UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Vinicius Claudino Antunes

Desenvolvimento de modelos de deep learning para detecção de emoções durante saltos de paraquedismo

Florianópolis,
2024

Vinicius Claudino Antunes

Desenvolvimento de modelos de deep learning para detecção de emoções durante saltos de paraquedismo

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação, do Departamento de Informática e Estatística, do Centro Tecnológico da Universidade Federal de Santa Catarina, como parte dos requisitos para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Elder Rizzon Santos.

Florianópolis,
2024
Vinicius Claudino Antunes

Desenvolvimento de modelos de deep learning para detecção de emoções durante saltos de paraquedismo

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Sistemas de Informação da Universidade Federal de Santa Catarina.

Florianópolis, 18 de novembro de 2024.

_____

Coordenação do curso

Banca examinadora

_____

Prof. Dr. Elder Rizzon Santos (UFSC)
Orientador

_____

Prof. Dr. Alexandre Gonçalves Silva (UFSC)

_____

Dr. Rodrigo Rodrigues Pires de Mello (UFSC)

Florianópolis, 2024.

RESUMO

O reconhecimento de expressões faciais tem uma longa história, com contribuições significativas de estudiosos como Charles Darwin e Paul Ekman, que estabeleceram as bases para a ciência moderna de detecção de emoções. Com os avanços tecnológicos, especialmente em visão computacional, sistemas automáticos de reconhecimento de emoções têm encontrado aplicações em áreas como robótica e interação humano-computador. Este trabalho propõe o desenvolvimento de modelos de deep learning para detecção de emoções durante saltos de paraquedismo, um contexto de alto impacto emocional. Utilizando o algoritmo YOLO, o modelo visa identificar três emoções – medo, neutro e feliz –, contribuindo para o avanço da pesquisa na área de reconhecimento de emoções em cenários dinâmicos e desafiadores. Os resultados indicaram que os modelos YOLOv10 apresentaram melhor equilíbrio entre as métricas de treinamento e teste, enquanto os modelos YOLOv8 mostraram indícios de overfitting devido à menor capacidade de generalização. O conjunto de dados limitado e as semelhanças visuais entre certas classes foram as principais limitações, destacando a necessidade de ampliar e diversificar o dataset. Como trabalhos futuros, propõe-se a criação de um conjunto de dados mais robusto e a exploração de arquiteturas alternativas, para melhorar o desempenho e a capacidade de generalização dos modelos.

Palavras-chave: deep learning; detecção de objetos; reconhecimento de expressões faciais; visão computacional; YOLO; paraquedismo.

ABSTRACT

Recognition of facial expressions has a long history, with significant contributions from scholars such as Charles Darwin and Paul Ekman, which have established themselves as the basis for the modern science of emotion detection. With technological advances, especially in computer vision, automatic emotion recognition systems have found applications in areas such as robotics and human-computer interaction. This work proposes the development of a deep learning model to detect emotions during skydiving, a context of high emotional impact. Using the YOLO algorithm, the model can identify three emotions – fear, neutral and happy – contributing to the search for emotion recognition in dynamic and distressing scenarios. The results indicated that the YOLOv10 models presented a better balance between training and testing metrics, while the YOLOv8 models showed signs of overfitting due to lower generalization capacity. The limited dataset and the visual similarities between certain classes were the main limitations, highlighting the need to expand and diversify the dataset. As future work, we propose the creation of a more robust dataset and the exploration of alternative architectures, such as ResNet, to improve the performance and generalization capacity of the models.

Keywords: deep learning; object detection; facial expression recognition; computer vision; YOLO; skydiving.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIAÇÕES

CNN	Convolutional Neural Network
FACS	Facial Action Coding
FDDB	Face Detection Dataset and Benchmark
FER2013	Facial Expression Recognition 2013 Dataset
GPU	Graphic Processing Unit
HOG	Histogram of Oriented Gradients
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NMS	Non-Maximum Suppression
PAN	Path Aggregation Network
PAN-Net	Path Aggregation Network
R-CNN	Regions with Convolutional Neural Network
RBF	Radial Basis Function
RBFN	Radial Basis Function Network
ReLu	Rectified Linear Unit
ResNet	Residual Network
SeNet	Squeezed-and-Excitation Network
SGD	Stochastic Gradient Descent
SIFT	Scale-Invariant Feature Transform
SSD	Single Shot MultiBox Detector
SVM	Support Vector Machine
UA	Unit Action
YOLO	You Only Look Once

SUMÁRIO

1. INTRODUÇÃO

O interesse pelo estudo das expressões faciais vem sendo despertado há séculos, com registros que datam da época de Aristóteles. Desde então, filósofos, artistas, psicólogos e cientistas têm explorado a relação entre a aparência do rosto humano e as emoções. Obras de figuras como John Bulwer, Charles Darwin e Paul Ekman são marcos históricos que ajudaram a estabelecer os fundamentos para a compreensão das expressões faciais. Esses estudos formaram a base para os atuais sistemas de reconhecimento automático de emoções, que são utilizados em uma ampla gama de aplicações, desde a interação humano-computador até a robótica e a inteligência artificial (BETTADAPURA, 2012).

O reconhecimento de expressões faciais não se limita ao campo da psicologia, expandindo-se para a ciência da computação nas últimas décadas. Os avanços tecnológicos, especialmente no campo da visão computacional, permitiram o desenvolvimento de sistemas de reconhecimento facial, que podem detectar e classificar as expressões em tempo real. O trabalho pioneiro de Ekman (1993), que identificou seis expressões faciais universais – alegria, tristeza, raiva, medo, surpresa e nojo –, foram cruciais para o desenvolvimento de algoritmos modernos que classificam essas emoções de forma precisa.

Esses sistemas encontram aplicação em diversas áreas, como a robótica, onde robôs precisam interpretar e responder a estados emocionais, e na interação humano-computador, que busca tornar as interfaces tecnológicas mais intuitivas e sensíveis ao estado emocional dos usuários. Além disso, áreas como a medicina, segurança e entretenimento digital se beneficiam do reconhecimento automático de expressões faciais para melhorar a experiência do usuário, diagnosticar condições emocionais e aprimorar a interação em ambientes virtuais (BETTADAPURA, 2012).

Diante desse cenário, o presente trabalho tem como objetivo o desenvolvimento de modelos de *deep learning* capazes de detectar emoções durante saltos de paraquedismo. A escolha desse contexto específico se justifica pela intensidade emocional envolvida na atividade, no qual as emoções de medo, neutro e feliz são predominantes. Utilizando o algoritmo *You Only Look Once* (YOLO) para detecção e classificação, os modelos propostos visam identificar expressões faciais com precisão em um ambiente desafiador e dinâmico, contribuindo para a pesquisa na área de reconhecimento de emoções em situações de alto impacto emocional. A importância deste trabalho está na crescente necessidade de sistemas capazes de

reconhecer e interpretar emoções humanas de maneira eficiente em diferentes contextos, principalmente em ambientes não controlados e com emoções totalmente genuínas.

1.1 OBJEIVO GERAL

O objetivo geral deste trabalho é desenvolver modelos de *deep learning* capazes de identificar e classificar expressões faciais em imagens capturadas tanto durante saltos de paraquedismo quanto momentos antes do salto, dentro da aeronave. O modelo será treinado para prever, com precisão, se as expressões detectadas correspondem a uma das três classes emocionais: feliz, neutro ou medo, contribuindo para a análise de emoções em contextos de situações extremas e momentos prévios ao salto.

1.2 OBJETIVOS ESPECÍFICOS

De forma a cumprir com o objetivo geral deste trabalho, os seguintes objetivos específicos foram definidos:

a) Analisar a fundamentação teórica e o estado da arte referente à detecção de objetos na análise de expressões faciais.
b) Coletar e preparar o conjunto de dados, contendo imagens de rostos capturados previamente e durante o salto.
c) Desenvolvimento de modelos de *machine learning* para detecção de objetos na análise de expressões faciais.
d) Delinear experimentos para avaliação dos modelos construídos.
e) Analisar os resultados obtidos por meio dos experimentos.

2. FUNDAMENTAÇÃO TEÓRICA

Esta seção tem como objetivo fornecer os principais conceitos fundamentais para a compreensão e desenvolvimento deste trabalho, para que cada objetivo específico seja atingido. Além disso, revisou-se outros trabalhos, a fim de fornecer o embasamento necessário ao trabalho proposto.

