CAPÍTULO 4

AVALIAÇÃO DE MODELOS YOLO

A avaliação da acurácia e do custo computacional orienta o desenvolvimento de detectores de objetos. Ao contrário da classificação de imagens, em que se verifica apenas se a categoria predita coincide com a real, a detecção exige avaliar conjuntamente a posição espacial das caixas delimitadoras (*bounding boxes*) e a categoria de cada alvo (PADILLA; NETTO; SILVA, 2020). Em modelos YOLO, essas métricas guiam o ajuste de hiperparâmetros e a escolha do modelo para uso em campo, seja em servidores ou em dispositivos de borda.

1. MATRIZ DE CONFUSÃO E DIAGNÓSTICO DE ERROS

A matriz de confusão cruza, para cada categoria, a classe real anotada com a classe predita pelo modelo. Dela derivam-se três quantidades básicas:

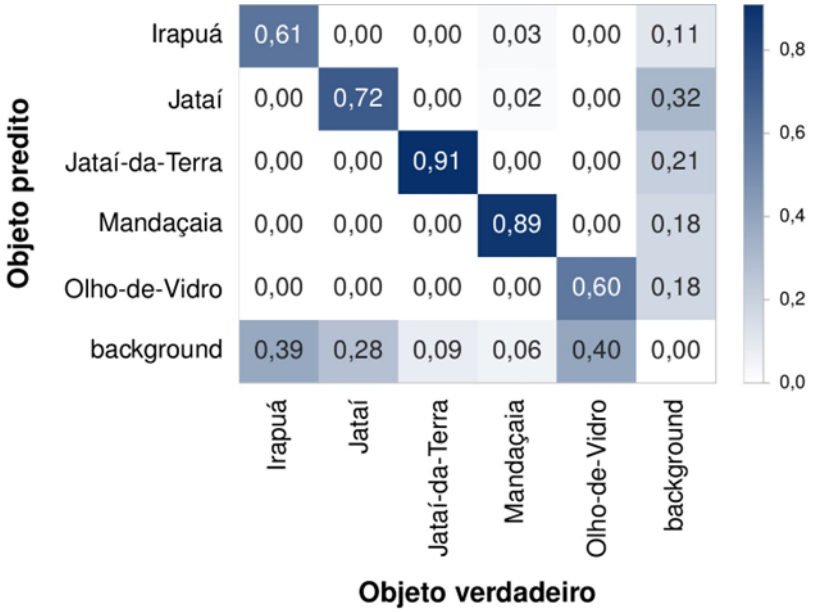
- **Verdadeiro Positivo (VP/TP):** o objeto existe e foi detectado e classificado corretamente;
- **Falso Positivo (FP):** o modelo detectou um objeto que não existe (confundiu o fundo com um alvo) ou atribuiu a classe errada a ele;
- **Falso Negativo (FN):** o objeto existe na imagem, mas o modelo não o detectou.

Em detecção de objetos, a matriz produzida pelo framework Ultralytics possui uma linha e uma coluna adicionais denominadas *background* (fundo), que registram, respectivamente, os objetos reais que o modelo deixou de detectar (FN) e as detecções feitas sobre regiões que não continham nenhum objeto (FP).

1.1 Exemplo: Detecção de Espécies de Abelhas Nativas

A Figura 14 apresenta a matriz de confusão normalizada obtida por França et al. (2025) ao treinar a arquitetura YOLOv8 para detectar diferentes espécies de abelhas nativas do Cerrado brasileiro: Irapuá, Jataí, Jataí-da-Terra, Mandaçaia e Olho-de-Vidro. As colunas correspondem à espécie real e as linhas, à espécie predita pelo modelo; cada valor indica a proporção dos exemplares de uma espécie real que recebeu determinada predição.

Figura 14: Matriz de confusão normalizada da detecção de cinco espécies de abelhas nativas do Cerrado brasileiro com YOLOv8. Fonte: França et al. (2025). A leitura da matriz começa pela diagonal principal, onde a classe predita coincide com a classe real (acertos VP). Células fora da diagonal indicam confusões entre espécies ou com o fundo.

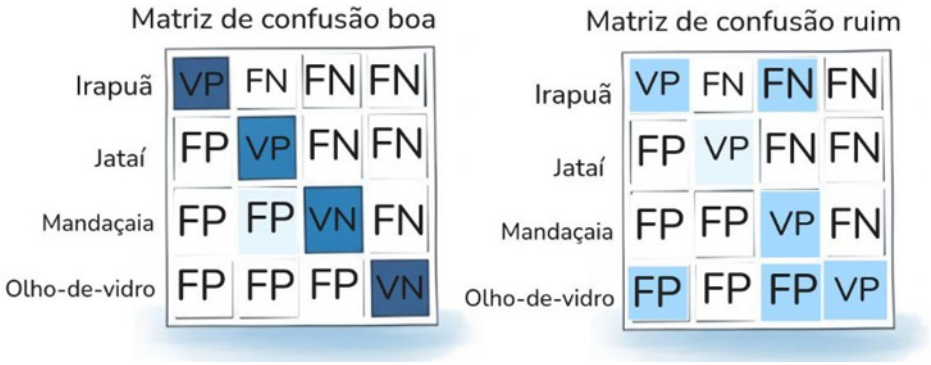


No exemplo das abelhas, observa-se que:

- As espécies Jataí-da-Terra (0,91) e Mandaçaia (0,89) apresentam as células mais escuras da diagonal, sendo as mais bem reconhecidas pelo modelo;
- Praticamente não há confusão entre espécies: as células fora da diagonal entre as abelhas são quase todas iguais a 0,00, com exceção de pequenas trocas envolvendo a Mandaçaia (0,06 predita como Irapuá e 0,02 como Jataí);
- O principal erro do modelo está na linha *background*: 39% dos indivíduos de Irapuá e 40% dos de Olho-de-Vidro não foram detectados (Falsos Negativos), o que explica os valores mais baixos da diagonal para essas espécies (0,61 e 0,60);
- A coluna *background* mostra como se distribuem os Falsos Positivos, isto é, regiões de fundo erroneamente detectadas como abelhas, sendo a Jataí a classe com maior ocorrência indevida (0,32). O diagnóstico mostra que a dificuldade principal reside em separar as abelhas do fundo, e não em distinguir as espécies entre si. Essa constatação orienta a ampliação da base com mais amostras de fundo e de exemplares de Irapuá e Olho-de-Vidro.

A Figura 15 ilustra comparativamente a estrutura de uma matriz de confusão com elevado desempenho discriminativo em contraposição a uma matriz com degradação de performance.

Figura 15: Comparação entre uma matriz de confusão satisfatória (diagonal principal dominante com altos índices de Verdadeiros Positivos) e uma matriz de confusão ruidosa (com dispersão de Falsos Positivos e Falsos Negativos entre as classes de abelhas nativas).



Fonte: Adaptado de França et al. (2025).

