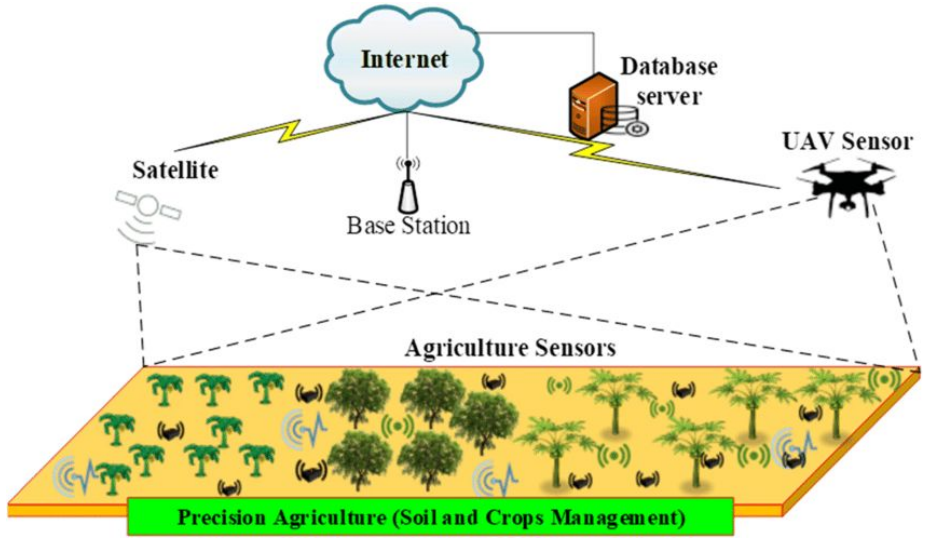
- Melhoria na **extração de características da imagem** usando mecanismos de atenção e blocos do tipo transformer.
- Criação de uma **rede que combina informações de diferentes camadas da imagem** para destacar melhor os objetos.
- Desenvolvimento de uma **função de perda (RIOU)** que ajuda a caixa delimitadora prevista a se alinhar melhor com o objeto real.

Fundamentação teórica

Sensoriamento Remoto

- Obtenção de informações sobre um alvo sem contato direto (satélites, drones, aviões).
- Desafio central: identificar objetos de interesse em imagens enormes, complexas e cheias de ruído.

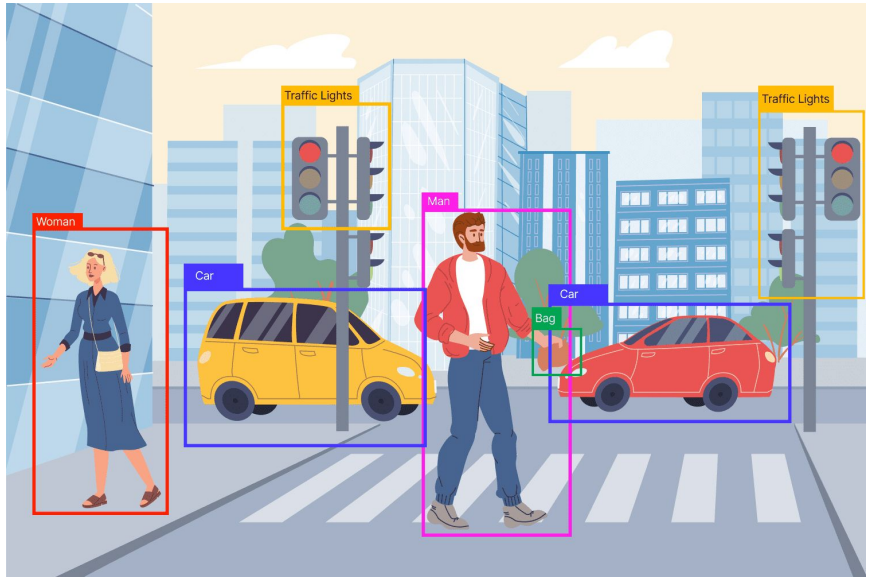
Agricultura de Precisão



Fundamentação teórica

Detecção de Objetos

- Objetivo: localizar e classificar objetos em imagens.
- Saída típica: caixas delimitadoras (bounding boxes) + rótulo da classe.
- Desafios:
  - Variação de escala (pequeno x grande).
  - Oclusão (objetos sobrepostos).
  - Ruído de fundo.



Fundamentação teórica

Detecção de Pequenos Objetos

- Definição: objetos que ocupam poucas dezenas de pixels.
- Problemas:
  - Pouca informação visual.
  - Fácil confusão com o fundo.
- Por isso, modelos padrão têm baixa precisão em objetos pequenos.

Fundamentação teórica

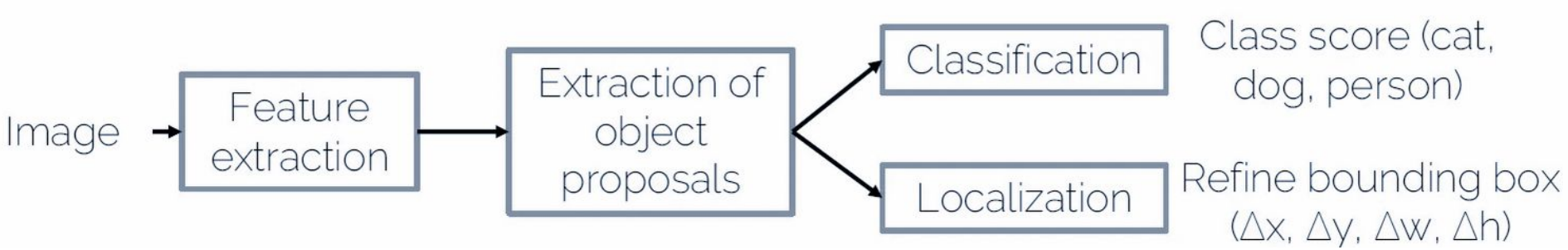
Redes Neurais Convolucionais (CNNs)

- Funcionamento:
  - Camadas convolucionais → detectam padrões (bordas, formas, texturas).
  - Camadas mais profundas → detectam objetos complexos.
- Limitação: perdem detalhes locais em objetos muito pequenos.

Fundamentação teórica

Algoritmos Clássicos

- Algoritmos de 2 estágios (R-CNN, Faster R-CNN)

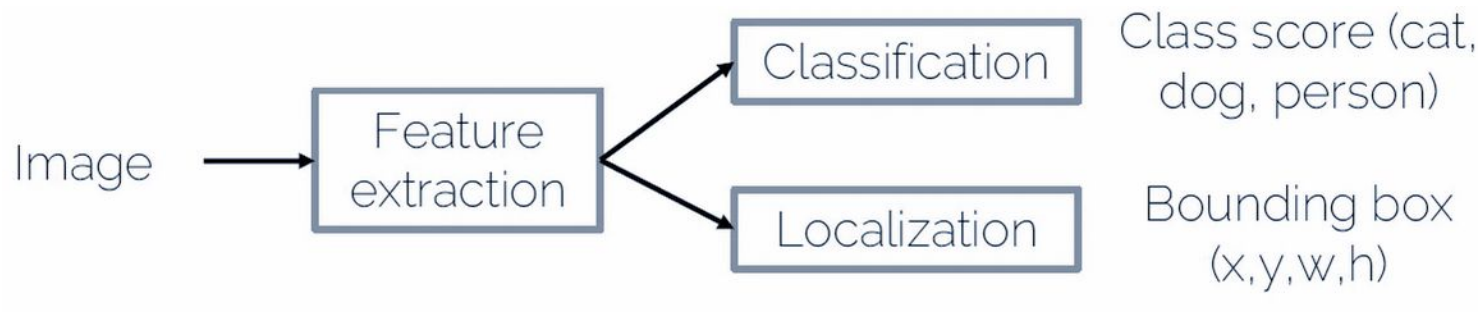


- Alta precisão, mas são lentos lentos, o que é um problema para aplicações em tempo real.

Fundamentação teórica

Algoritmos Clássicos

- Algoritmos de 1 estágio (YOLO, SSD)



- São mais rápidos, mas ainda falham em pequenos objetos.

Fundamentação teórica

Evolução do Yolo (Pequenos Objetos)

- O YOLOv1 não é eficiente para detectar objetos próximos uns dos outros ou grupos muito pequenos.
- O YOLOv2 adicionou um mecanismo de âncora que facilita a detecção de objetos pequenos.
- O YOLOv3 realiza fusão em múltiplas escalas por meio de uma estrutura de rede piramidal, o que melhora a detecção de pequenos objetos.
- O YOLOv4 melhora a detecção de alvos de diferentes tamanhos.

Fundamentação teórica

Evolução do Yolo (Pequenos Objetos)

- O YOLOv5 introduz a função de perda GIoU e o otimizador Adam, o que o torna melhor na detecção de alvos densamente ocluídos.
- O YOLOv6 e YOLOv7 apresentam melhorias e otimizações a nível de algoritmo, como estrutura da rede e estratégia de treinamento.
- O YOLOv8 apresenta uma nova arquitetura, uma nova camada convolucional e um novo detection head, além de melhorias em velocidade e precisão (detecção de objetos em tempo real).

Fundamentação teórica

Transformers em Visão Computacional

- Originalmente criados para NLP (tradução automática).
- Em visão computacional (ViT – Vision Transformer):
  - Divide a imagem em blocos (patches).
  - Cada bloco “conversa” com os outros (atenção global).
- Benefício: captura contexto global (bom para entender a cena inteira).
- Limitação: precisa ser combinado com CNNs para não perder detalhes locais.