2.1 INTELIGÊNCIA ARTIFICIAL

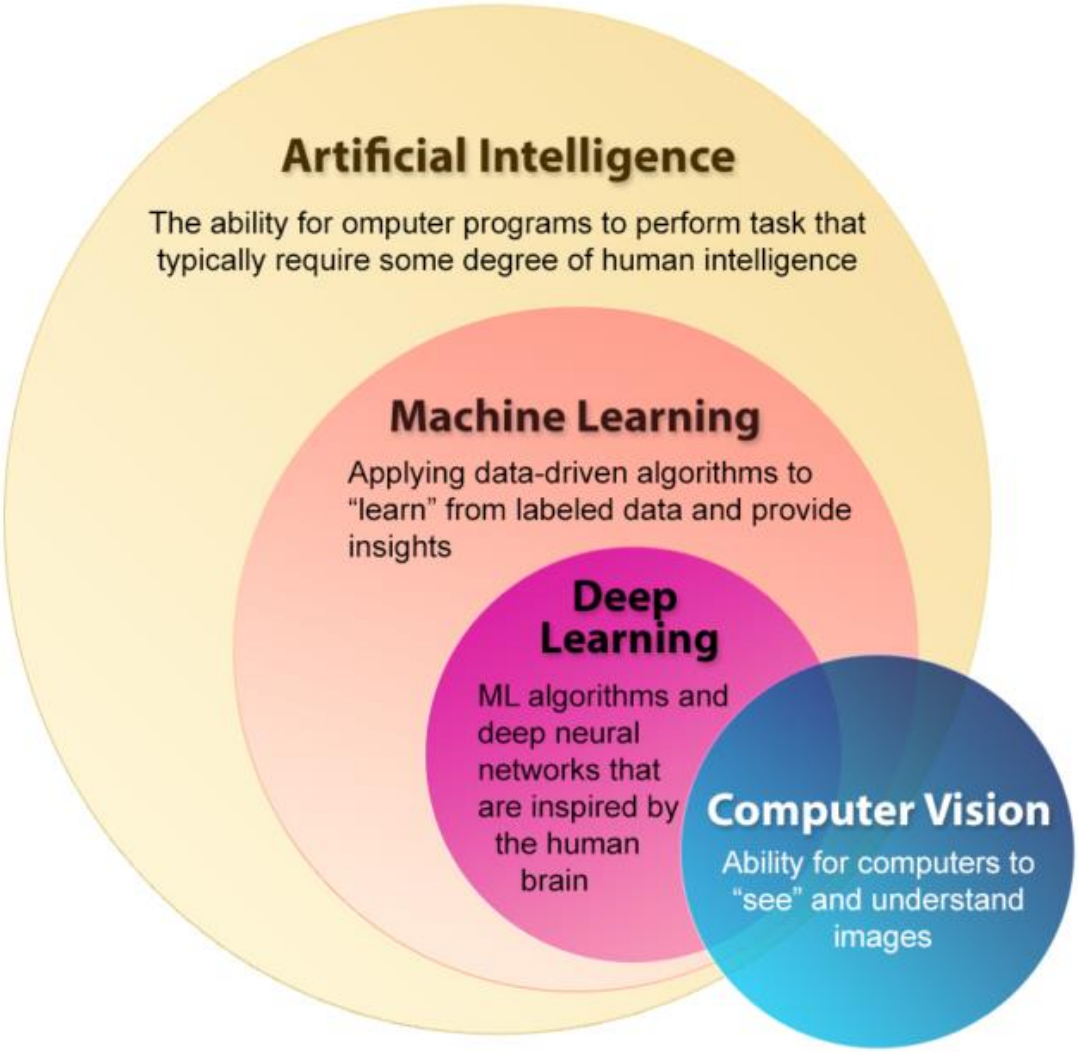
A Inteligência Artificial (IA) é uma área interdisciplinar da ciência da computação dedicada ao desenvolvimento de sistemas e algoritmos capazes de realizar tarefas que, quando realizadas por humanos, requerem inteligência. Desde sua concepção, a IA tem evoluído e se diversificado, abrangendo várias subáreas que refletem a complexidade e a amplitude de suas aplicações. De acordo com Russel e Norvig (2013), a inteligência artificial pode ser compreendida a partir de quatro perspectivas distintas: pensar como humanos, agir como humanos, pensar racionalmente e agir racionalmente.

Entre essas abordagens, a definição preferida pelos autores é a de agentes racionais. Um agente inteligente, conforme descrito por Russel e Norvig, é uma entidade que percebe seu ambiente por meio de sensores e age sobre ele através de atuadores, tomando decisões que maximizam suas chances de sucesso na realização de seus objetivos. Esses agentes são projetados para avaliar continuamente seu estado atual, considerar possíveis ações, prever os resultados dessas ações e escolher a melhor ação com base em um conjunto de critérios ou objetivos (RUSSELL e NORVIG, 2013).

A capacidade de um agente inteligente de tomar decisões racionais está intimamente ligada à sua habilidade de aprender e se adaptar ao longo do tempo, o que nos leva ao campo do aprendizado de máquina (*machine learning*). *Machine learning* é uma subárea da IA que se concentra em desenvolver algoritmos que permitam aos agentes aprender a partir de dados, melhorando seu desempenho de forma contínua sem serem explicitamente programados para cada tarefa específica. Este aprendizado pode ser supervisionado, não supervisionado ou por reforço, cada um com suas próprias metodologias e aplicações.

A figura a seguir ilustra a relação hierárquica e interconexão entre os campos da Inteligência Artificial (IA), Aprendizado de Máquina (Machine Learning), Aprendizado Profundo (Deep Learning) e Visão Computacional (Computer Vision). A IA abrange todas as tecnologias que permitem que máquinas realizem tarefas que normalmente exigem inteligência humana. Dentro da IA, está o Aprendizado de Máquina, que utiliza algoritmos baseados em dados para gerar insights a partir de dados rotulados. O Aprendizado Profundo, por sua vez, é uma subárea do Aprendizado de Máquina que se baseia em redes neurais profundas inspiradas no funcionamento do cérebro humano. A Visão Computacional, destacada na imagem como uma área relacionada, aplica esses conceitos para permitir que computadores interpretem e entendam imagens.

Figura 1 - Diagrama de Venn: Inteligência Artificial.



Fonte: ESRI Canada Centre of Excellence (ECCE). Pavement marking inventory using GIS and computer vision – Pt. 1 (2022).

2.2 APRENDIZADO DE MÁQUINA

Mitchell (1997) define aprendizagem de máquina como o estudo de algoritmos que permitem que os computadores aprendam, a partir de dados, a melhorarem seu desempenho em tarefas específicas sem serem explicitamente programados para isso. Para Russel e Norvig (2013), aprendizado de máquina é uma abordagem para a construção de sistemas que melhoram automaticamente com a experiência. Envolve

a criação de algoritmos que permitem que um sistema use dados para detectar padrões e fazer previsões ou tomar decisões sem serem explicitamente programados para cada tarefa.

2.3 DETECÇÃO DE OBJETOS

Para compreender completamente uma imagem, é essencial não apenas classificar diferentes imagens, mas também identificar e localizar com precisão os objetos nela contidos. Essa tarefa, conhecida como detecção de objetos, engloba subtarefas como detecção de rostos e detecção de pedestres. Como um dos problemas fundamentais da visão computacional, a detecção de objetos fornece informações valiosas para a compreensão semântica de imagens e vídeos, sendo relevante para diversas aplicações, incluindo classificação de imagens, análise do comportamento humano, reconhecimento facial e condução autônoma (ZHAO *et al*. 2019).