Na matriz de confusão ideal (à esquerda), a maior parte das predições concentra-se na diagonal principal, onde a classe prevista coincide com a classe real (Verdadeiros Positivos, VP). Em contrapartida, em uma matriz com desempenho insatisfatório (à direita), observa-se a dispersão das detecções fora da diagonal, gerando elevados Falsos Positivos (FP) por erro de classificação cruzada e Falsos Negativos (FN) quando a rede omite a presença de indivíduos na cena visual.

2. MÉTRICAS FUNDAMENTAIS DE AVALIAÇÃO

A partir das quantidades VP, FP e FN extraídas da matriz de confusão, a comunidade científica adota um conjunto padronizado de métricas estatísticas e computacionais para quantificar a qualidade das detecções produzidas pelas arquiteturas YOLO:

1. **Interseção sobre União (*Intersection over Union*, IoU):** Medida geométrica que quantifica o grau de sobreposição entre a caixa delimitadora prevista pelo modelo (B_p) e a caixa de gabarito real (B_g). O IoU é definido matematicamente pela razão entre a área de interseção e a área de união das duas regiões:

$$\text{IoU} = \frac{\text{Área}(B_p \cap B_g)}{\text{Área}(B_p \cup B_g)}.$$

Um limiar de IoU (geralmente IoU ≥ 0,50) é estabelecido para determinar se uma detecção é considerada um Verdadeiro Positivo (TP) ou um Falso Positivo (FP).

2. **Precisão (*Precision*) e Revocação (*Recall*):**
   - **Precisão (*P*):** Quantifica a fração de detecções corretas em relação ao total de predições realizadas pelo modelo:

$$P = \frac{TP}{TP + FP}$$

- **Revocação (*R*):** Quantifica a fração de objetos reais presentes na imagem que foram identificados com sucesso pela rede:

$$R = \frac{TP}{TP + FN}$$

3. **Precisão Média (*Mean Average Precision* — mAP):** É a métrica consolidada para resumir o desempenho global de detecção. O mAP representa a média da área sob a curva de Precisão-Revocação (*Precision-Recall Curve*) calculada para todas as categorias de objetos:

   - **mAP@0.50:** Avalia a precisão média adotando um limiar fixo de IoU = 0,50, métrica tradicional herdada do benchmark PASCAL VOC (EVERINGHAM et al., 2010).
   - **mAP@0.50:0.95:** Métrica padrão do benchmark MS COCO, calculada pela média do mAP obtido em 10 limiares consecutivos de IoU (variando de 0,50 a 0,95 com incremento de 0,05), exigindo alta precisão de alinhamento espacial das caixas delimitadoras.

4. **Métricas de Eficiência Operacional e Tempo Real:**

   - **Quadros por Segundo (*Frames Per Second* — FPS):** Velocidade de processamento contínuo expressa pela quantidade de imagens analisadas por segundo.
   - **Latência de Inferência (ms):** Tempo total exigido para o processamento de uma única imagem (incluindo pré-processamento, passagem direta pela rede e pós-processamento).
   - **Parâmetros (M) e FLOPs (G):** Quantidade de parâmetros treináveis da arquitetura e contagem de operações de ponto flutuante, refletindo a pegada de memória e a demanda computacional do modelo.

No exemplo das abelhas da Figura 14, os valores da diagonal correspondem diretamente à revocação de cada espécie: o modelo recupera 91% dos exemplares de Jataí-da-Terra, mas apenas 60% dos de Olho-de-Vidro, evidenciando como a matriz de confusão e as métricas se complementam na avaliação do modelo.

3. ESTRUTURA E ARTEFATOS DO DIRETÓRIO DE SAÍDA DO TREINAMENTO

Ao executar a rotina de treinamento de um modelo YOLO utilizando o framework Ultralytics (seja via interface de linha de comando ou pela API Python *model.train()*), o sistema gera automaticamente uma estrutura de diretórios padronizada e altamente organizada dentro do caminho configurado (por padrão *runs/detect/train/*). Esse repositório de saída contém o histórico completo da execução, englobando desde os arquivos de configuração de hiperparâmetros até os gráficos de convergência, matrizes de confusão, amostras visualmente anotadas de batches de treino/validação e os pesos binários otimizados da rede neural.

A Figura 16 ilustra esquematicamente a topologia típica de um diretório de treinamento do YOLOv8/YOLOv12, destacando a organização hierárquica dos arquivos e subdiretórios gerados durante uma sessão de treinamento:

```
runs/detect/train/
+--- args.yaml (Hiperparametros e configuracoes da sessao)
+--- BoxF1_curve.png (Curva F1-Score em funcao da Confianca)
+--- BoxP_curve.png (Curva de Precisao em funcao da Confianca)
+--- BoxPR_curve.png (Curva de Precisao vs. Revocacao / mAP)
+--- BoxR_curve.png (Curva de Revocacao em funcao da Confianca)
+--- confusion_matrix.png (Matriz de confusao absoluta em contagem)
+--- confusion_matrix_normalized.png (Matriz de confusao normalizada por classe)
+--- labels.jpg (Analise estatistica das anotacoes)
+--- results.csv (Tabela com perdas e metricas por epoca)
+--- results.png (Painel grafico de evolucao das perdas)
+--- train_batch0.jpg ... (Amostras de imagens de treino com mosaic)
+--- val_batch0_labels.jpg (Anotacoes reais do lote de validacao)
+--- val_batch0_pred.jpg (Predicoes geradas no lote de validacao)
+--- weights/
      +--- best.pt (Checkpoint com melhor metrica fitness)
      +--- last.pt (Checkpoint final da ultima epoca)
```

Figura 16: Representação hierárquica dos arquivos e artefatos de diagnóstico produzidos ao final de um treinamento YOLO.

Estrutura de Arquivos Gerada pelo Treinamento do YOLO (*runs/detect/train/*)

```
.
+--- args.yaml  (Hiperparâmetros e configurações da execução)
+--- BoxF1_curve.png  (Curva F1-Score em função da Confiança)
+--- BoxP_curve.png ..... (Curva de Precisão em função da
Confiança)
+--- BoxPR_curve.png ..... (Curva de Precisão vs.  Revocação /
mAP)
+--- BoxR_curve.png ..... (Curva de Revocação em função da
Confiança)
+--- confusion_matrix.png .... (Matriz de confusão absoluta em
contagem de objetos)
+--- confusion_matrix_normalized.png ..... (Matriz de confusão
normalizada por classe)
+--- labels.jpg ..... (Análise estatística e espacial das
anotações do dataset)
+--- results.csv .. (Tabela com perdas e métricas registradas
época a época)
+--- results.png .... (Painel gráfico de evolução das perdas e
métricas)
+--- train_batch0.jpg ...train_batch2.jpg ..... (Amostras de
imagens de treino com data augmentation)
+--- train_batch11800.jpg ...train_batch11802.jpg ..... (Lotes
finais de treino processados pelo modelo)
+--- val_batch0_labels.jpg . (Anotações reais/ground-truth do
lote de validação)
+--- val_batch0_pred.jpg .. (Predições geradas pelo modelo no
lote de validação)
+--- weights/
+--- best.pt ..... (Checkpoint do modelo com a melhor métrica
fitness)
+--- last.pt ..... (Checkpoint do estado final da última época
treinada)
```

A Tabela 2 categoriza os principais artefatos produzidos segundo a sua funcionalidade técnica no fluxo de desenvolvimento e auditoria de modelos de visão computacional.