Fundamentação teórica

Funções de Perda em Detecção de Objetos

- O que é: mede a diferença entre a caixa prevista e a caixa real.
- Exemplos:
  - IoU (Intersection over Union), GIoU (Generalized Intersection over Union), DIoU (Distance Intersection over Union), CIoU (Complete Intersection over Union).
- Problema: em alguns casos, caixas diferentes podem ter a mesma pontuação de erro, o que dificulta convergência.
- Necessidade de funções mais robustas (como a RIOU do artigo).

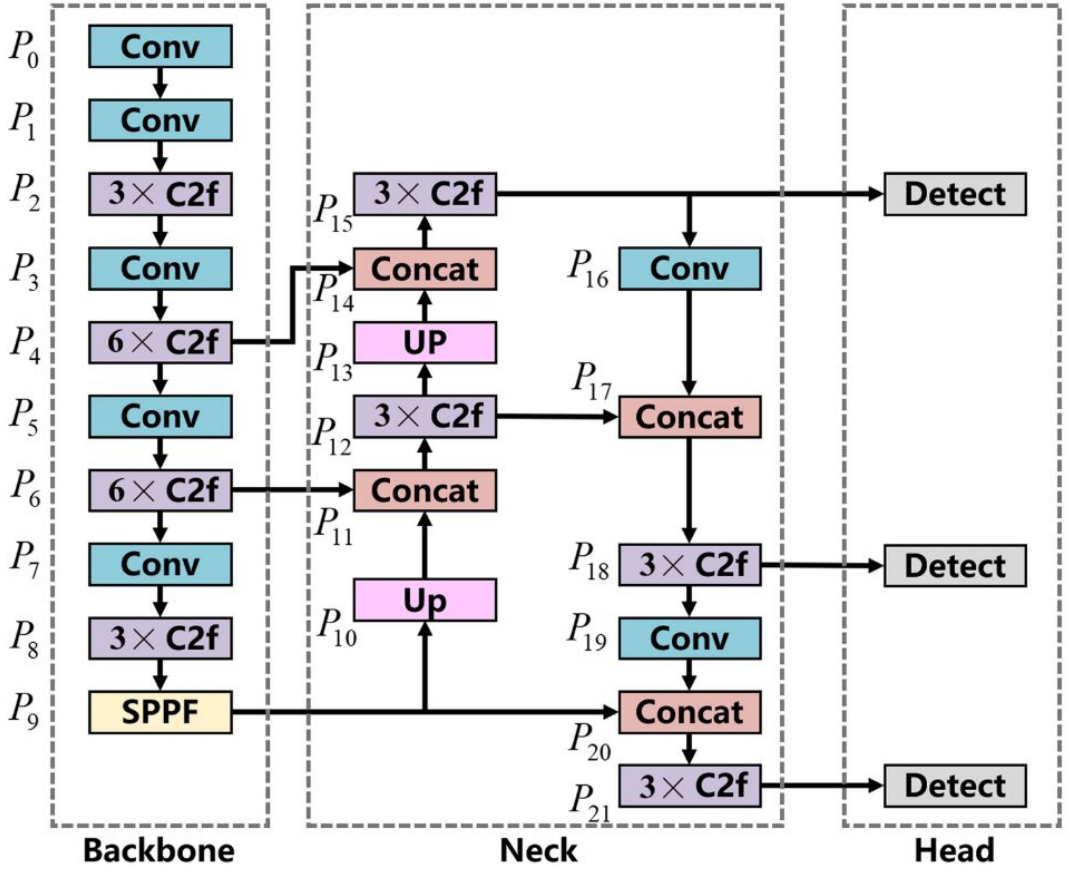
Fundamentação teórica

Arquitetura do YOLOv8

O backbone identifica partes importantes da imagem como bordas, formas e cores. Ele passa a imagem por várias camadas que filtram e destacam essas informações (ex: Conv, C2f e SPPF).

O neck organiza e combina essas informações de diferentes níveis como uma pirâmide, do detalhe fino até a visão geral, para entender o que realmente importa.

A cabeça de detecção (head) decide o que é objeto, de que tipo, e onde ele está na imagem.



O C2F tem como objetivo melhorar a extração de características.

UP (ou Upsample) tem a função de aumentar a resolução espacial do mapa de características

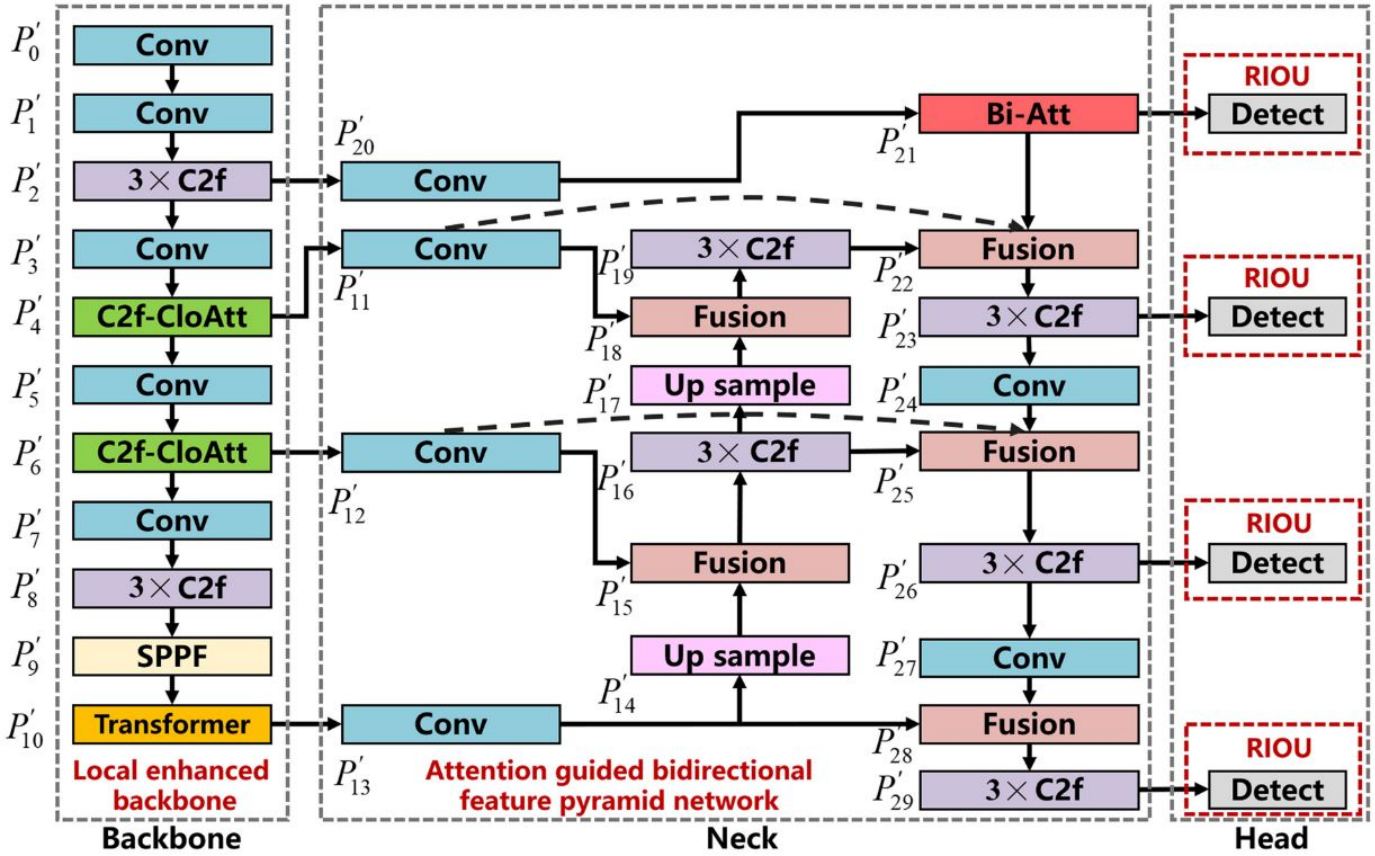
O SPPF tem como objetivo extrair informações de diferentes escalas do mapa de características.

Arquitetura e funcionamento

Visão Geral da Arquitetura do LAR-YOLOv8

No backbone do YOLOv8, a utilização do módulo C2f para extração de características nas camadas P4 e P6 pode levar à degradação das informações extraídas.