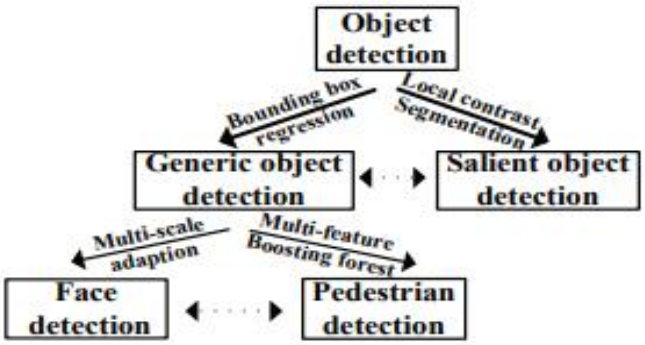
Para Zhao *et al*. (2019), a definição do problema de detecção de objetos envolve determinar onde os objetos estão localizados em uma imagem (localização de objetos) e a qual categoria cada objeto pertence (classificação de objetos). O processo dos modelos de detecção de objetos pode ser dividido em três etapas principais – identificação de regiões relevantes, extração de características e classificação:

- Identificação de regiões relevantes: como diferentes objetos podem aparecer em qualquer posição da imagem e ter diferentes proporções ou tamanhos, uma abordagem comum é escanear toda a imagem com uma janela deslizante em várias escalas. Embora essa estratégia exaustiva possa identificar todas as possíveis posições dos objetos, ela é computacionalmente cara e gera muitas janelas redundantes.
- Extração de características: para reconhecer objetos, é essencial extrair características visuais que ofereçam uma representação semântica e robusta, como as características SIFT, HOG e *Haar-like*. Contudo, devido à diversidade de aparências, iluminação e fundos, é desafiador projetar manualmente um descritos de características que descreva todos os tipos de objetos de maneira robusta.
- Classificação: um classificador é necessário para distinguir um objeto alvo de todas as outras categorias, tornando as representações mais

hierárquicas, semânticas e informativas para o reconhecimento visual. Um exemplo desses classificadores é o algoritmo *Support Vector Machine* (SVM).

A figura a seguir organiza os métodos de detecção de objetos em duas abordagens principais: detecção genérica, que usa técnicas como regressão de caixas delimitadoras e adaptação em múltiplas escalas, e detecção de objetos salientes, que foca em segmentação e contraste local. Essas abordagens são aplicadas em áreas específicas, como detecção de rostos e pedestres, mostrando a interconexão entre as técnicas para melhorar os resultados.

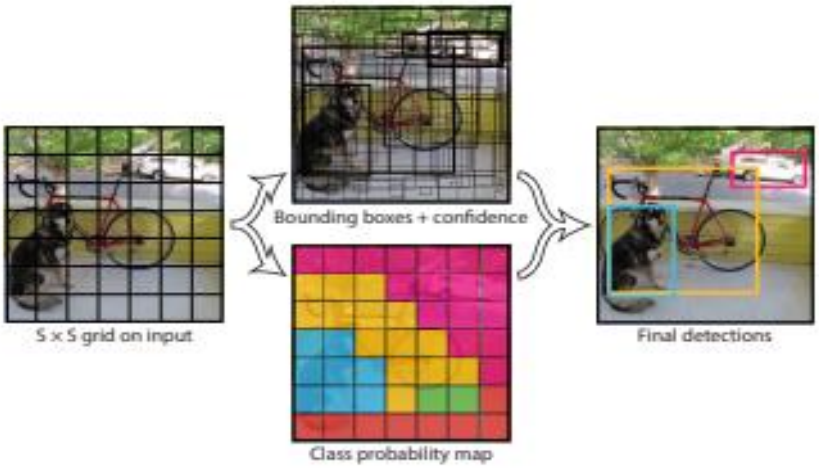
Figura 2 - Domínios de aplicação de detecção de objetos.



Fonte: ZHAO, Zhong-Qiu *et. al*., (2018).

Os métodos mais avançados de detecção de objetos podem ser divididos em duas categorias principais: métodos de um estágio e métodos de dois estágios. Os métodos de um estágio, como YOLO, SSD e RetinaNet, são projetados para priorizar a velocidade de inferência, permitindo a detecção rápida de objetos em tempo real. Por outro lado, os métodos de dois estágios, como Faster R-CNN, Mask R-CNN e Cascade R-CNN, focam na precisão da detecção. Eles primeiro geram propostas de regiões no qual os objetos podem estar localizados e, em seguida, refinam essas propostas para classificar e localizar os objetos com maior precisão. Para o presente estudo, será abordado o algoritmo YOLO para a detecção de objetos.

Figura 3 - Funcionamento algoritmo YOLO.



Fonte: REDMON, Joseph *et al*. (2016).

O algoritmo YOLO é uma abordagem de detecção de objetos que trata a detecção como um problema de regressão único e direto, partindo da imagem completa para coordenadas de *bounding boxes* e probabilidades de classes. O algoritmo divide a imagem de entrada em uma grade e, para cada célula dessa grade, prevê *bounding boxes* e probabilidades de classe diretamente. Esta abordagem permite que o modelo seja extremamente rápido e eficiente, possibilitando a detecção de objetos em tempo real com alta precisão (REDMON *et al.* 2016). Na figura 4, observa-se a detecção da face com sua respectiva *bounding box*.

Figura 4 - Exemplo de *Bounding Box* da face detectada.

Fonte: imagem do autor (2024).

2.4 MODELOS PRÉ-TREINADOS EM MACHINE LEARNING

Modelos pré-treinados em *machine learning* são algoritmos que já foram treinados em grandes volumes de dados e podem ser reutilizados em novas tarefas, com a possibilidade de adaptação a contextos específicos. Esses modelos economizam tempo e recursos, pois aproveitam o conhecimento previamente adquirido, eliminando a necessidade de treinar redes complexas do zero. Um exemplo comum é o uso de modelos pré-treinados para reconhecimento de objetos, que podem ser ajustados para detectar categorias específicas em novas aplicações (GOODFELLOW, Ian *et al*., 2016).

2.4.1 RetinaFace

Um dos modelos pré-treinados utilizados neste projeto é o RetinaFace, um modelo de detecção de faces baseado em *deep learning*, projetado para realizar detecção facial de alta precisão. Ele utiliza uma abordagem de rede neural convolucional ancorada no framework RetinaNet, o que lhe permite detectar faces em diferentes ângulos e em várias escalas. O modelo é capaz de identificar não apenas a presença de uma face, mas também pontos faciais-chave, como olhos, nariz e boca. O RetinaFace se destaca pela sua precisão em detectar faces em condições desafiadoras, como iluminação variável e oclusões parciais (DENG, Zhong-Qiu *et al*., 2020).

2.4.2 YOLOv8

Outro modelo pré-treinado utilizado é o YOLOv8, amplamente utilizado em tarefas de detecção de objetos em tempo real. Ele introduz melhorias significativas em relação às versões anteriores, como maior precisão e velocidade na detecção, além de uma arquitetura mais otimizada e modular. O YOLOv8 é conhecido por sua eficiência computacional, sendo capaz de realizar predições em uma única passada pela imagem, o que o torna ideal para aplicações em tempo real. Além disso, ele oferece suporte à múltiplas tarefas, incluindo segmentação e detecção de objetos, e foi projetado para ser mais flexível e fácil de ajustar a diferentes domínios (ULTRALYTICS, 2023).

2.4.3 YOLOv10

O YOLOv10 é uma abordagem avançada para detecção de objetos em tempo real, projetada para equilibrar precisão e eficiência. Diferente de suas versões anteriores, ele elimina a necessidade de *Non-Maximum Suppression* (NMS) durante a inferência, um processo que tradicionalmente causava aumento na latência. Para isso, o YOLOv10 introduz um sistema de atribuições duplas consistentes que permite um treinamento sem NMS, resultando em inferências mais rápidas. A arquitetura do modelo é composta por várias inovações, incluindo o *backbone* baseado em uma versão aprimorada do CSPNet, que melhora o fluxo de gradientes e minimiza redundâncias computacionais. O *neck* do YOLOv10 utiliza camadas PAN para combinar informações de diferentes escalas, o que aumenta a precisão na detecção de objetos de tamanhos variados. Além disso, o modelo conta com dois *heads*: o *One-to-Many*, usado no treinamento para gerar várias previsões por objeto e melhorar a aprendizagem, e o *One-to-One*, que produz a melhor previsão única por objeto na inferência, eliminando a necessidade de NMS. Outro diferencial do YOLOv10 é seu design holístico, que otimiza tanto a eficiência quanto a precisão, integrando técnicas como convoluções de kernel grande e módulos de auto atenção parcial, que melhoram o desempenho sem aumentar significativamente o custo computacional. O YOLOv10 também vem em diferentes variantes, desde a versão Nano (YOLOv10-N) para dispositivos de recursos limitados, até a versão *Extra-Large* (YOLOv10-X), voltada para aplicações que exigem máxima precisão e desempenho (ULTRALYTICS, 2023).