Tabela 2: Classificação e descrição funcional dos artefatos do diretório de saída do YOLO.

Categoria	Arquivos	Função e Utilidade Diagnóstica
Configuração e Reprodutibilidade	*args.yaml*	Armazena o registro imutável de todos os hiperparâmetros, caminhos de dados e arquitetura utilizados na sessão.
Checkpoints de Pesos	*weights/best.pt*, *weights/last.pt*	Arquivos binários PyTorch contendo a arquitetura e os parâmetros otimizados para implantação ou retomada.
Logs Numéricos	*results.csv*	Tabela tabular detalhada com o valor numérico exato de cada perda e métrica a cada época de treino.
Gráficos de Convergência	*results.png*	Painel visual com 10 subgráficos monitorando as perdas de treino/validação e a evolução do mAP, Precisão e Revocação.
Curvas de Desempenho	*BoxP_curve.png*, *BoxR_curve.png*, *BoxF1_curve.png*, *BoxPR_curve.png*	Curvas paramétricas em função do limiar de confiança, essenciais para a calibração operacional do modelo em produção.
Diagnóstico Matricial	*confusion_matrix.png*, *confusion_matrix_normalized.png*	Matrizes de confusão absoluta e relativa para identificação de erros de classificação cruzada e confusão com o fundo.
Inspeção de Dados e Batches	*labels.jpg*, *train_batch*.jpg*, *val_batch*_*.jpg*	Imagens compostas para auditoria visual da qualidade das anotações, aumentos de dados (mosaic) e predições.

4. REGISTRO DE CONFIGURAÇÃO E CHECKPOINTS DE PESOS

4.1 O Arquivo de Hiperparâmetros (args.yaml)

A reprodutibilidade científica exige que qualquer experimento de aprendizado profundo possa ser reconstruído exatamente com os mesmos parâmetros operacionais. No ecossistema YOLO, o arquivo *args.yaml* desempenha essa função ao gravar a totalidade dos hiperparâmetros aplicados durante a chamada de treinamento. Entre os parâmetros gravados, destacam-se:

- **Hiperparâmetros de Otimização:** taxa de aprendizado inicial (*lr0*), taxa de aprendizado final (*lrf*), momento (*momentum*), decaimento de pesos (*weight_decay*) e épocas de aquecimento (*warmup_epochs*);
- **Pesos das Funções de Perda:** coeficientes de ponderação para a perda de caixa delimitadora (*box*), perda de classificação (*cls*) e perda focal de distribuição (*dfl*);
- **Aumento de Dados (*Data Augmentation*):** probabilidades e amplitudes de variações no espaço de cores HSV (*hsv_h*, *hsv_s*, *hsv_v*), rotações (*degrees*), translações (*translate*), escala (*scale*), espelhamento horizontal (*fliplr*), corte mosaico (*mosaic*) e mistura de imagens (*mixup*).

O registro em formato YAML permite que frameworks de inferência ou scripts de acompanhamento recarreguem a configuração exata do experimento para posterior ajuste fino (*fine-tuning*).

4.2 Checkpoints de Pesos: best.pt e last.pt

Dentro do subdiretório *weights/*, o YOLO mantém continuamente atualizados dois arquivos binários no formato PyTorch (*.pt*):

1. ***best.pt* (Melhor Modelo):** É o arquivo de pesos correspondente à época em que a rede atingiu o valor máximo da métrica composta de qualidade (*fitness score*). O Ultralytics calcula o *fitness* como uma combinação linear ponderada do mAP@0.50 e do mAP@0.50:0.95:

$$\text{Fitness} = 0{,}1 \cdot \text{mAP@0.50} + 0{,}9 \cdot \text{mAP@0.50:0.95}.$$

A ponderação de 90% atribuída ao mAP@0.50:0.95 prioriza modelos que demonstram alta precisão geométrica na localização das caixas delimitadoras. O arquivo *best.pt* é o artefato indicado para implantação direta em ambiente de produção.

2. ***last.pt* (Último Estado):** Preserva os pesos e os estados do otimizador referentes à última época processada (por exemplo, a época 200 de 200). Este arquivo é fundamental por duas razões:
   - **Retomada de Treinamento (*Resume*):** Permite reiniciar o treinamento exatamente a partir do ponto onde foi interrompido (devido a falhas de energia, estouro de tempo

em clusters ou interrupções voluntárias), utilizando o comando *YOLO("last.pt"). train(resume=True)*;

- **Análise de Sobreajuste (*Overfitting*):** Permite comparar o comportamento do modelo no final da execução em relação ao seu pico de desempenho mantido no *best.pt*.

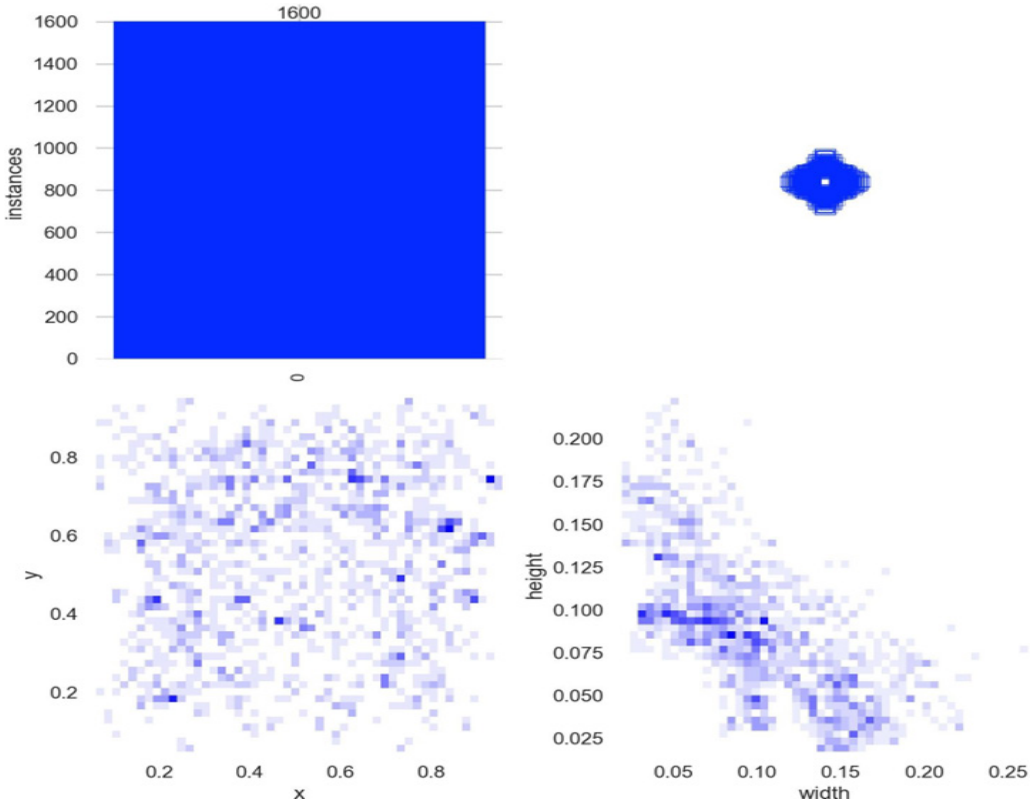
5. INSPEÇÃO QUALITATIVA DOS DADOS E BATCHES (TREINO E VALIDAÇÃO)