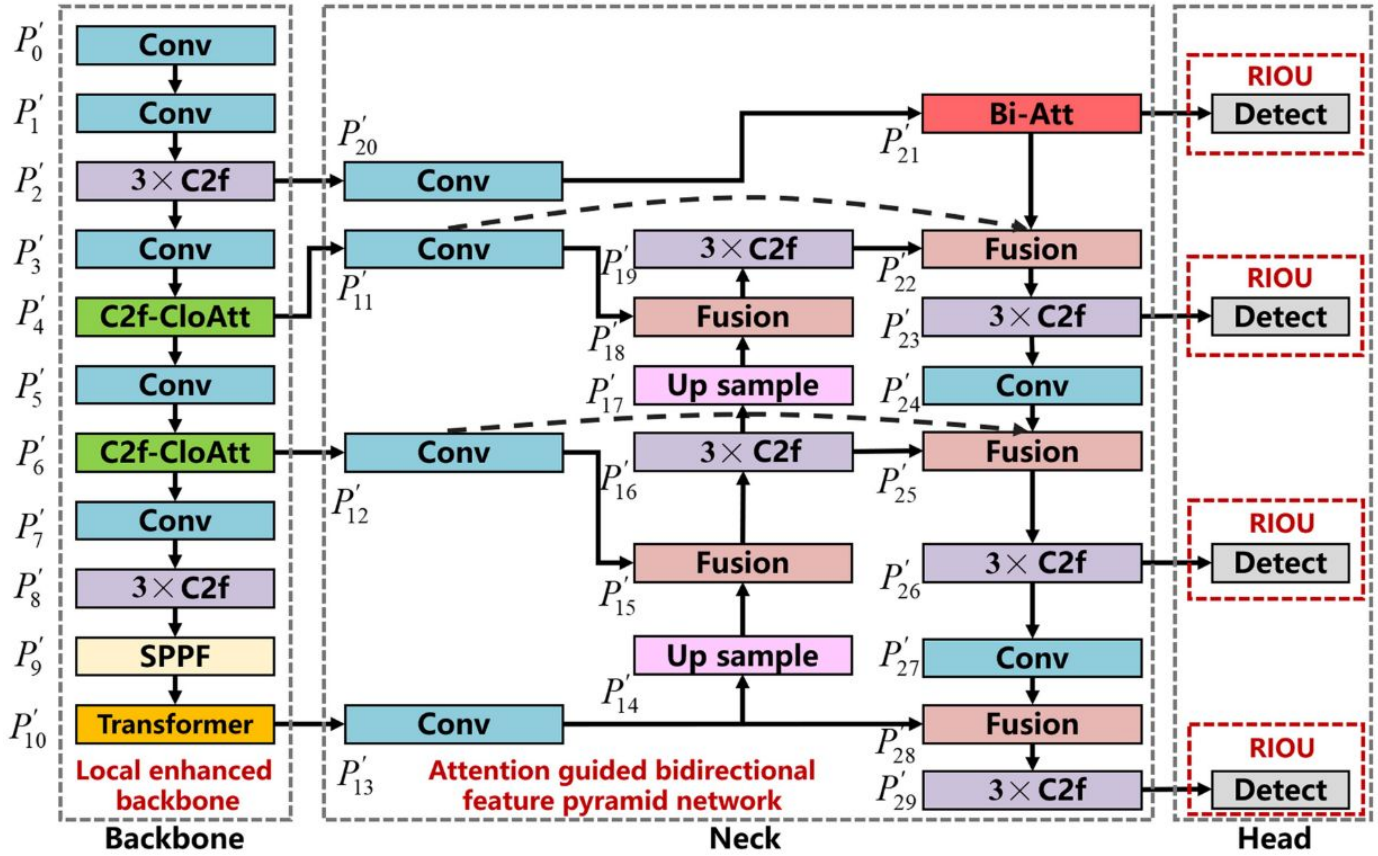
No LAR-YOLO um mecanismo de atenção de ramo duplo é utilizado para reforçar o módulo C2f, melhorando a eficiência da extração de características.



Arquitetura e funcionamento

Visão Geral da Arquitetura do LAR-YOLOv8

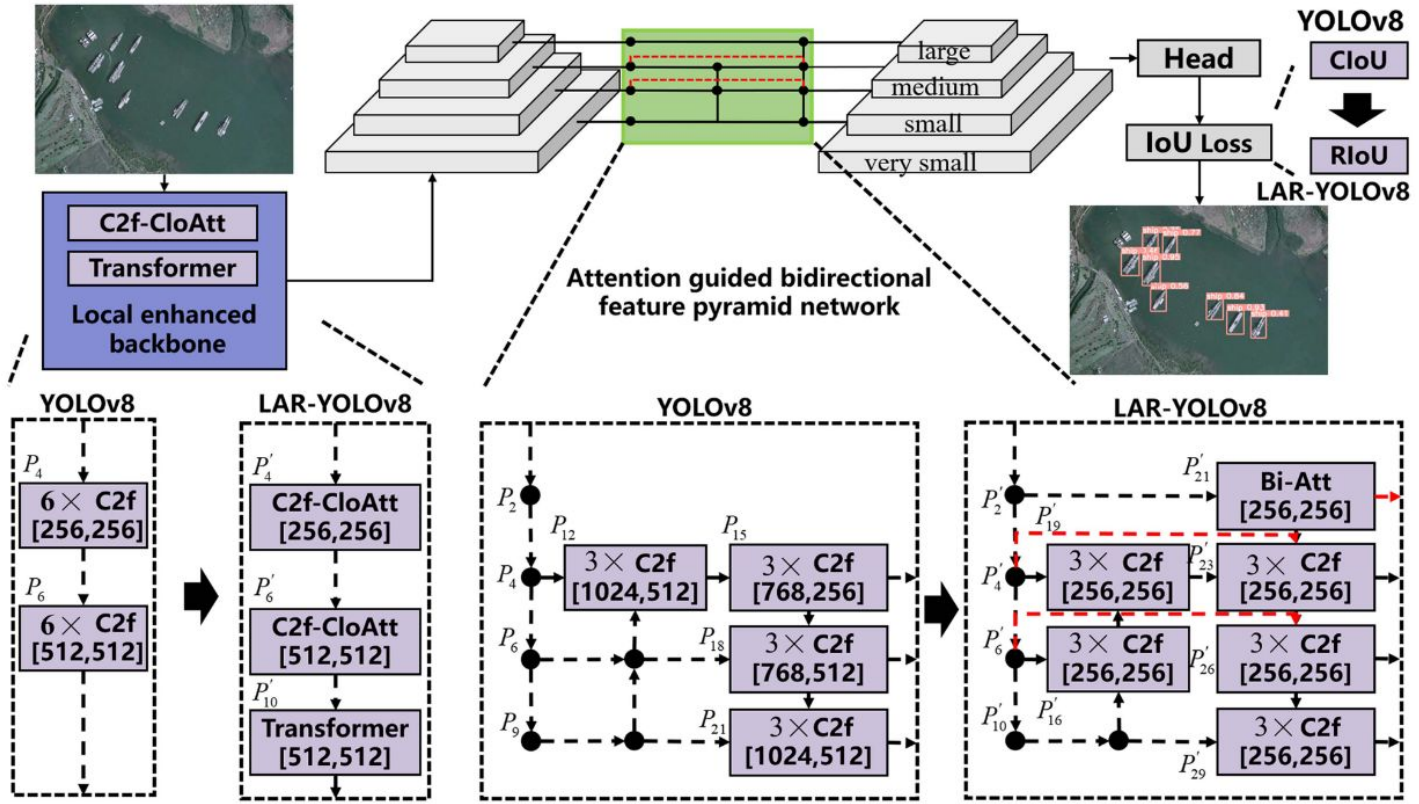
Além disso, no LAR-YOLO, um módulo vision transformer é adicionado na base do backbone para maximizar a representação do mapa de características.



Arquitetura e funcionamento

Visão Geral da Arquitetura do LAR-YOLOv8

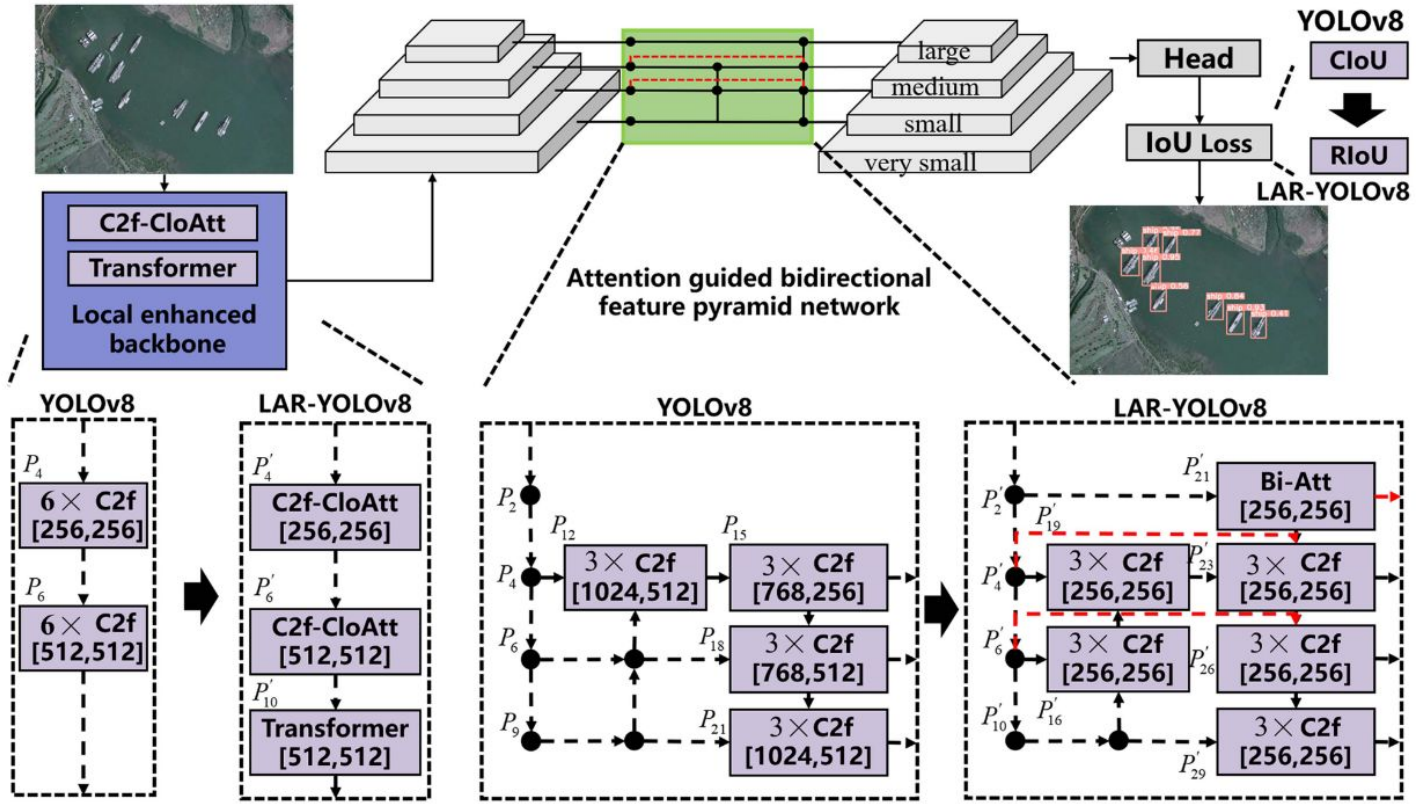
Em seguida, no neck, é projetada uma rede piramidal de características bidirecional guiada por atenção.