Figura 5 - Arquitetura YOLOv10.

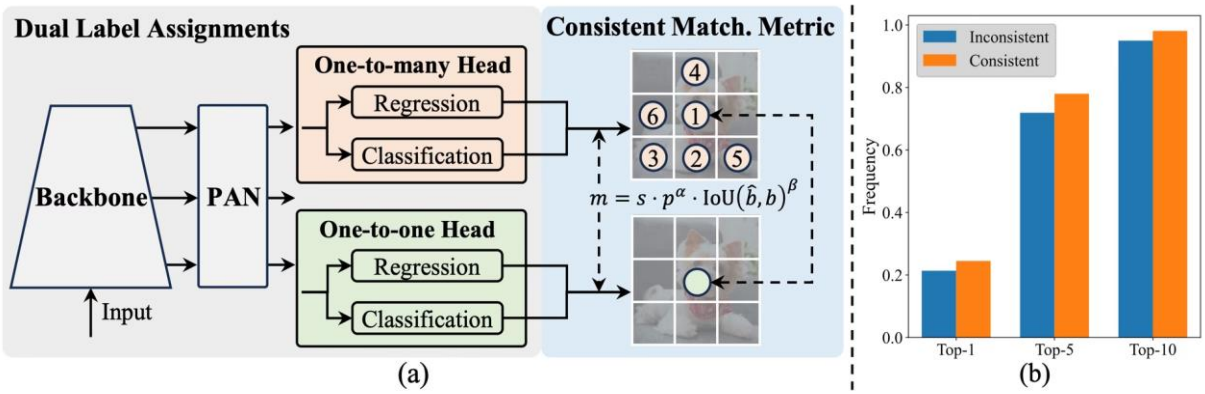


Figure 2: (a) Consistent dual assignments for NMS-free training. (b) Frequency of one-to-one assignments in Top-1/5/10 of one-to-many results for YOLOv8-S which employs α_{o2m}=0.5 and β_{o2m}=6 by default [20]. For consistency, α_{o2o}=0.5; β_{o2o}=6. For inconsistency, α_{o2o}=0.5; β_{o2o} =2.

Fonte: ULTRALYTICS (2023).

3. TRABALHOS RELACIONADOS

Através da ferramenta Google Scholar, foram realizadas pesquisas sobre estudos de detecção de objetos, reconhecimento facial, e reconhecimento de emoções que melhor se relacionam com os objetivos deste trabalho, a fim de dar embasamento para o trabalho proposto. Por fim, foram escolhidos três trabalhos e realizada uma apresentação detalhada dos mesmos. Os trabalhos selecionados foram escolhidos devido à sua relevância direta com o tema do desenvolvimento de modelos de machine learning para detecção de emoções em contextos desafiadores, como durante saltos de paraquedismo. O primeiro trabalho foi selecionado por explorar diferentes técnicas computacionais para classificar emoções com base na teoria do Facial Action Coding System (FACS), uma abordagem que permite compreender as emoções através dos movimentos faciais, algo essencial para o cenário dinâmico do paraquedismo. O segundo trabalho foi escolhido por empregar técnicas modernas de deep learning, como transfer learning e data augmentation, que podem ser altamente eficazes em contextos onde os dados são limitados e as condições são variáveis. Já o terceiro trabalho foi selecionado por demonstrar a aplicação do algoritmo YOLO para reconhecimento facial em tempo real, uma técnica de alta performance e velocidade, alinhada às necessidades de detecção rápida e precisa em atividades dinâmicas como o salto de paraquedas.

No primeiro trabalho, intitulado “Estudo comparativo de técnicas computacionais para classificação de emoções” os autores, Sara L. Melo *et al*. (2014), compararam três técnicas computacionais para a detecção de seis emoções básicas (alegria, tristeza, raiva, desgosto, medo e surpresa), utilizando a teoria do *Facial Action Coding System* (FACS), que descreve os movimentos faciais visíveis como Unidades de Ação (UA’s). Utilizando o banco de dados de Kanade e Cohn (2005), foram aplicadas Redes Neurais *Multi-Layer Perceptron* (MLP), Redes *Basis Functions Networks* (RBFN) e Redes Bayesianas para classificar as emoções.

No segundo trabalho, intitulado “*Facial expression recognition with deep learning*”, os autores Amil Kahnzada, Charles Bai e Turker Ferhat (2020), buscaram melhorar a performance de modelos de reconhecimento de emoções, utilizando técnicas como *transfer learning*, *data augmentation* e *class weighting*. Os autores compararam diferentes abordagens usando conjuntos de dados como FER2013, Cohn-Kanade e JAFFE, além de um dataset próprio. O modelo base foi uma rede CNN simples com camadas convolucionais, *max-pooling*, normalização por lotes e *dropout*.

Por fim, no terceiro trabalho, intitulado “*A Deep Learning Approach for Face Detection using YOLO*”, os autores, Dweepna Garg *et al*. (2018), focaram no desenvolvimento de um modelo de *deep learning* para reconhecimento facial utilizando o algoritmo YOLO. Os autores optaram pelo YOLO devido à sua capacidade de detectar objetos em tempo real, diferente das abordagens tradicionais baseadas em redes neurais convolucionais (CNNs) e Fast R-CNN.

Na última parte desta seção, serão apresentados outros trabalhos que são importantes para a compreensão aprofundada do presente trabalho e das metodologias abordadas.

3.1 ESTUDO COMPARATIVO DE TÉCNICAS COMPUTATIONAIS PARA CLASSIFICAÇÃO DE EMOÇÕES

Neste trabalho, os autores Sara L. Melo *et al*. (2014), apresentam um estudo comparativo entre três técnicas computacionais usadas para detecção de seis emoções básicas (alegria, tristeza, raiva, desgosto, medo e surpresa), aliado à teoria de Facial *Action Coding System* (FACS). A teoria FACS descreve todos os possíveis movimentos faciais visíveis, denominados de Unidades de Ação (UA’s) do indivíduo, onde cada UA está relacionada com a contração de seus músculos e, também, a cada

movimento da face. Através das análises das técnicas escolhidas, os autores verificaram a mais adequada na classificação de emoções de um estudante durante seu processo de estudo, para que no futuro pudesse ser incorporada a um Sistema Adaptativo (SA), o qual, baseado em fatores cognitivo-emocionais, oferece o melhor Objeto de Aprendizagem (OA).

Para o desenvolvimento do trabalho, os autores optaram pelo Banco de Dados desenvolvido por Kanade and Cohn (2005), por apresentarem informações de pessoas de várias etnias, sexo e idade. Além disso, os dados contêm um conjunto de UA's, na qual foi realizada uma etapa de mineração de dados que consistiu em relacionar cada agrupamento de UA's a cada uma das seis emoções. Posteriormente, foram utilizados três métodos computacionais e foi observado como cada técnica faria a classificação das emoções. As três técnicas computacionais que os autores utilizaram foram: Redes Neurais Artificiais do tipo *Multi-Layer Perceptron* (MLP), com a utilização do algoritmo de treinamento *Backpropagation*, Rede *Basis Functions Networks* (RBFN) e Redes Bayesianas.

Para a classificação das emoções utilizando Redes Neurais MLP, os autores arquitetaram a rede com a utilização do algoritmo *backpropagation* como algoritmo de aprendizado; sete neurônios na camada de entrada correspondente à combinação de UA's na base de dados; uma camada oculta com seis neurônios e, ainda, uma camada de saída contentando seis neurônios (emoções básicas). Os autores obtiveram uma taxa de precisão de 75,72%, onde 287 instâncias foram classificadas corretamente e 92 instâncias foram detectadas como falsos positivos. Pode-se observar os resultados da rede neural (MLP) na matriz de confusão na figura 5.

Figura 6 - Resultados obtidos (MLP).

TI	Alegria	Tristeza	Desgosto	Surpresa	Raiva	Medo
101	93	4	2	0	1	1
72	0	60	3	3	3	3
45	6	0	27	0	11	1
77	0	5	0	66	1	5
36	1	1	9	0	25	0
48	2	8	4	10	8	16

Fonte: DE MELO, SARA L. *et al*. (2014).