Uma premissa fundamental da visão computacional aplicada é a verificação da qualidade dos dados antes e durante o processo de otimização. O YOLO produz automaticamente representações visuais dos dados de entrada e dos lotes (*batches*) processados pela rede.

5.1 Estatísticas Espaciais e Categorias do Dataset (labels.jpg)

O arquivo *labels.jpg* reúne quatro painéis estatísticos que resumem a distribuição categórica e a geometria das anotações contidas no conjunto de treinamento, conforme demonstrado na Figura 17.

Figura 17: Painel estatístico de rótulos do conjunto de dados (labels.jpg). Superior esquerdo: Histograma de contagem por classe. Superior direito: Distribuição espacial e dimensões das caixas delimitadoras. Inferior esquerdo: Mapeamento dos centros (x, y) das caixas. Inferior direito: Relação entre largura (w) e altura (h) dos objetos.



A análise detalhada de cada quadrante do *labels.jpg* fornece diagnósticos valiosos:

- **Histograma de Classes (Superior Esquerdo):** Apresenta a contagem absoluta de instâncias por categoria. Permite identificar o grau de desbalanceamento de classes no dataset, alertando o projetista para a necessidade de técnicas de reponderação ou amostragem balanceada.
- **Distribuição de Caixas por Imagem (Superior Direito):** Mostra a sobreposição e densidade espacial das caixas delimitadoras anotadas no conjunto de dados.
- **Centros das Caixas Delimitadoras (Inferior Esquerdo):** Mapeia as posições relativas dos centros das caixas (*x, y*) normalizadas no intervalo [0, 1]. Concentrações excessivas no centro do gráfico indicam viés de enquadramento (*center bias*), comum quando fotógrafos tendem a centralizar o objeto principal.
- **Dimensões das Caixas (Inferior Direito):** Plota a largura (*w*) versus a altura (*h*) de todas as marcações. Permite verificar a variação de escala dos alvos e a prevalência de objetos pequenos, médios ou grandes, orientando a escolha dos tamanhos de *anchors* ou da resolução de entrada do modelo (*imgsz*).

5.2 Batches de Treinamento e Data Augmentation (train_batch*.jpg)

Durante o treinamento, o gerador de dados do YOLO aplica transformações estocásticas nas imagens de cada lote para aumentar a capacidade de generalização da rede. Os arquivos *train_batch0.jpg*, *train_batch1.jpg* e *train_batch2.jpg* exibem as primeiras amostras do dataset já processadas pelas técnicas de aumento de dados, com as caixas de gabarito sobrepostas. A Figura 18 ilustra um lote inicial de treinamento (*train_batch0.jpg*).

Entre as transformações visíveis em *train_batch0.jpg*, destaca-se o Aumento por Mosaico (*Mosaic Augmentation*), no qual quatro imagens distintas são recortadas, redimensionadas e combinadas em um único painel composto. Essa técnica força o modelo a detectar objetos em contextos visuais variados, em escalas reduzidas e próximos às bordas da imagem, reduzindo drasticamente a dependência da rede quanto ao fundo original das fotografias. A Figura 19 apresenta o lote subsequente (*train_batch1.jpg*), confirmando a variabilidade das transformações aplicadas entre lotes consecutivos.

Figura 18: Visualização do lote de treinamento train_batch0.jpg, demonstrando a aplicação do aumento de dados por colagem Mosaico 4-way (Mosaic Augmentation) e ajuste de espaço de cor HSV.

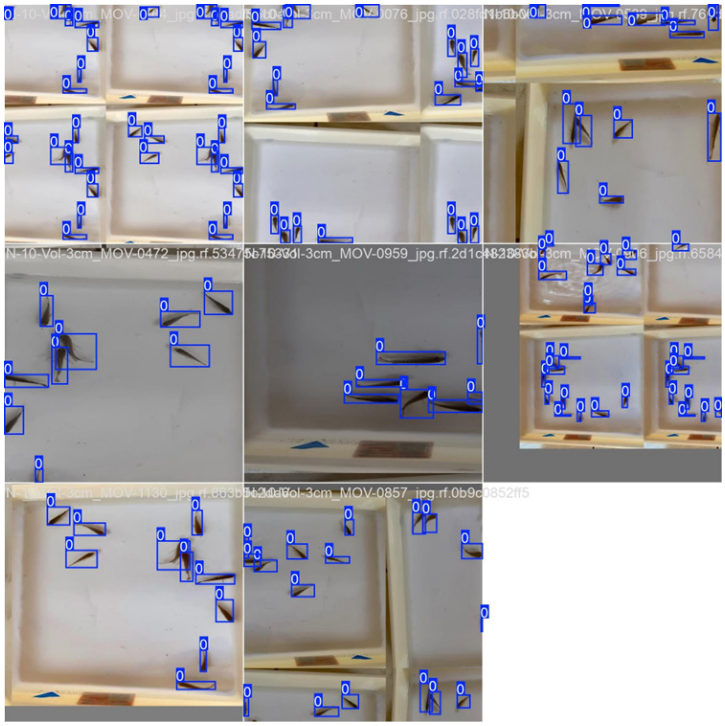
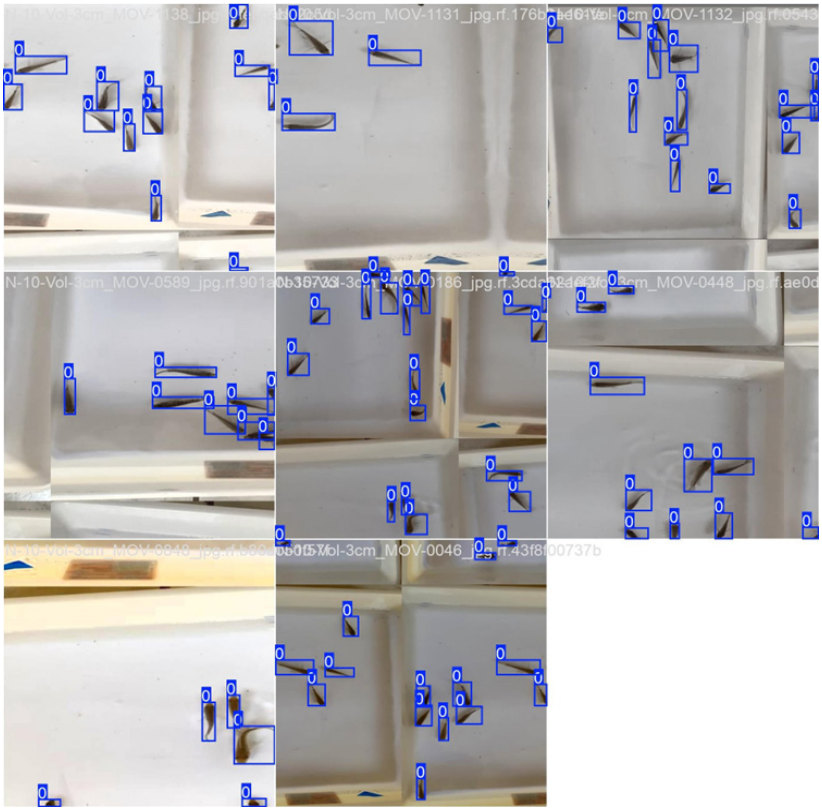