Arquitetura e funcionamento

Visão Geral da Arquitetura do LAR-YOLOv8

Um módulo C2f adicional é introduzido para fusão de características, e um caminho é criado para que as camadas P4 e P6 se fundam com a parte final da pirâmide, aumentando a quantidade de informações combinadas.



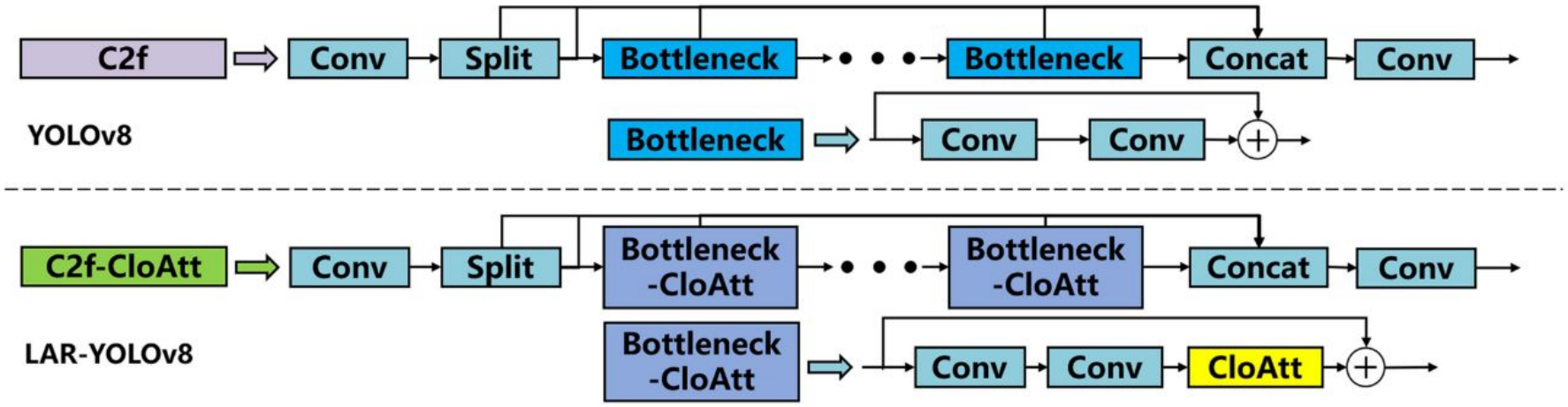
Arquitetura e funcionamento

Visão Geral da Arquitetura do LAR-YOLOv8

Como essa rede gera recursos de forma mais eficiente, o número de parâmetros do modelo é mantido, reduzindo o número de mapas de características gerados pelo módulo C2f.

Arquitetura e funcionamento

Local Enhanced Backbone



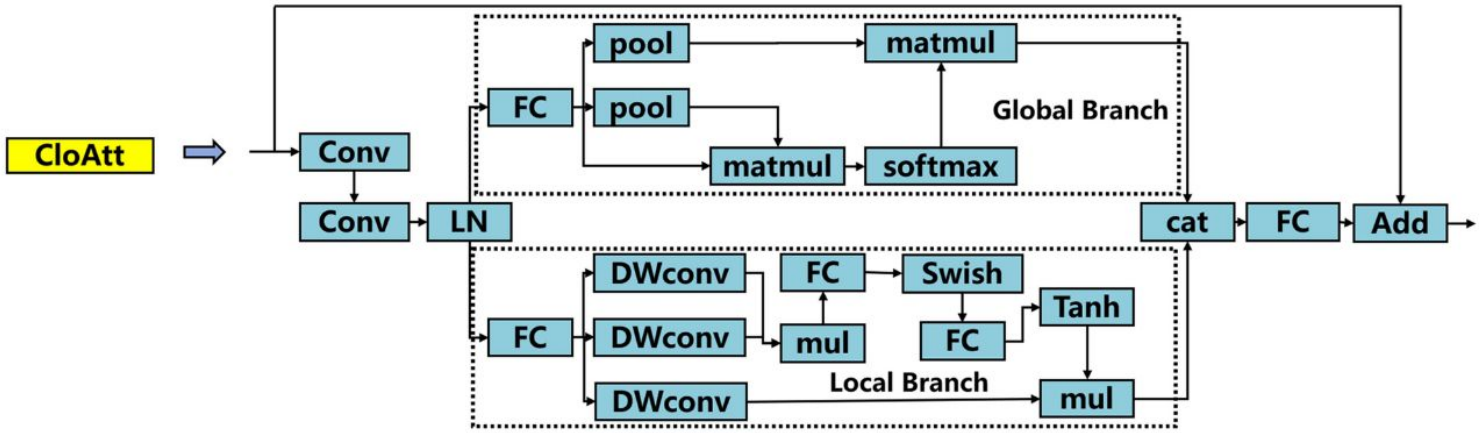
Como o backbone do YOLOv8 é apenas um empilhamento simples de módulos convolucionais, ele não consegue distinguir bem o objeto do fundo. Para resolver isso, o módulo C2f no backbone é aprimorado com o cloatt.

O bottleneck ajuda a extrair características de forma eficiente, reduzindo parâmetros e mantendo o desempenho.

Arquitetura e funcionamento

Local Enhanced Backbone

Um dos ramos captura informações locais usando convolução separável em profundidade (depth-separable convolution) e depois aprimora a características locais com pesos sensíveis ao contexto.



No outro ramo, informações são obtidas através de mecanismos de atenção. Por fim, os dois ramos são combinados para reforçar as características locais.

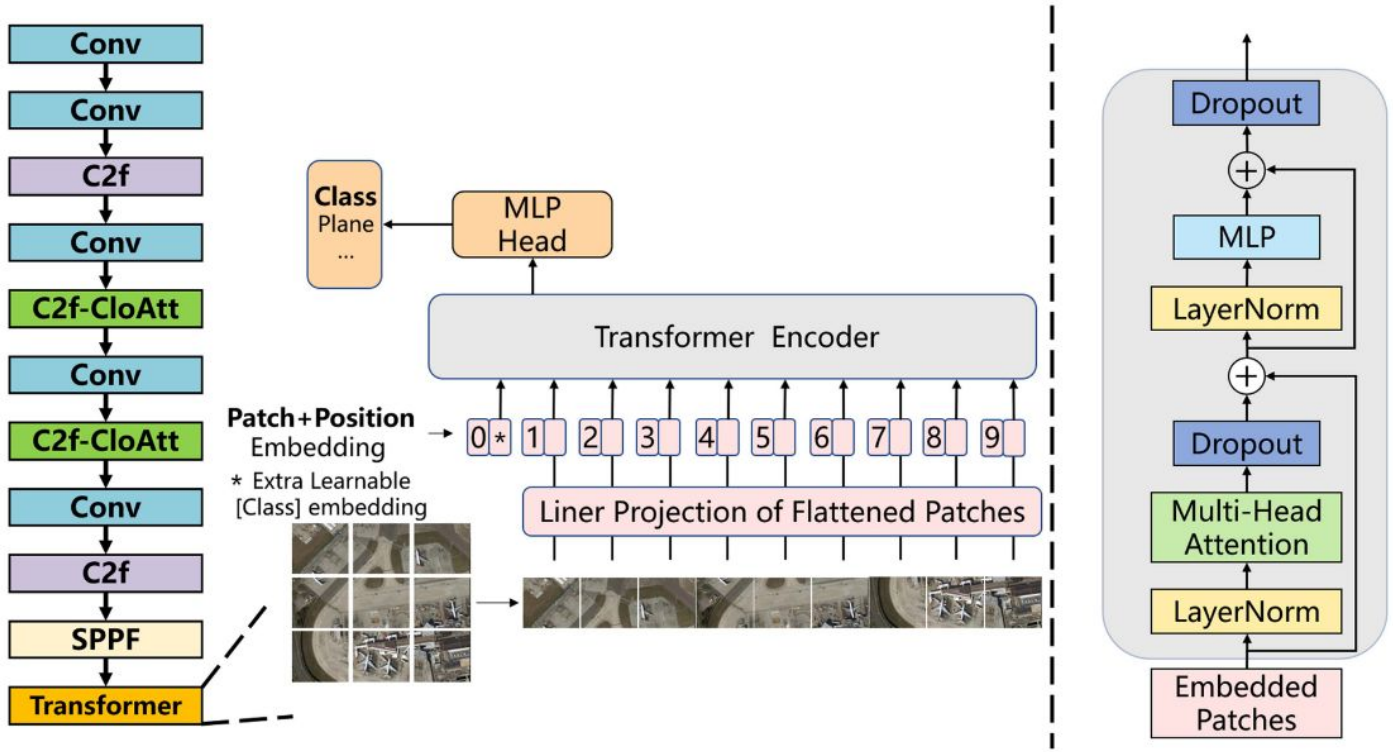
FC - Fully Connected.
Pool - usada para reduzir a dimensão espacial de mapas de características, resumindo informações locais e destacando padrões importantes.
Matmul - multiplicação de matrizes.
Softmax - transforma um vetor de valores reais em probabilidades.
Swish - função de ativação não linear.