Para a rede neural com Função de Ativação de Base Radial (RBF), que possui apenas uma única camada oculta – cujos neurônios possuem função de ativação gaussiana, em vez de sigmoidal –, os autores obtiveram uma taxa de precisão de 71,76% na classificação das emoções, onde 272 instâncias foram classificadas corretamente e 107 instâncias foram detectadas como falsos positivos. Pode-se observar na figura 6 os resultados da rede neural (RBF) na matriz de confusão.

Figura 7 - Resultados obtidos (RBF).

TI	Alegria	Tristeza	Desgosto	Surpresa	Raiva	Medo
101	91	3	4	0	1	2
72	22	40	1	1	2	6
45	7	0	29	0	7	2
77	0	2	0	73	0	2
36	5	2	9	1	16	3
48	2	4	6	9	4	23

Fonte: DE MELO, SARA L. *et al*. (2014).

Para o último método, os autores utilizaram Redes Bayesianas, que oferecem uma estrutura intuitiva de representar o raciocínio incerto. A vantagem de sua utilização concentra-se no sentido de permitir a representação e manipulação da incerteza com base em princípios matemáticos fundamentados. Com a utilização das Redes Bayesianas, os autores obtiveram uma taxa de 86% de precisão, onde 326 instâncias foram classificadas corretamente e apenas 53 instâncias foram classificadas como falsos positivos. Pode-se observar na figura 7 os resultados das Redes Bayesianas na matriz de confusão na figura.

Figura 8 - Resultados obtidos (Redes Bayesianas).

TI	Alegria	Tristeza	Desgosto	Surpresa	Raiva	Medo
101	97	2	0	0	0	2
72	0	63	1	1	2	5
45	2	0	36	0	7	0
77	2	0	0	72	0	3
36	0	2	9	0	23	2
48	6	0	1	2	4	35

Fonte: DE MELO, SARA L. *et al*. (2014).

3.2 *FACIAL EXPRESSION RECOGNITION WITH DEEP LEARNING*

O estudo de referência teve por objetivo entender e aprimorar a performance de modelos de reconhecimento de emoções. Para isso, os autores Amil Kahnzada, Charles Bai e Turker Ferhat (2020), utilizaram várias abordagens de publicações recentes de outros estudos como, *transfer learning*, *data augmentation* e *class weighting*. Também foi desenvolvido um aplicativo *web* para executar os modelos em dispositivos.

Os autores destacam que reconhecer expressões básicas em condições controladas como fotos frontais e expressões faciais forçadas é um problema resolvido, no qual esses modelos atingem uma acurácia de 98,90%. Porém, distinguir essas expressões em condições naturais é um problema desafiador devido a variações na pose da cabeça e iluminação. Com o avanço do *deep learning*, a tecnologia de reconhecimento de expressões faciais em ambientes naturais permitiu inovações em robótica, medicina, e sistemas de interação humano-computador como já mencionado anteriormente.

Para criação dos modelos, os autores utilizaram três principais conjuntos de dados, nos quais podemos destacar o FER2013 como o principal, Cohn-Kanade e JAFFE como conjunto de dados auxiliares. Também foi utilizado um conjunto de dados próprio dos autores, para ajudar no funcionamento do aplicativo em ambientes reais. O conjunto de dados FER2013 é um dos principais conjuntos de dados quando se trata de expressões faciais. O *dataset* é composto por 35.887 imagens normalizadas em 48x48 pixels e possui sete expressões faciais, sendo elas: raiva, nojo, medo, feliz, triste, surpreso e neutro.

Como mencionado, foram abordados diversos modelos para verificar qual teria o melhor desempenho na classificação das emoções. O modelo base foi caracterizado por uma rede neural CNN simples, composta por quatro camadas convolucionais com filtros ReLU de 3x3x32, todas com preenchimento igual. Essas camadas convolucionais foram intercaladas com duas camadas de *Max Pooling* de 2x2, que reduziram a dimensionalidade dos mapas de características. Em seguida, foi adicionada uma camada totalmente conectada (FC) e, por fim, uma camada *softmax* para a classificação das emoções. Além disso, foram incluídas camadas de normalização de lotes (*batch normalization*) e *dropout* de 50% para controlar a variabilidade e melhorar a precisão do modelo – que foi aprimorado de 53,0% para 64,0%. Os autores também exploraram modelos pré-treinados, como ResNet50, SeNet50 e VGG16. A figura 8 apresenta os resultados obtidos pelos autores em relação aos modelos propostos.

Figura 9 - Resultados obtidos pelos autores.

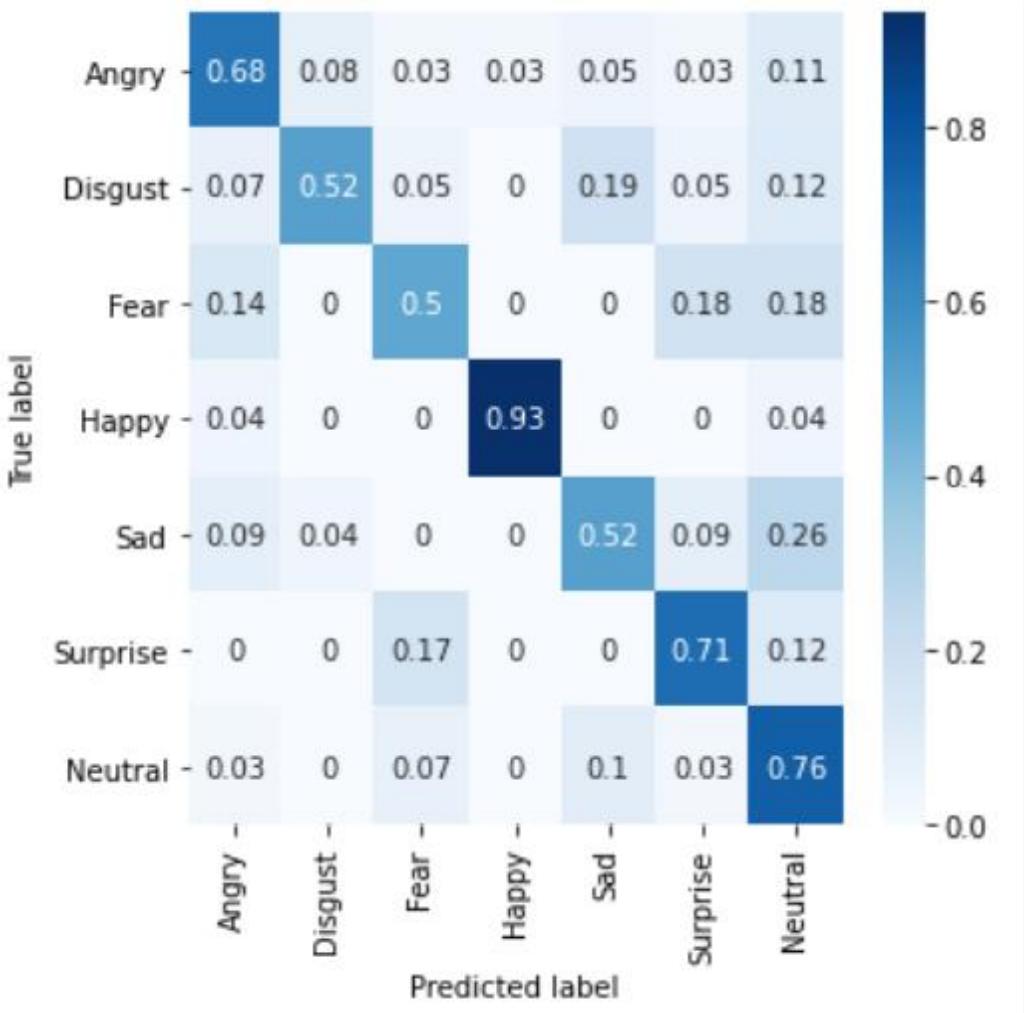
Model	Depth	Parameters	Accuracy
(Human-level)	-	-	65±5%
Tang [4]	4	12(m)	71.2%
Pramerdorfer et al. [2]	10/16/33	1.8/1.2/5.3(m)	75.2%
Baseline	5	37.8(m)	64.0%
Five-Layer Model	5	2.4(m)	66.3%
VGG16	16	138(m)	70.2%
SeNet50	50	27(m)	72.5%
ResNet50	50	25(m)	73.2%
Ensemble	-	-	**75.8%**

Fonte: KHANZADA, Amil *et al*. (2020).