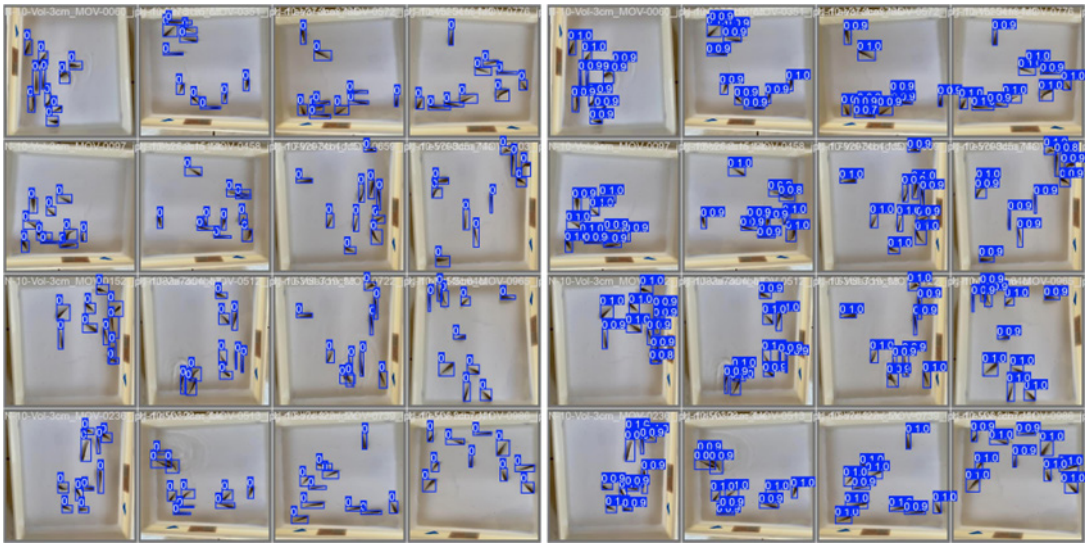
Figura 19: Visualização do lote de treinamento train_batch1.jpg, exibindo diferentes composições de mosaico e variação de iluminação.



5.3 Validação Qualitativa: Gabarito vs. Predições do Modelo

Ao final de cada época ou etapa de validação, o YOLO seleciona lotes do conjunto de validação para gerar duas imagens comparativas: *val_batch0_labels.jpg* (caixas de gabarito reais anotadas por especialistas humanos) e *val_batch0_pred.jpg* (caixas delimitadoras e rótulos previstos pela rede neural, acompanhados das pontuações de confiança). A Figura 20 apresenta a comparação lado a lado entre o gabarito real e a predição gerada pela rede no primeiro lote de validação.

Figura 20: Inspeção qualitativa de validação no lote 0: comparação entre as anotações reais de referência (val_batch0_labels.jpg, à esquerda) e as detecções inferidas pelo modelo YOLO (val_batch0_pred.jpg, à direita).



Essa inspeção visual comparativa é uma ferramenta indispensável para identificar padrões sistemáticos de erro:

1. **Falsos Positivos Visualizáveis:** Caixas presentes na predição que não existem no gabarito revelam regiões do cenário que enganam o modelo;
2. **Falsos Negativos (Omissões):** Objetos marcados no gabarito que não receberam caixa predita indicam alvos não detectados;
3. **Precisão do Ajuste da Caixa:** Permite avaliar se os limites das caixas previstas pela rede se ajustam perfeitamente ao contorno dos objetos ou se apresentam folgas ou cortes significativos.

6. ANÁLISE DETALHADA DAS CURVAS DE DESEMPENHO E LIMIARES DE CONFIANÇA

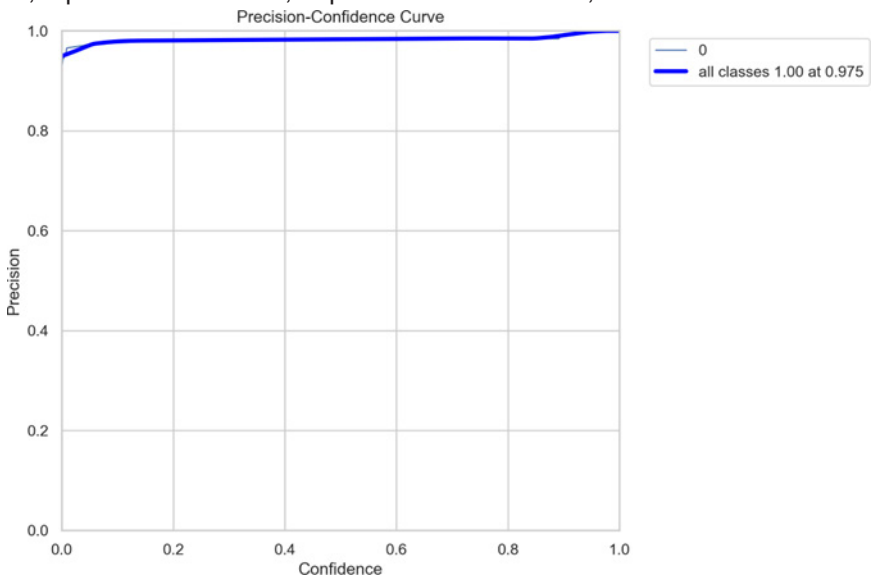
Durante o processo de inferência, o YOLO atribui a cada detecção uma pontuação de confiança conf $\in [0, 1]$, que expressa a probabilidade estimada de a caixa conter um objeto multiplicada pelo IoU previsto. A escolha do limiar de corte de confiança ($conf_{thresh}$) altera radicalmente o equilíbrio entre Precisão e Revocação do sistema. Para orientar essa

calibração, o YOLO gera quatro curvas paramétricas fundamentais.