Arquitetura e funcionamento

Local Enhanced Backbone

O bloco transformer aprimora a capacidade de representação do mapa de características.

A atenção multicanal ajuda o nó atual a não apenas focar no pixel em questão, mas também a capturar a semântica contextual ao seu redor.

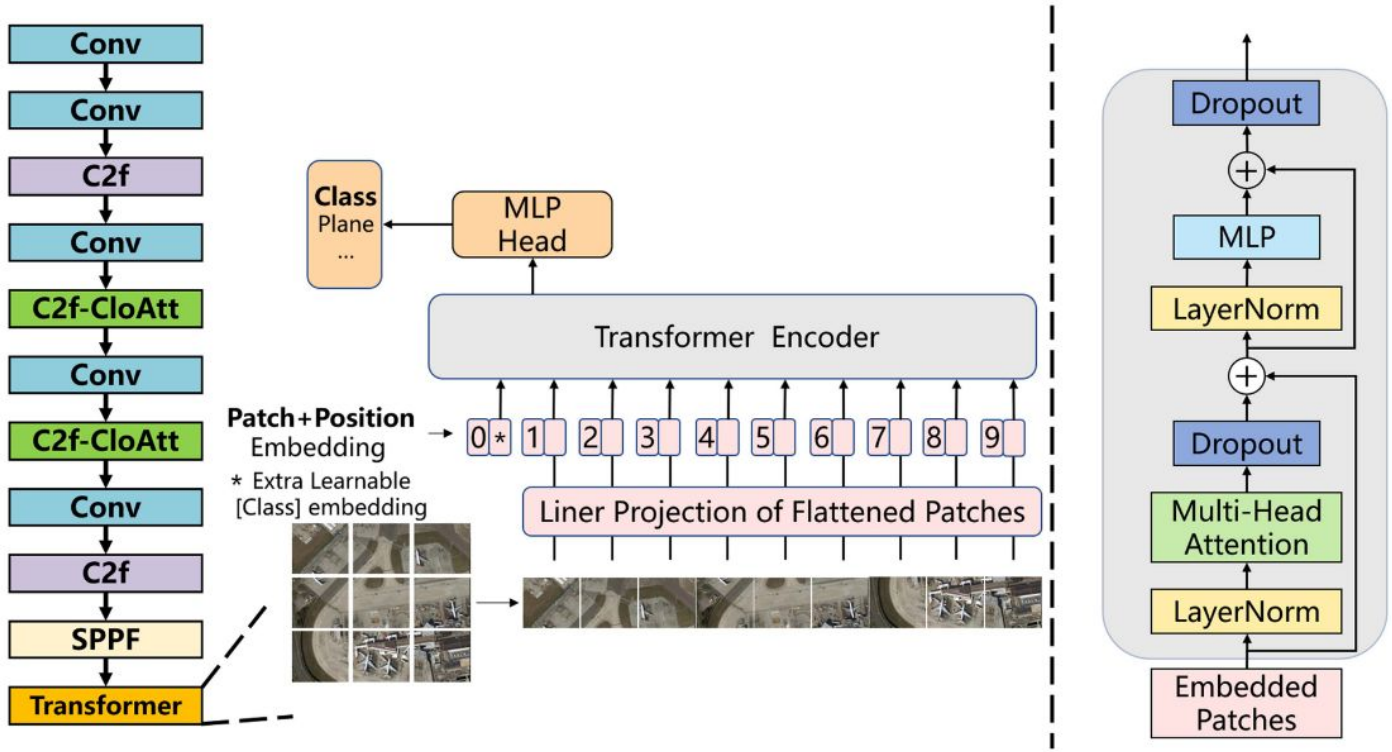


MLP - **Multilayer Perceptron** (Perceptron Multicamadas), que é um tipo de **rede neural artificial** clássica usada em aprendizado de máquina.

Arquitetura e funcionamento

Local Enhanced Backbone

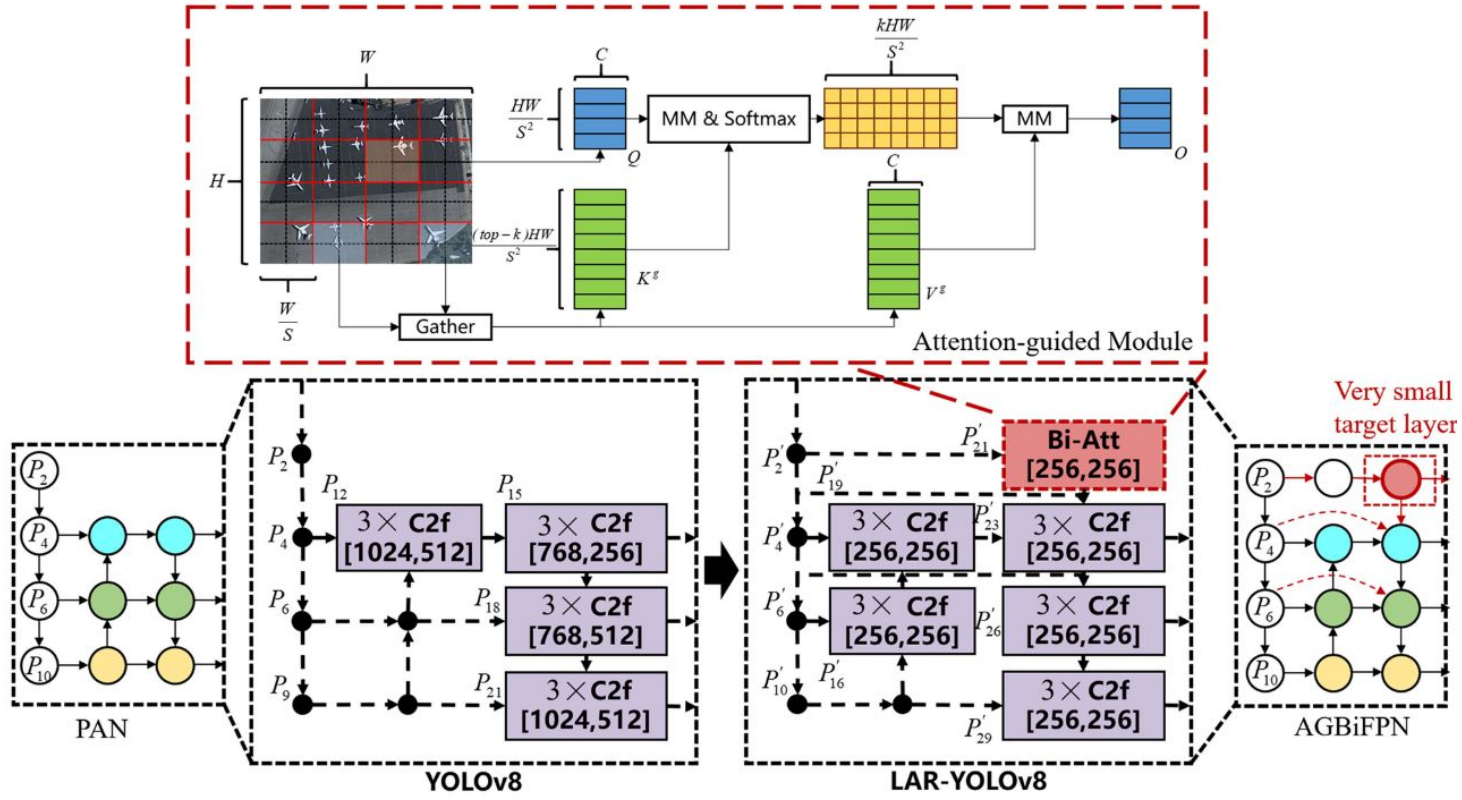
Ao mesmo tempo, como a parte final do backbone possui baixa resolução, aplicar o transformer nos mapas de características reduz os custos computacionais e de memória.



Arquitetura e funcionamento

Attention Guided Bidirectional Feature Pyramid Network

O YOLOv8 utiliza a estrutura PAN para melhorar a fusão de características por meio de uma fusão secundária, resultando em uma fusão bidirecional simples, enquanto muitas convoluções podem levar à degradação dos detalhes relacionados a alvos pequenos.