Por fim, a arquitetura do modelo para o aplicativo *web* se caracterizou por três estágios de camadas convolucionais e camadas de *max-pooling*, seguidos por uma camada totalmente conectada (FC), com 1.024 neurônios, e uma camada de saída *softmax*. As camadas convolucionais foram configuradas, respectivamente, com 32, 32 e 64 filtros, e com tamanhos de 5x5, 4x4 e 5x5. Para ativação, foi utilizada a função ReLU em todas as camadas. Com o objetivo de melhorar o desempenho, foram adicionadas camadas de normalização por lote (*batch normalization*). O treinamento

do modelo foi realizado ao longo de 300 épocas, utilizando a função de perda de entropia cruzada. A figura B expõe os resultados obtidos do modelo proposto, alcançando 69,80% de acurácia.

Figura 10 - Resultados modelo aplicativo para web.



Fonte: KHANZADA, Amil *et al*. (2020).

3.3 *A DEEP LEARNING APPROACCH FOR FACE DETECTION USING YOLO*

O estudo de referência se concentra na criação de um modelo de *deep learning* para reconhecimento facial, utilizando o algoritmo YOLO. Os autores, Dweepna Garg *et al*. (2018), antes de apresentar a sua proposta, fazem uma contextualização sobre a evolução das abordagens de reconhecimento facial, no qual as redes neurais convolucionais (CNNs) e a Fast R-CNN alcançam ótimos resultados quando tratamos

de detecção de objetos. Para o modelo proposto, os autores optaram por uma abordagem diferente, com a utilização de uma arquitetura baseada na estrutura YOLO, que utiliza CNNs para detectar rostos em tempo real.

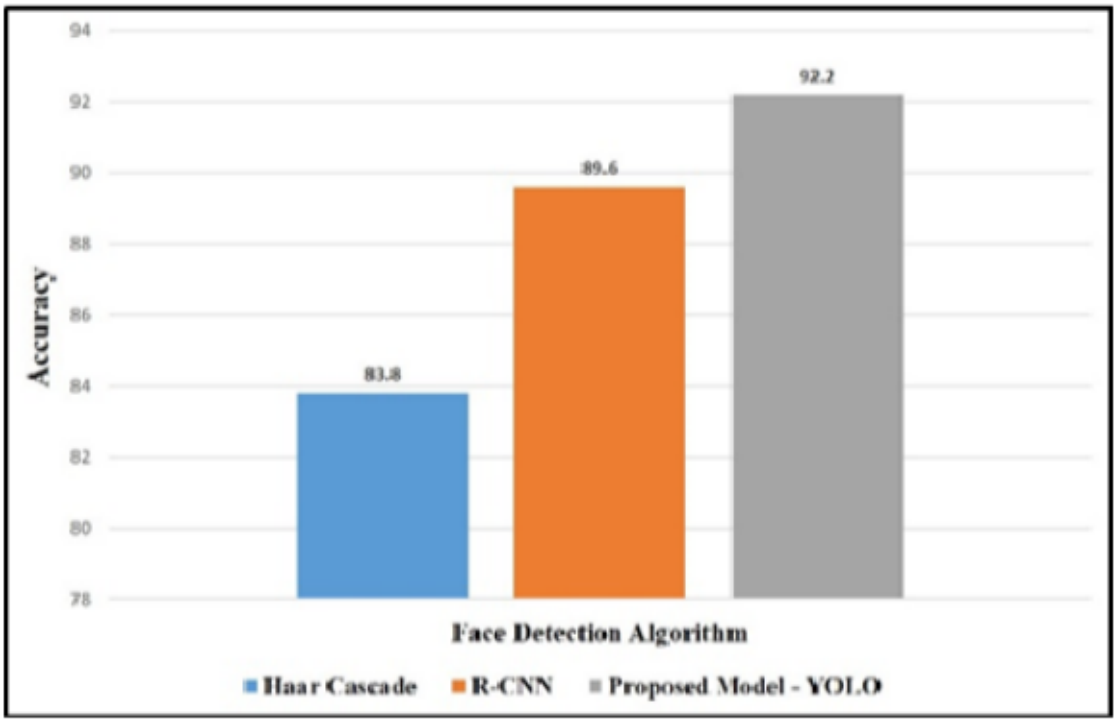
O modelo pode ser executado em diferentes resoluções, proporcionando boa velocidade e precisão. As imagens também podem ser redimensionadas aleatoriamente, permitindo ao detector aprender características de diversos tamanhos de imagem. Apesar de ser eficaz e rápido, o algoritmo também enfrenta limitações com pequenos conjuntos de dados. A detecção de objetos com YOLO envolve a previsão paralela de *bounding boxes* e classes de objetos, codificando informações de aparência e categorias.

O algoritmo YOLO divide a imagem de entrada em uma grade S x S e atribui a cada célula a tarefa de detectar objetos cujo centro cai dentro dela. Cada célula prevê a caixa delimitadora e a pontuação de confiança, que indica a precisão da detecção baseada na interseção sobre união entre a caixa prevista e a verdade fundamental. As previsões incluem as coordenadas do centro da caixa (x, y), a altura (h), a largura (w) e a confiança. As probabilidades de classe condicional são multiplicadas pela confiança da caixa para fornecer a pontuação final.

O modelo proposto pelos autores consiste em uma imagem colorida de tamanho 448x448 pixels como entrada da rede. A arquitetura consiste em sete camadas convolucionais seguidas por uma camada de *max pooling* de tamanho 2x2. Em seguida, três camadas totalmente conectadas são adicionadas, e a camada de saída vem após a última camada totalmente conectada. As camadas convolucionais encontram características simples a complexas nas imagens, e a camada totalmente conectada prevê as coordenadas e probabilidades. Finalmente, a camada de saída prevê tanto as probabilidades das classes quanto as coordenadas da caixa delimitadora usando a técnica de supressão não máxima (NMS).

O banco de dados utilizado foi o FDDB, que consiste em 2.845 imagens. O *dataset* foi dividido em 70% para treinamento e 30% para o teste. O modelo foi treinado com 25 épocas, obtendo uma acurácia de 92,2%. Por fim, na figura 11, os autores fizeram uma comparação com outros modelos de reconhecimento facial.

Figura 11 - Comparação de modelos de reconhecimento facial.



Fonte: GARG, Dweepna *et al.* (2018).

3.4 OUTROS TRABALHOS

Outros trabalhos, além dos já mencionados, foram analisados para o desenvolvimento do presente trabalho. Considerando suas relevâncias, serão apresentados de forma resumida, com o objetivo de contribuir ainda mais para o desenvolvimento do presente estudo.

Rajesh e Naveenkumar (2016) – no trabalho "*A robust method for face recognition and face emotion detection system using support vector machines*" – abordam um método de reconhecimento facial e de emoções utilizando o método *Support Vector Machines* (SVM). O SVM é utilizado para achar diferentes emoções de faces e, também, para classificá-las. Também é utilizado *Principal Component Analysis* (PCA) para reduzir as dimensões das imagens e extrair *features* das faces como olhos, nariz, lábios e contorno da face. Para a classificação das emoções, o algoritmo Multi SVM foi utilizado. Foram utilizadas imagens de dimensão 640x480 pixels e o banco de dados utilizado foi o Cohn-Kanade e Extended Cohn-Kanade.

Awais Shaikh *et al.* (2023) apresentam um estudo de detecção de emoções faciais de imagens utilizando modelos YOLO – intitulado "Comprehensive Study on Emotion Detection with Facial Expression Images Using YOLO Models". Os autores começaram o estudo apresentando as versões do algoritmo YOLO e suas características, assim como o processo de classificação e localização das faces. A

arquitetura proposta pelos autores consiste no modelo YOLOv5, que foi estruturado com três componentes principais: as camadas YOLO, responsáveis pela detecção das emoções nas imagens faciais; o pescoço, utilizando a PANet Head para a fusão de recursos, combinando informações de diferentes camadas da rede para aprimorar a precisão da detecção; e o *backbone*, baseado na arquitetura CSP-Darknet, que realiza a extração de recursos, capturando características essenciais das imagens. Essa combinação de componentes permite que o modelo YOLOv5 realize a detecção de emoções de forma eficiente e precisa, mesmo com entradas de baixa resolução, como as imagens de 48x48 *pixels* utilizados no estudo. O banco de dados utilizado foi o FER2013, ambos para treinamento e teste, atingindo uma acurácia de 50%.