6.1 Curva Precisão-Confiança (BoxP_curve.png)

A Curva Precisão-Confiança (Figura 21) ilustra a evolução da Precisão (P) à medida que o limiar de confiança mínimo exige predições cada vez mais rigorosas.

Figura 21: Curva Precisão-Confiança (BoxP_curve.png). À medida que o limiar de confiança se aproxima de 1,00, a precisão tende a 1,00 para todas as classes, eliminando Falsos Positivos.

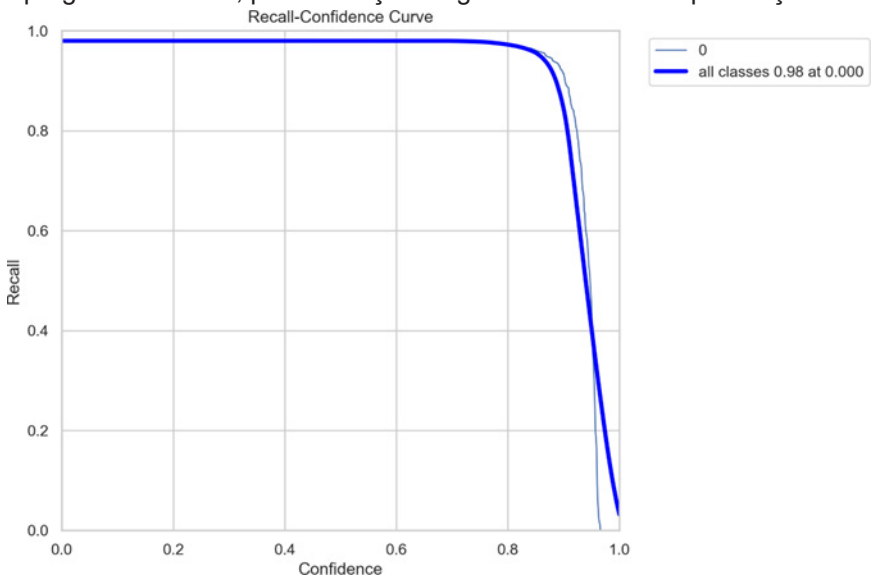


Como regra geral, aumentando o limiar $conf_{thresh}$, o modelo descarta predições incertas, reduzindo o número de Falsos Positivos (FP) e fazendo a Precisão se aproximar de 1,00 (100%). Em aplicações industriais que não toleram alarmes falsos (como controle de qualidade automatizado), seleciona-se um limiar elevado nesta curva.

6.2 Curva Revocação-Confiança (BoxR_curve.png)

Em contrapartida, a Curva Revocação-Confiança (Figura 22) demonstra a variação da Revocação (R) em função do limiar de confiança. Ao elevar o limiar de confiança, o sistema torna-se extremamente seletivo e deixa de detectar objetos que apresentem menor grau de certeza, aumentando os Falsos Negativos (FN) e reduzindo a Revocação. Em sistemas de segurança ou diagnóstico médico, onde falhar na detecção de um alvo é crítico, escolhe-se um limiar de confiança mais baixo nesta curva.

Figura 22: Curva Revocação-Confiança (BoxR_curve.png). Conforme o limiar de confiança aumenta, a revocação cai progressivamente, pois detecções legítimas com menor pontuação são descartadas.



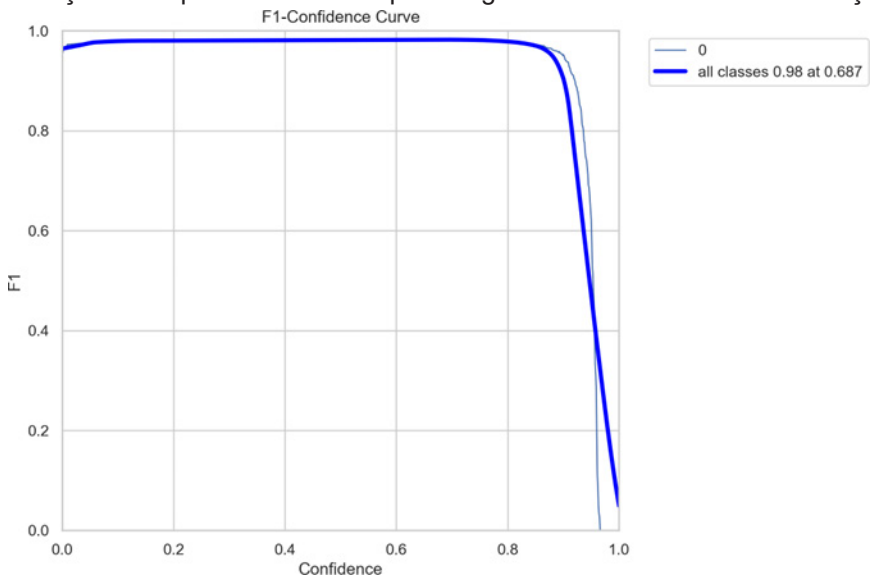
6.3 Curva F1-Score-Confiança (BoxF1_curve.png)

Para encontrar o ponto de equilíbrio ótimo entre Precisão e Revocação sem priorizar arbitrariamente nenhuma das duas métricas, adota-se o F_1-Score, definido pela média harmônica entre P e R:

$$F_1(\text{conf}) = 2 \cdot \frac{P(\text{conf}) \cdot R(\text{conf})}{P(\text{conf}) + R(\text{conf})}$$

A Figura 23 apresenta a curva F_1-Score em função da Confiança (*BoxF1_curve.png*). O gráfico destaca o valor de confiança que maximiza o F_1-score específico desse experimento de exemplo (neste caso, conf = 0,38 com F_1 = 0,89). Ressalta-se que esse valor de limiar de corte não é um padrão fixo universal, devendo ser calibrado dinamicamente para cada aplicação conforme o custo operacional de Falsos Positivos versus Falsos Negativos.

Figura 23: Curva F1-Score-Confiança (BoxF1_curve.png). O ponto de pico da curva indica o limiar de confiança ótimo que maximiza o equilíbrio global entre Precisão e Revocação.