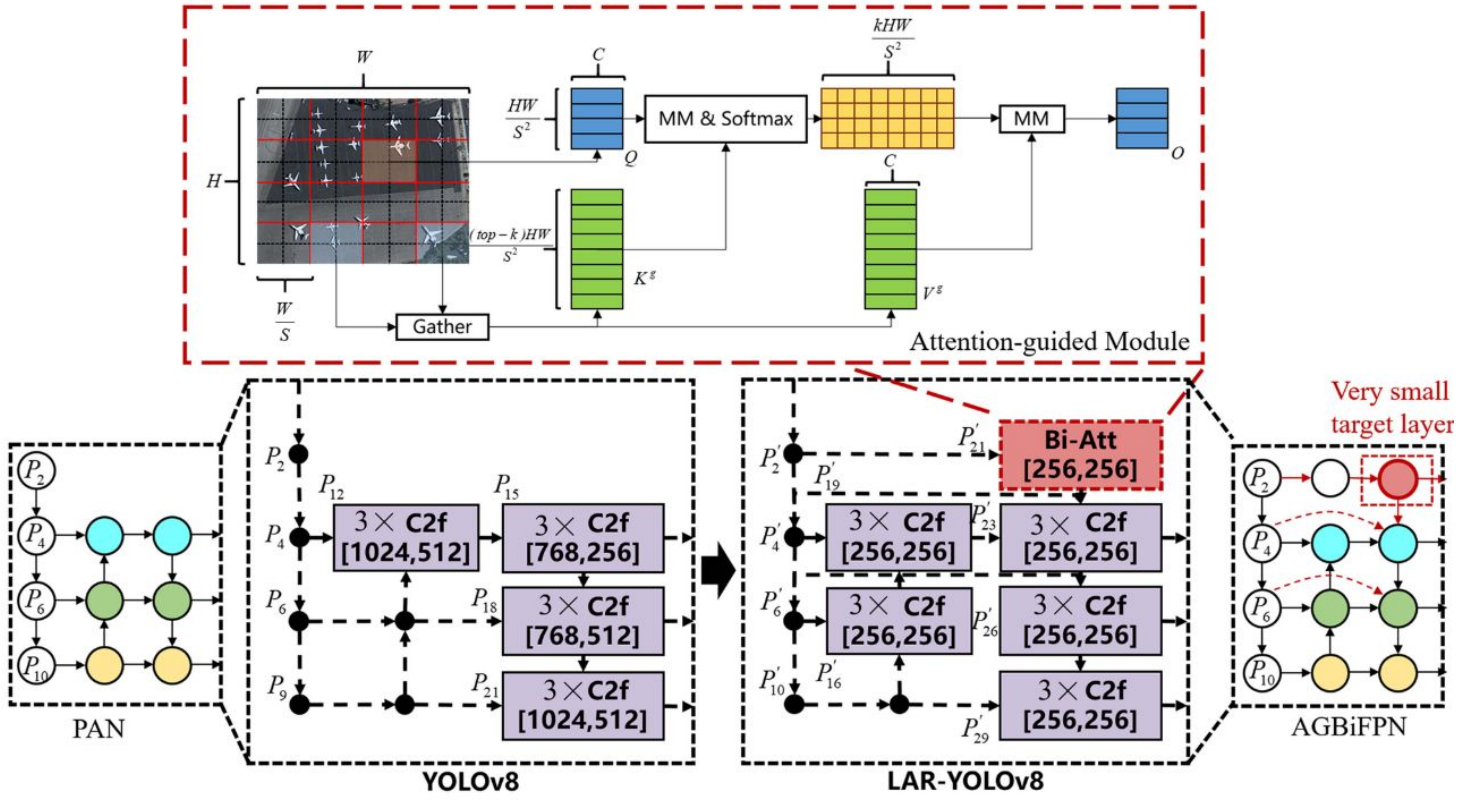


Arquitetura e funcionamento

Attention Guided Bidirectional Feature Pyramid Network

Este artigo propõe uma rede piramidal de características bidirecional guiada por atenção.

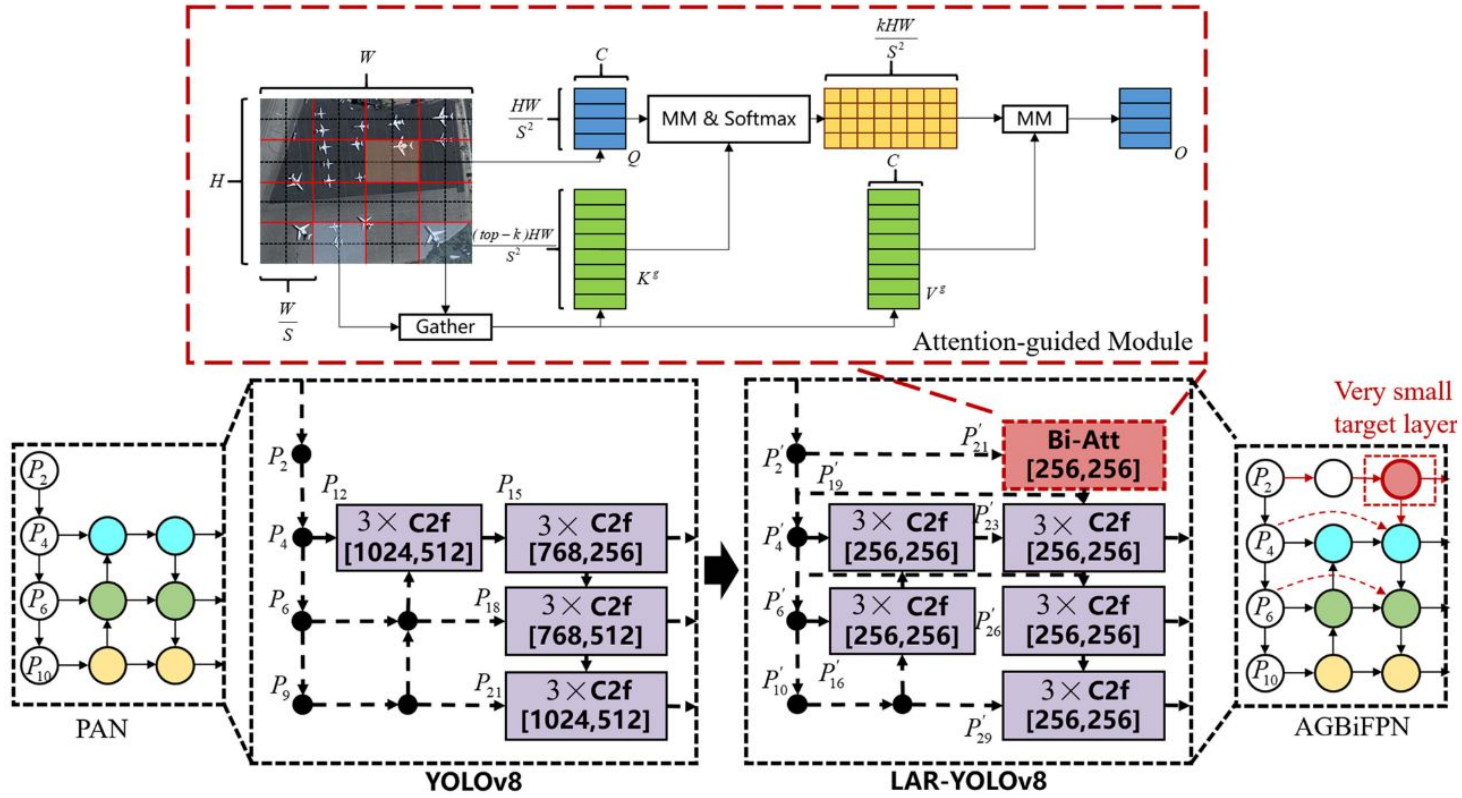
Um mecanismo de atenção esparsa dinâmica é utilizado para extrair informações da camada de alvos muito pequenos.



Arquitetura e funcionamento

Attention Guided Bidirectional Feature Pyramid Network

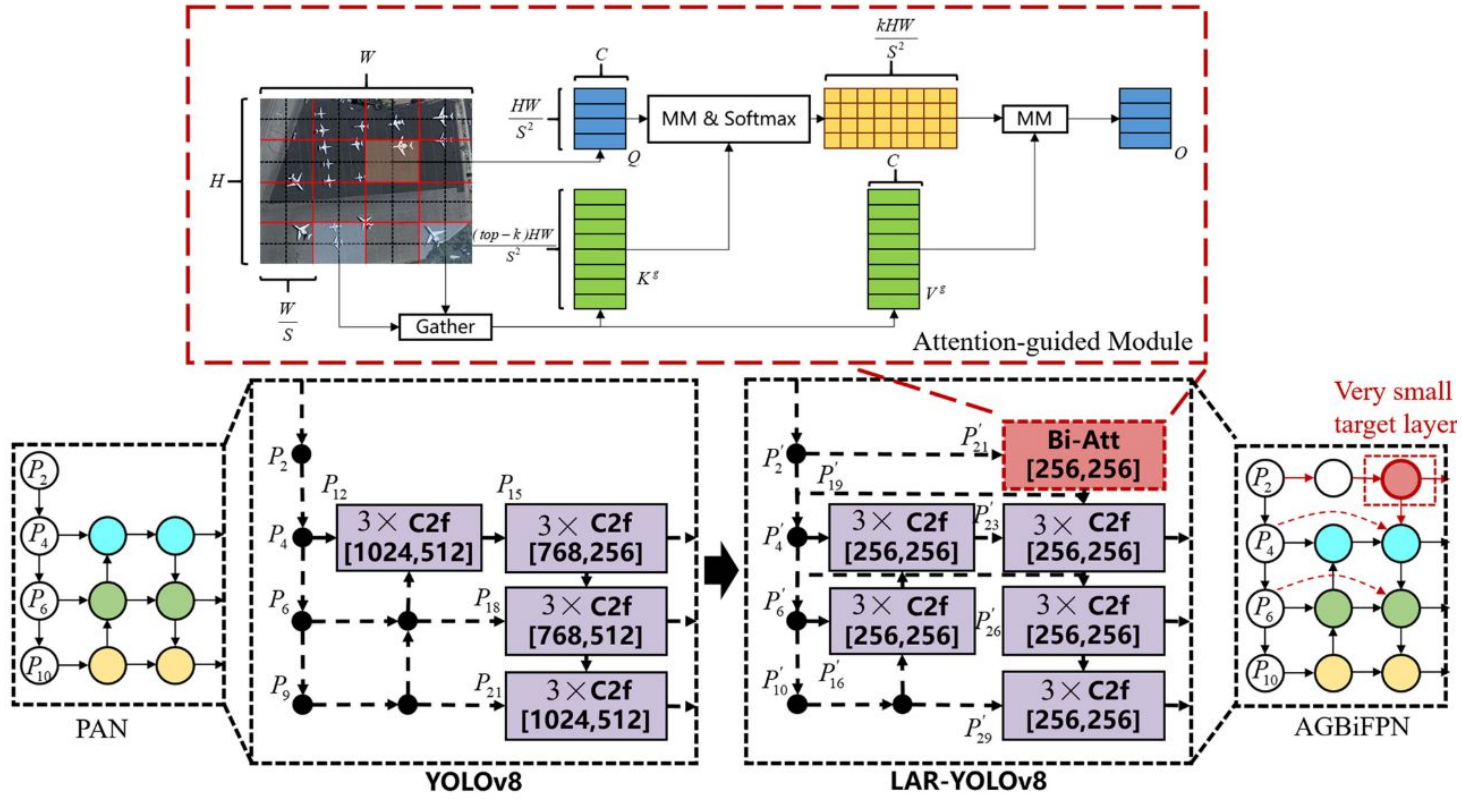
Em seguida, caminhos de cima para baixo são adicionados para transmitir informações de alto nível e guiar os módulos subsequentes na fusão de características, gerando representações mais discriminativas.