Riyantoko, Sugiarto e Hindrayani (2021) realizaram um estudo – intitulado "*Facial emotion detection using Haar-cascade classifier and convolutional neural networks*" – que propõe classificar a emoção facial utilizando o Haar-Cascade Classifier e Redes Neurais Convolucionais (CNN). O experimento utilizou o conjunto de dados FER2013, que foi coletado para reconhecimento de expressão facial, propondo a classificação de sete expressões faciais. A CNN utilizada no estudo possui seis camadas convolucionais, duas camadas de subamostragem, doze camadas de convolução e duas redes neurais de subamostragem. O modelo obteve resultados de erro quadrático médio (MSE) e precisão com base em uma variedade de épocas. Os resultados mostraram que, com o aumento do número de épocas, o valor do MSE diminuiu, enquanto a precisão aumentou. Assim, o algoritmo da CNN provou ser eficaz para a detecção de emoções faciais.

3.5 ANÁLISE COMPARATIVA

A tabela 1 faz uma comparação entre os trabalhos apresentados em relação ao tipo de abordagem utilizada, as métricas utilizadas, o tipo de arquitetura do modelo e por fim o conjunto de dados utilizados.

Tabela 1 - Comparação dos tipos de abordagem, métricas, arquitetura e conjunto de dados.

Autor	**Métrica utilizada**	**Arquitetura**	**Conjunto de dados**

DE MELO, Sara L. *et al.* (2014)	Precisão	MLP, RBF, Redes Bayesianas	Cohn-Kanade AU-Coded
KHANZADA, Amil; BAI, Charles; CELEPCIKAY, Ferhat T. (2020)	Acurácia, Precisão	CNN, ResNet50, SeNet50, VGG16	FER2013, Cohn-Kanade, JAFFE
GARG, Dweepna *et al.* (2018)	Acurácia	CNN, YOLO	FDDB
RAJESH, K. M; NAVEEN KUMAR, M. (2016)	Acurácia	SVM, MultiSVM	Cohn-Kanade, Extended Cohn-Kanade
SHAIKH, Awais *et al*. (2023)	Acurácia	YOLO v5	FER2013
RIYANTOKO, Prismahardi Aji SUGIARTO, K.M; HINDRAYAINI, Kartika Maulida (2021)	Acurácia, Precisão	CNN	FER2013
ANTUNES, Vinicius (2024)	Precisão, mAP50, Recall, F1-score	YOLOv8, YOLOv10	Conjunto de dados próprio

Fonte: elaboração própria (2024).

Pode-se perceber que é possível desenvolver diversos modelos de detecções com diversos tipos de abordagens. Quanto aos conjuntos de dados utilizados, existe uma concordância entre os estudos analisados, destacando o conjunto FER2013 e Cohn-Kanade. Apesar do presente trabalho ser desenvolvido com o próprio conjunto de dados, é importante conhecer outros conjuntos nos quais as expressões faciais foram amplamente estudadas.

Em relação às métricas de avaliação dos modelos, todos os trabalhos adotaram acurácia e precisão como forma de avaliação. Para este estudo, essas métricas também serão importantes para a avaliação do modelo.

4. DESENVOLVIMENTO DE MODELOS DE *DEEP LEARNING*

Conforme descrito na Seção 1, subseção 1.2 (Objetivos específicos), o presente trabalho visa o desenvolvimento de modelos de *deep learning* que sejam capazes de identificar e classificar expressões faciais em imagens capturadas, tanto durante saltos de paraquedismo quanto momentos antes do salto. Os dados de entrada para o treinamento e construção do modelo serão de imagens capturadas através de vídeos de pessoas saltando e de dentro da aeronave. Para avaliação dos modelos, será adotado um conjunto de dados não utilizado no treinamento, assegurando a validade e a eficácia das técnicas aplicadas.

Para cumprir o objetivo, o desenvolvimento do modelo proposto segue um fluxo composto por seis principais etapas, sendo elas: coleta dos dados, preparação dos dados, pré-processamento dos dados, anotação dos dados, treinamento dos modelos e, por fim, avaliação e validação de cada modelo. Verifica-se a seguir o fluxograma da solução (Figura 12).

Figura 12 - Fluxograma do desenvolvimento da proposta.

Fonte: elaboração própria (2024).

Desse modo, a primeira etapa consiste na coleta de dados, que serão provenientes de vídeos de saltos; já na segunda etapa será realizada uma preparação

dos dados coletados para posterior processamento dos mesmos na terceira etapa; na quarta etapa, será realizada a anotação dos dados, utilizando um modelo pré-treinado para detecção de faces em imagens. Com os conjuntos de teste e validação prontos, será realizado o treinamento dos modelos YOLO para a detecção das emoções, configurando a etapa cinco. Por fim, na etapa seis, os modelos serão avaliados.

4.1 FERRAMENTAS

Para o desenvolvimento deste trabalho, utilizou-se o *Google Colab*, uma ferramenta que permite a execução de códigos em *Python* diretamente no navegador, facilitando o acesso a recursos computacionais robustos, como GPUs, sem a necessidade de configurações complexas. O *Google Colab* é amplamente utilizado em projetos de *machine learning* devido à sua integração com o *Google Drive*, possibilitando o armazenamento e o acesso fácil aos dados e resultados diretamente na nuvem.

A linguagem *Python*, por sua vez, foi escolhida por ser uma das mais utilizadas na área de inteligência artificial e ciência de dados, graças à sua simplicidade, extensa biblioteca de pacotes especializados, como NumPy, pandas e OpenCV, que agilizam a implementação e experimentação de modelos complexos, como o YOLO e o RetinaFace utilizados neste projeto.

4.2 COLETA E PREPARAÇÃO DE DADOS

A coleta de dados para este trabalho envolveu a reunião de uma série de vídeos de saltos de paraquedismo, que serviram como a principal fonte para a análise das expressões faciais durante o salto. Esses vídeos capturaram os paraquedistas em diferentes momentos do salto, oferecendo expressões faciais em diferentes condições de iluminação, ângulos e distâncias da câmera.

Para garantir que os dados fossem adequados para o treinamento dos modelos, utilizou-se um script em *Python* para a extração de frames específicos[1]. Esse processo não foi automatizado, pois envolvia a seleção de frames em que as faces dos indivíduos apareciam de forma clara e frontal, com boa proximidade da câmera e sob condições adequadas de iluminação. A identificação desses frames foi realizada manualmente, observando-se os instantes dos vídeos em que as faces atendiam aos

[1] Notebook para extração dos frames dos vídeos. Disponível em: https://colab.research.google.com/drive/1jih1bmy11W8V5MZlSlj3vB97u9cI2Qm-?usp=sharing

critérios. Em seguida, esses instantes foram indicados no script, que se encarregava de realizar a extração dos frames desejados. Cada vídeo foi analisado individualmente, para garantir que as expressões faciais capturadas fossem suficientemente nítidas para a análise posterior. Todos os frames selecionados foram então organizados e armazenados no *Google Drive*, para posterior uso nas fases de anotação e treinamento dos modelos.

Foi construído um *dataset*[2] específico para este trabalho – disponibilizado no *GitHub* para facilitar o acesso e replicação dos experimentos. Apresenta-se abaixo a tabela relacionando a quantidade de vídeos e de frames selecionados, e alguns exemplos de imagens do conjunto, conforme a Figura 13.

Tabela 2 - Quantidade de vídeos analisados e frames selecionados.

Quantidade de Vídeos	**Quantidade de Frames**
33	3.054

Fonte: elaboração própria (2024).

Figura 13 - Exemplo de imagens do conjunto.

(A)	**(B)**	**(C)**

Fonte: elaboração própria (2024).
Legenda: (A) representa a classe medo, (B) representa a classe neutro e (C) representa a classe feliz.