6.4 Curva Precisão-Revocação (BoxPR_curve.png)

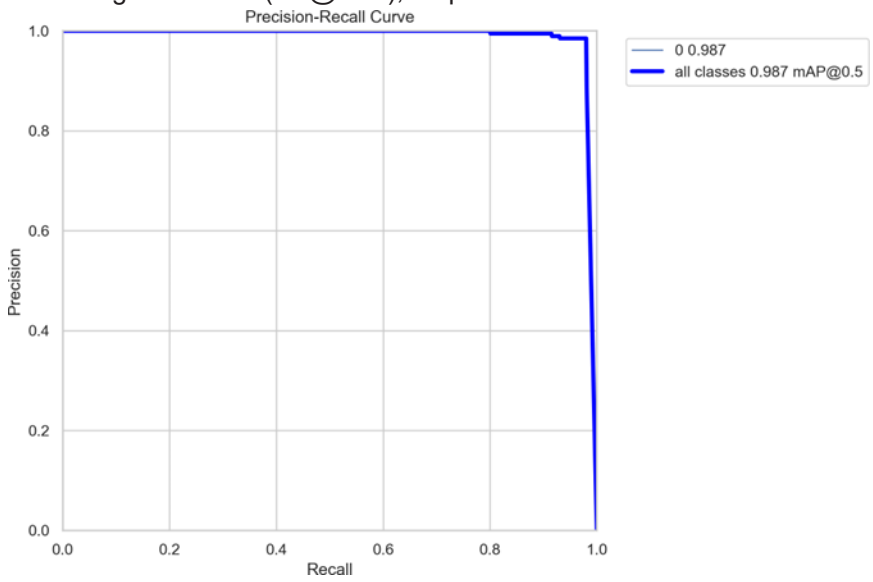
A Curva Precisão-Revocação (Figura 24) sintetiza o *trade-off* direto entre Precisão (eixo Y) e Revocação (eixo X) para todas as categorias de objetos, independentemente do limiar de confiança explícito. Em uma curva PR ideal, a linha estende-se até o canto superior direito (coordenada (1.0, 1.0)), indicando que o modelo alcança alta precisão mantendo alta revocação. A área delimitada abaixo de cada curva individual representa o AP daquela classe específica; o valor global exibido na legenda (mAP@0.50) é a média aritmética das áreas de todas as categorias.

7. EVOLUÇÃO TEMPORAL DAS FUNÇÕES DE PERDA E MÉTRICAS (RESULTS.PNG E RESULTS.CSV)

7.1 Painel Visual de Desempenho (results.png)

O arquivo *results.png* agrega 10 gráficos de linha dispostos em duas fileiras principais, permitindo o acompanhamento temporal simultâneo da convergência das perdas e da ascensão das métricas ao longo de todas as épocas de treinamento, conforme ilustrado na Figura 25.

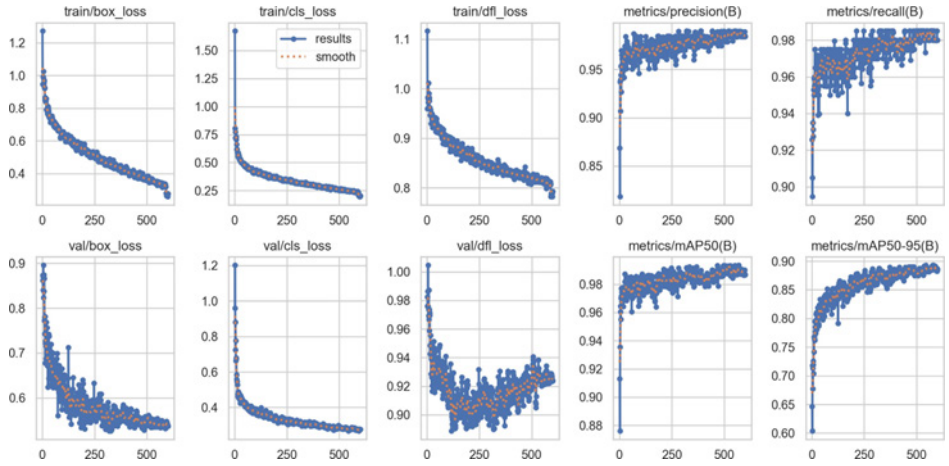
Figura 24: Curva Precisão-Revocação (BoxPR_curve.png). A área sob a curva (AUC) de cada classe corresponde ao seu Average Precision (AP@0.50), enquanto a média entre as classes define o mAP@0.50.



As três funções de perda monitoradas pelo YOLO possuem papéis matemáticos distintos na otimização da rede:

1. **Perda de Caixa (*Box Loss* — L_{box}):** Mensura o erro de alinhamento espacial entre as caixas delimitadoras previstas e os gabaritos reais. Utiliza a métrica Complete IoU (CIoU), que penaliza diferenças de posição central, sobreposição de área e proporção de aspecto (*aspect ratio*);
2. **Perda de Classificação (*Class Loss* — L_{cls}):** Avalia a acurácia na identificação das categorias dos objetos utilizando a perda de Entropia Cruzada Binária (*Binary Cross-Entropy Loss* — BCE);
3. **Perda Focal de Distribuição (*Distribution Focal Loss* — L_{dfl}):** Utilizada nas arquiteturas YOLO de última geração (YOLOv8/YOLOv12) para tratar a incerteza nos limites exatos das bordas dos objetos quando a imagem apresenta oclusão ou contornos difusos, otimizando uma distribuição de probabilidade contínua sobre as posições dos limites.

Figura 25: Painel completo de evolução temporal do treinamento (results.png). Linha 1: Perdas de treinamento (train/box_loss, train/cls_loss, train/dfl_loss) e métricas (metrics/precision, metrics/recall). Linha 2: Perdas de validação (val/box_loss, val/cls_loss, val/dfl_loss) e métricas de mAP (metrics/mAP50, metrics/mAP50-95).



7.2 Diagnóstico de Convergência e Sobreajuste (Overfitting)

A comparação direta entre a fileira de perdas de treino (*train/*) e a fileira de perdas de validação (*val/*) no gráfico *results.png* fornece o diagnóstico definitivo sobre a dinâmica do aprendizado:

- **Aprendizado Saudável (Convergência Estável):** As perdas de treino e de validação decrescem de forma suave e paralela ao longo das épocas, enquanto o mAP50 e o mAP50-95 sobem continuamente até atingirem um patamar de estabilização (*plateau*);
- **Sintoma de Sobreajuste (*Overfitting*):** A perda de treino continua caindo, mas a perda de validação estabiliza e começa a subir a partir de determinada época, acompanhada da estagnação ou queda do mAP de validação. Nesse cenário, o modelo memorizou os padrões do treino mas perdeu a capacidade de generalizar;
- **Sintoma de Subajuste (*Underfitting*):** Todas as perdas mantêm valores elevados ao final das épocas programadas, indicando que o modelo necessita de mais épocas de treinamento, maior capacidade arquitetural ou taxas de aprendizado ajustadas.