Arquitetura e funcionamento

Attention Guided Bidirectional Feature Pyramid Network

Além disso, os primeiros e últimos nós da camada intermediária de características são conectados para alcançar uma fusão ainda melhor.



Arquitetura e funcionamento

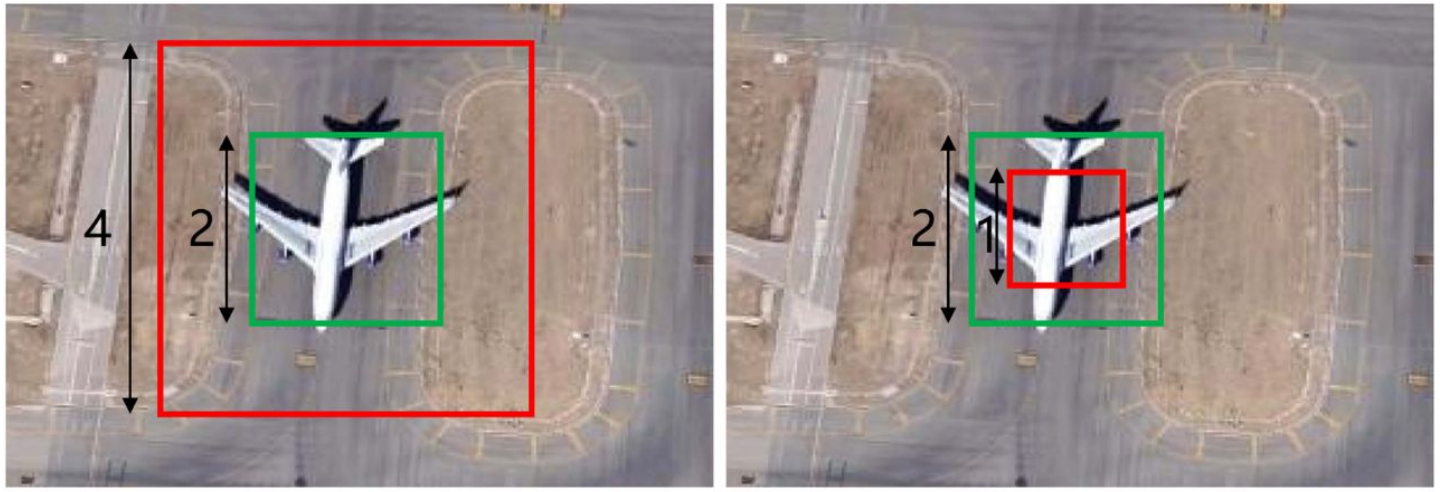
Improve Loss Function with RIOU

Para o problema de regressão de caixas delimitadoras, o YOLOv8 utiliza a função de perda CIoU. Como a CIoU define a razão de aspecto como um valor relativo, podem ocorrer situações ambíguas. Além da CIoU, outras funções de perda como GIoU, DIoU, EIoU e outras também são amplamente utilizadas para regressão de caixas delimitadoras.

Arquitetura e funcionamento

Improve Loss Function with RIOU

No entanto, quando a caixa prevista e a caixa real possuem a mesma razão de aspecto, mas valores diferentes de largura e altura, os valores da função de perda dessas regressões permanecem iguais. Isso limita a precisão da convergência do modelo.



$L_{GIOU} = 0.75, L_{DIOU} = 0.75,$
$L_{CIOU} = 0.75, L_{EIOU} = 1.25, L_{RIOU} = 0.97$

(a)

$L_{GIOU} = 0.75, L_{DIOU} = 0.75,$
$L_{CIOU} = 0.75, L_{EIOU} = 1.25, L_{RIOU} = 0.92$

(b)

Arquitetura e funcionamento

Attention Guided Bidirectional Feature Pyramid Network

Os autores desenvolveram uma função de perda chamada LRIOU. Ao introduzir a diferença logarítmica entre o comprimento e a largura da caixa prevista e da caixa real, bem como a largura e altura da menor caixa externa que as engloba, a RIOU apresenta uma convergência mais rápida e melhores resultados de localização.

Arquitetura e funcionamento

Algoritmo: RIOU (Robust IoU Loss)
Entrada:
- B_pred = (x1p, y1p, x2p, y2p) // Caixa predita
- B_gt = (x1g, y1g, x2g, y2g) // Caixa real

Saída:
- L_RIOU // Valor da perda

Passos:
1. Calcular largura e altura do menor retângulo que contém as duas caixas:
   wc = max(x2p, x2g) - min(x1p, x1g)
   hc = max(y2p, y2g) - min(y1p, y1g)

2. Calcular distância entre os centros das caixas (ρ^2_centro).

3. Calcular diferença entre larguras e entre alturas (ρ^2_w, ρ^2_h).

4. Calcular áreas:
   A_gt = área da caixa real
   A_pred = área da caixa predita

5. Calcular interseção I entre as duas caixas.
   Se não houver sobreposição, I = 0.

6. Calcular IoU:
   IoU = I / (A_gt + A_pred - I)

7. Calcular perda RIOU:
   L_RIOU = 1 - IoU
   + (ρ^2_centro / (wc^2 + hc^2))
   + log(1 + ρ^2_w) / [5 * log(1 + wc^2)]
   + log(1 + ρ^2_h) / [5 * log(1 + hc^2)]

Treinamento

Datasets

- O dataset NWPU VHR-10 é um conjunto de dados de sensoriamento remoto, coletado do Google Earth, composto por dez categorias, incluindo 800 imagens.
- O dataset RSOD contém 936 imagens, incluindo 4993 instâncias da categoria “aeronaves”, 191 instâncias da categoria “playgrounds”, 1180 instâncias da categoria “viadutos” e 1586 instâncias da categoria “tanques de óleo” obtidas por sensoriamento remoto.
- O dataset CARPK é uma coleção de 1448 fotos de quatro diferentes estacionamentos, contendo aproximadamente 90.000 carros, coletadas por drone (PHANTOM 3 PROFESSIONAL).

Treinamento

Métricas de Avaliação

- Precisão (P), Recall (R), F1-score.
- mAP @0.5.
- mAP (small) → foco em pequenos objetos.
- Velocidade (FPS).

Treinamento

Resultados no NWPU VHR-10

Model	Class	D00	D01	D02	D03	D04	D05	D06	D07	D08	D09	Avg
YOLOv8	P	0.905	0.982	0.925	0.938	0.914	0.894	0.683	0.981	0.959	0.906	0.905
YOLOv8	R	0.807	0.944	0.829	0.923	0.973	0.706	0.417	1	0.797	0.647	0.807
YOLOv8	F1	0.853	0.963	0.874	0.930	0.943	0.789	0.518	0.990	0.871	0.755	0.848
YOLOv8	mAP	0.864	0.99	0.865	0.953	0.969	0.852	0.545	0.995	0.908	0.679	0.864
LAR-YOLOv8	P	0.959	0.934	0.916	0.92	0.951	0.929	0.975	0.981	0.93	0.919	0.942
LAR-YOLOv8	R	0.989	0.809	0.908	0.934	0.763	0.729	0.963	0.681	0.765	0.772	0.831
LAR-YOLOv8	F1	0.974	0.867	0.912	0.927	0.847	0.817	0.969	0.804	0.839	0.839	0.879
LAR-YOLOv8	mAP	0.993	0.907	0.89	0.966	0.878	0.858	0.994	0.861	0.848	0.886	0.908

Treinamento

Resultados no RSOD

Model	Class	Aircraft	Oiltank	Overpass	Playground	Avg
YOLOv8	P	0.965	0.978	0.719	0.951	0.903
YOLOv8	R	0.913	0.944	0.711	0.968	0.884
YOLOv8	F1	0.938	0.961	0.715	0.959	0.893
YOLOv8	mAP	0.953	0.977	0.678	0.98	0.897
LAR-YOLOv8	P	0.967	0.964	0.804	0.925	0.915
LAR-YOLOv8	R	0.899	0.918	0.778	0.968	0.891
LAR-YOLOv8	F1	0.932	0.940	0.791	0.946	0.902
LAR-YOLOv8	mAP	0.948	0.982	0.854	0.989	0.943

Resultados no CARPK

Model	P	R	F1	mAP
YOLOv8	0.953	0.896	0.923	0.957
LAR-YOLOv8	0.981	0.956	0.968	0.984

Vantagens

- Alta precisão em objetos pequenos (mAP(small) ↑).
- Backbone mais eficiente (menos parâmetros, –40%).
- Melhor generalização em cenários complexos (ruído, fundo denso).
- Mais interpretável ⟶ Grad-CAM mostra foco nos alvos.
- Leve para deploy ⟶ redução de FLOPs e tamanho do modelo.