7.3 Registro Numérico Estruturado (results.csv)

Enquanto o arquivo *results.png* destina-se à inspeção visual rápida, o arquivo *results.csv* armazena os valores numéricos exatos de todas as variáveis monitoradas em cada época no formato tabular CSV (*Comma-Separated Values*). Cada linha do *results.csv* contém as seguintes colunas padronizadas:

epoch, train/box_loss, train/cls_loss, train/dfl_loss, metrics/precision, metrics/recall, metrics/mAP50, metrics/mAP50-95, val/box_loss, val/cls_loss, val/dfl_loss, lr/pg0, lr/pg1, lr/pg2

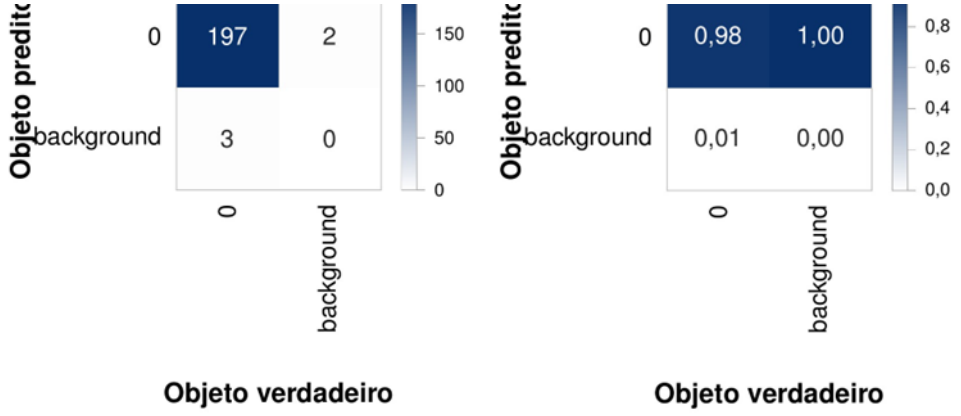
Esse registro tabular permite que pesquisadores importem os dados em bibliotecas de análise científica (como Pandas, NumPy e Seaborn) para gerar gráficos customizados, realizar análises estatísticas comparativas entre diferentes arquiteturas ou aplicar algoritmos de decisão automatizada para parada antecipada (*early stopping*).

8. MATRIZ DE CONFUSÃO ABSOLUTA VS. NORMALIZADA

O diretório de saída do treinamento gera duas versões suplementares da matriz de confusão: *confusion_matrix.png* (contagens absolutas) e *confusion_matrix_normalized.png* (proporções relativas).

A Figura 26 apresenta a comparação visual entre essas duas modalidades de diagnóstico matricial.

Figura 26: Comparação entre a matriz de confusão em contagens brutas de instâncias (confusion_matrix.png, à esquerda) e a matriz normalizada por classe real (confusion_matrix_normalized.png, à direita).



(a) Matriz de Confusão Absoluta (*confusion_matrix.png*)

(b) Matriz Normalizada (*confusion_matrix_normalized.png*)

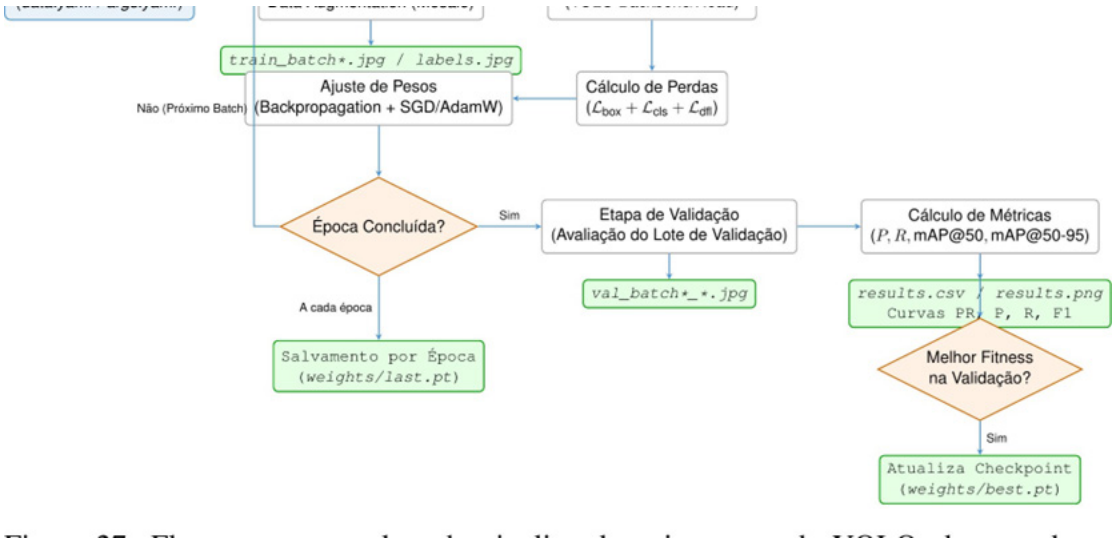
A utilidade complementar de cada versão reside na escala de análise:

1. **Matriz de Contagens Absolutas (*confusion_matrix.png*):** Exibe o número exato de instâncias físicas detectadas. É indispensável para avaliar a escala do dataset e identificar o volume absoluto de erros em categorias críticas;
2. **Matriz Normalizada (*confusion_matrix_normalized.png*):** Divide o valor de cada célula pela soma total de instâncias pertencentes àquela classe real, expressando a taxa de acertos e erros no intervalo [0, 1]. Esta versão é fundamental para conjuntos de dados desbalanceados, pois evita que a classe com maior número de exemplares domine a percepção visual do gráfico, permitindo avaliar a verdadeira taxa de sensibilidade de classes minoritárias.

9. FLUXOGRAMA INTEGRADO DO PIPELINE DE TREINAMENTO E DIAGNÓSTICO

Para sintetizar o ciclo de vida operacional da avaliação de modelos YOLO, a Figura 27 apresenta o fluxograma completo do pipeline de treinamento, validação época a época, geração de artefatos de saída e tomada de decisão para implantação final.

Figura 27: Fluxograma completo do pipeline de treinamento do YOLO, destacando as etapas de aumentos de dados, otimização de perdas, validação época a época e exportação dos artefatos de diagnóstico e checkpoints.



Em suma, a interpretação sistemática do conjunto de artefatos gerados pelo treinamento do YOLO permite transformar o processo de desenvolvimento de redes profundas de uma abordagem empírica de "caixa preta" em um fluxo de engenharia auditável, rigoroso e previsível.

REFERÊNCIAS

EVERINGHAM, Mark et al. The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, v. 88, n. 2, p. 303–338, 2010.

FRANÇA, Elzi Leida do Carmo et al. Detecção de abelhas nativas do Cerrado Brasileiro com Rede Neural Convolucional YOLOv8. Caderno Pedagógico, v. 22, n. 11, e19970, 2025. DOI: https://doi.org/10.54033/cadpedv22n11-207.

PADILLA, Rafael; NETTO, Sergio L.; SILVA, Eduardo A. B. da. A Survey on Performance Metrics for Object-Detection Algorithms. In: INTERNATIONAL Conference on Systems, Signals and Image Processing (IWSSIP). [S.l.: s.n.], 2020. P. 237–242.