Desvantagens/Limitações

- Velocidade menor (FPS ↓) comparado ao YOLOv8 puro (mais camadas).
- Maior complexidade arquitetural → difícil de treinar/implementar do zero.
- Requer hardware relativamente potente para treinamento (RTX 3080).
- Pouco testado fora de datasets específicos (NWPU VHR-10, RSOD, CARPK).
- Futuro necessário: compressão do modelo e adaptação multimodal.

Comparação com outros algoritmos

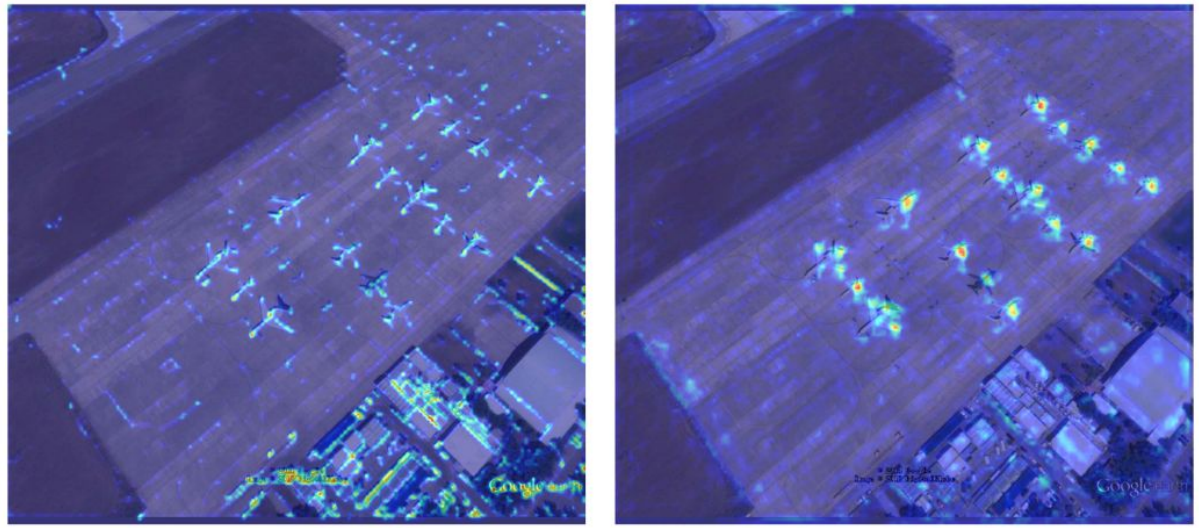
Resultados no NWPU VHR-10

- LAR-YOLOv8 supera YOLOv8 em:
  - P: +3.7%
  - R: +2.4%
  - mAP: +4.4%
- Grande melhoria em categorias difíceis (ex: Ground track field: +44.9%).

Comparação com outros algoritmos

Resultados no RSOD

- mAP: +4.6% sobre YOLOv8.
- Categoria “Overpass”: salto de 67.8% → 85.4%.

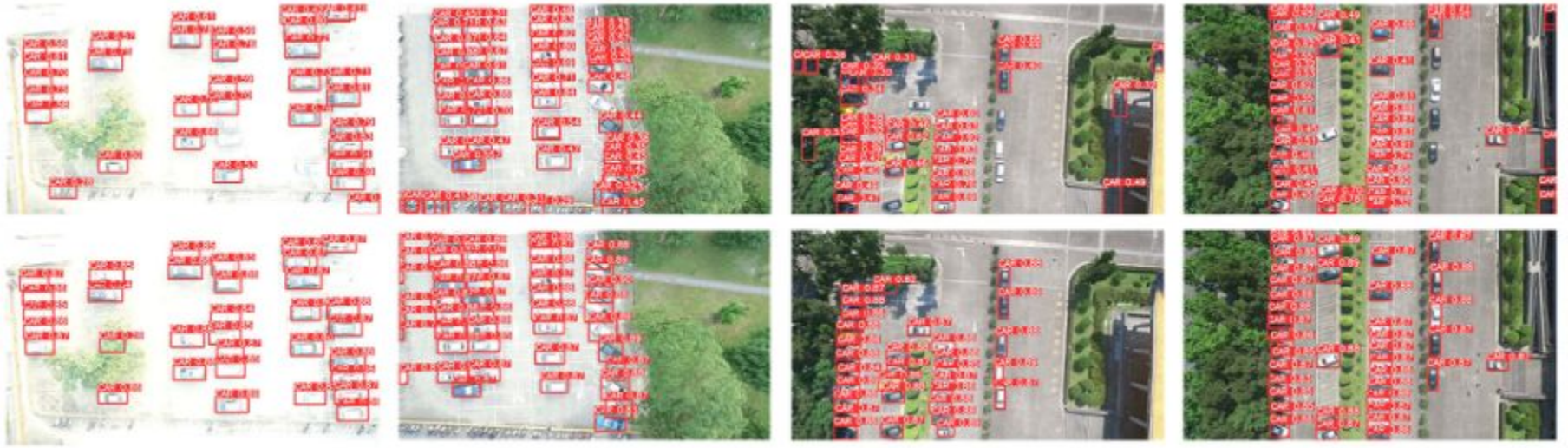


YOLOv8 LAR-YOLOv8

Comparação com outros algoritmos

Resultados no CARPK

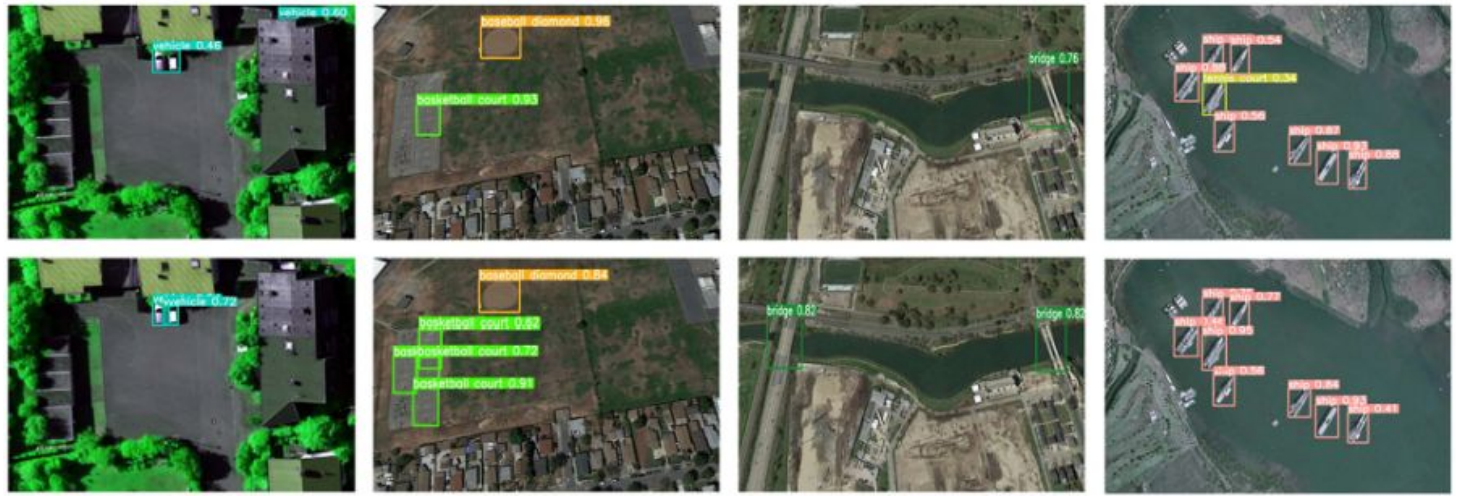
- mAP: +2.7% sobre YOLOv8.
- Melhora significativa em cenários densos (estacionamentos).



Comparação com outros algoritmos

Visualizações

- Grad-CAM ⟶ LAR-YOLOv8 foca mais nos alvos, menos em fundo.
- Exemplos de detecção: YOLOv8 falha em pequenos objetos, LAR-YOLOv8 corrige.



Comparação com outros algoritmos

Ablation Study

- Apenas backbone aprimorado → +1.8% mAP.
  - AGBiFPN → +1.4% mAP.
  - RIOU → +1.2% mAP.
- Total: +4.4% mAP.
- Redução de 40% dos parâmetros e 18% FLOPs.

Comparação com outros algoritmos

Discussão

- Pontos fortes:
  - Melhor detecção de alvos pequenos.
  - Mais leve (menos parâmetros).
  - Melhor interpretabilidade (Grad-CAM).
- Limitação:
  - FPS menor devido ao aumento de camadas.
- Futuro:
  - Compressão de modelo (distillation).
  - Multisource data + domain adaptation.

Conclusões

- Proposto: LAR-YOLOv8.
- Melhorias em:
  - Extração de características.
  - Fusão de recursos.
  - Função de perda.
- Resultados:
  - +4.4% mAP no NWPU VHR-10.
  - Melhor mAP (small).
  - Redução de 40% nos parâmetros.
- Conclusão: modelo eficaz para detecção de pequenos objetos em sensoriamento remoto.

Perguntas?

Quiz e Experimento Prático

Quiz

https://forms.gle/fYE4NFDSGCbcyE7C6

Github

https://github.com/jonasvm/seminario-lar-yolo

Referências

H. Yi, B. Liu, B. Zhao and E. Liu, "Small Object Detection Algorithm Based on Improved YOLOv8 for Remote Sensing," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp. 1734-1747, 2024, doi: 10.1109/JSTARS.2023.3339235.
keywords: {Feature extraction;Remote sensing;Transformers;Deep learning;Real-time systems;Optical sensors;Optical imaging;Attention mechanism;detection algorithms;remote sensing;vision transformer;YOLOv8},

Obrigado!