4.3 ANOTAÇÃO DE DADOS

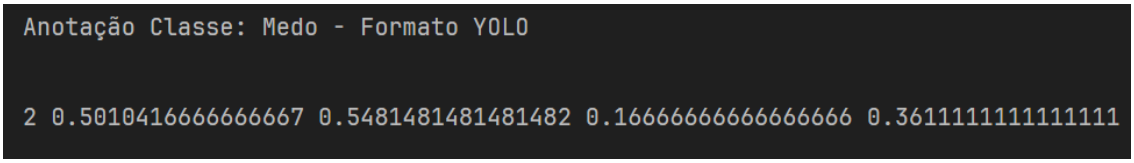
Durante o processo de seleção, os frames já foram previamente classificados nas três emoções propostas para o desenvolvimento do trabalho: medo, neutro e feliz. Para facilitar o processo de anotação, os frames foram organizados em três diretórios distintos, correspondendo a cada uma dessas emoções.

[2] Link do GitHub para o conjunto de dados. Disponível em: https://github.com/vin1tunes/TCC-VINI

Após a etapa de classificação, através de um script *Python* executado no *Google Colab*[3], o algoritmo pré-treinado RetinaFace (DENG, Jiankang et al., 2020), foi utilizado para detectar as faces presentes nos frames, gerando *bounding boxes* que delimitavam as áreas de interesse das faces e, assim, automatizando as anotações. As anotações das faces foram realizadas no formato YOLO, que exige que cada anotação contenha um identificador da classe correspondente, além das coordenadas das *bounding boxes*.

Como os frames já haviam sido separados por classes previamente, o identificador da classe foi inserido manualmente durante a iteração do algoritmo, garantindo que cada face fosse anotada corretamente de acordo com a emoção previamente atribuída ao frame. Esse processo de anotação manual, associado às *bounding boxes* geradas automaticamente, permitiu que as faces detectadas fossem associadas às suas respectivas classes emocionais, gerando os dados necessários para o treinamento do modelo. As figuras a seguir apresentam exemplos das anotações geradas.

Figura 14 - Anotação para a classe “medo”.



[3] Notebook para realização das anotações das imagens. Disponível em: https://colab.research.google.com/drive/1yCbIBINhBKkOpopRVRaXAeMfH0CzALfB?usp=sharing

Fonte: elaboração própria (2024).

Figura 15 - Anotação para a classe “neutro”.

```
Anotação Classe: Neutro - Formato YOLO

1 0.6385416666666667 0.4583333333333333 0.12604166666666666 0.26666666666666666
```

Fonte: elaboração própria (2024).

Figura 16 - Anotação para a classe "feliz".

```
Anotação Classe: Feliz - Formato YOLO

0 0.47486979166666665 0.5915178571428571 0.08776041666666666 0.10863095238095238
```

Fonte: elaboração própria (2024).

Cada anotação segue uma estrutura específica que contém informações sobre a classe do objeto detectado e as coordenadas da *bounding box* que delimita esse objeto na imagem. No desenvolvimento do presente trabalho, foram utilizadas três classes: 0 para a classe "feliz", 1 para a classe "neutro" e 2 para a classe "medo".

A primeira entrada de cada anotação indica o identificador da classe correspondente, que neste caso pode ser 0, 1 ou 2, dependendo da emoção associada ao frame. Os valores seguintes se referem às coordenadas da *bounding box* no formato normalizado, com as coordenadas do centro da *bounding box* e a largura e altura também normalizadas em relação ao tamanho total da imagem.

Figura 17 - Exemplo de *bounding box* da face detectada pelo algoritmo RetinaFace.

Fonte: imagem do autor (2024).

Figura 18 - Exemplo de *bounding box* da face detectada pelo algoritmo RetinaFace.

Fonte: imagem do autor (2024).

Figura 19 - Exemplo de bounding box da face detectada pelo algoritmo RetinaFace.

Fonte: imagem do autor (2024).

Ao realizar as anotações das faces utilizando o algoritmo RetinaFace, foi observado que algumas faces não foram detectadas. Isso ocorreu principalmente devido à má qualidade de imagem e às posições desfavoráveis das faces, como aquelas que estavam mais afastadas ou em ângulos pouco visíveis. Consequentemente, o número total de imagens anotadas reduziu-se em comparação ao conjunto de dados original. Essa diminuição impactou o tamanho do conjunto de treinamento para cada uma das classes de expressões faciais. A seguir, detalha-se em tabela a quantidade de anotações efetuadas por classe.

Tabela 3 - Quantidade total de anotações por classe.

Conjunto de dados	
Classe	**Total de Anotações**
Feliz	1.110
Neutro	1.158
Medo	517

Fonte: elaboração própria (2024).

4.4 EXECUÇÃO DO TREINAMENTO

O treinamento foi realizado utilizando os modelos YOLOv8n, YOLOv8s, YOLOv10n e YOLOv10s, com o objetivo de permitir que o modelo generalizasse as características das classes presentes no conjunto de dados. As versões "nano" e "s" se diferenciam principalmente em relação ao tamanho e complexidade. Os modelos nano são menores, mais leves e rápidos, indicados para aplicações com restrições de hardware e tempo de inferência, enquanto os modelos “s” são mais robustos, com maior capacidade de aprendizado, proporcionando melhor desempenho em cenários que permitem mais poder computacional.

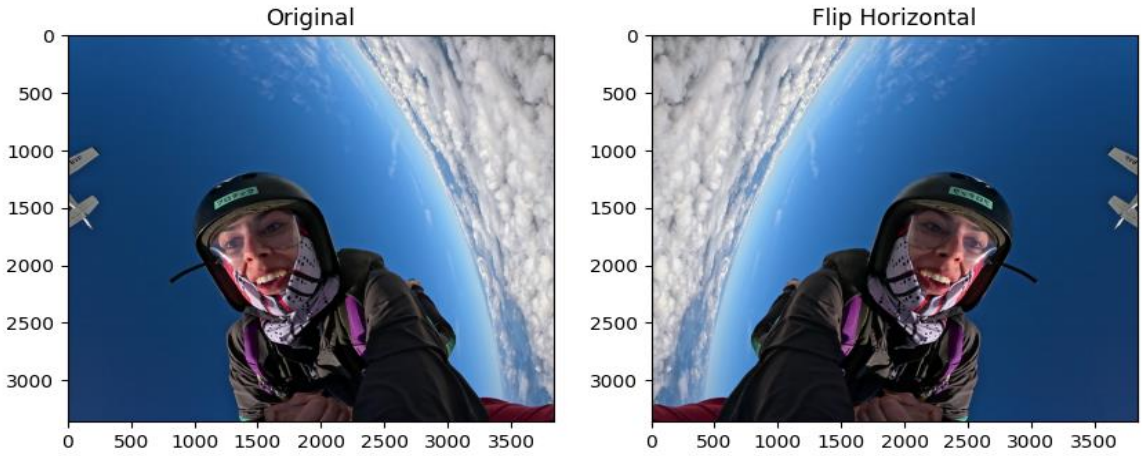
Para otimizar o processo de treinamento, foram utilizadas técnicas como o aumento de dados (*data augmentation*), o otimizador *Stochastic Gradient Descent* (SGD) e o critério de *early stop*. O aumento de dados, ou *data augmentation,* visa gerar mais variações nas imagens de treinamento, melhorando a capacidade de generalização do modelo. Isso é alcançado por meio de técnicas que incluem transformações – como rotação, espelhamento, ajustes de brilho e contraste, zoom e recortes aleatórios –, o que permite que o modelo aprenda a reconhecer objetos em diferentes condições e perspectivas. O otimizador SGD ajusta os pesos da rede de forma incremental, atualizando-os com base em cada lote de dados, contribuindo para

uma convergência mais rápida. Já o *early stop* interrompe o treinamento assim que não há mais melhoria significativa no desempenho do modelo, evitando *overfitting* (ULTRALYTICS, 2023).

Para avaliar o desempenho ao longo do processo, foram analisadas as métricas de mAP50, precisão e *recall*, garantindo a monitorização da evolução do aprendizado e dos resultados finais. Além disso, realizou-se uma análise do comportamento dessas métricas, bem como das métricas relacionadas ao *loss*, ao longo das épocas de treinamento. O monitoramento ao longo das épocas foi essencial para a identificação de possíveis casos de *overfitting*. Ao todo, foram realizados seis treinamentos: o primeiro com um conjunto reduzido de dados para uma análise inicial do desempenho; outros quatro treinamentos para cada tipo de arquitetura mencionado anteriormente, utilizando o conjunto completo de dados; e o último treinamento teve por objetivo buscar a melhora no desempenho. No total, foram utilizadas 2.785 imagens, sendo 1.973 destinadas ao treinamento, 463 à validação e 349 ao teste.

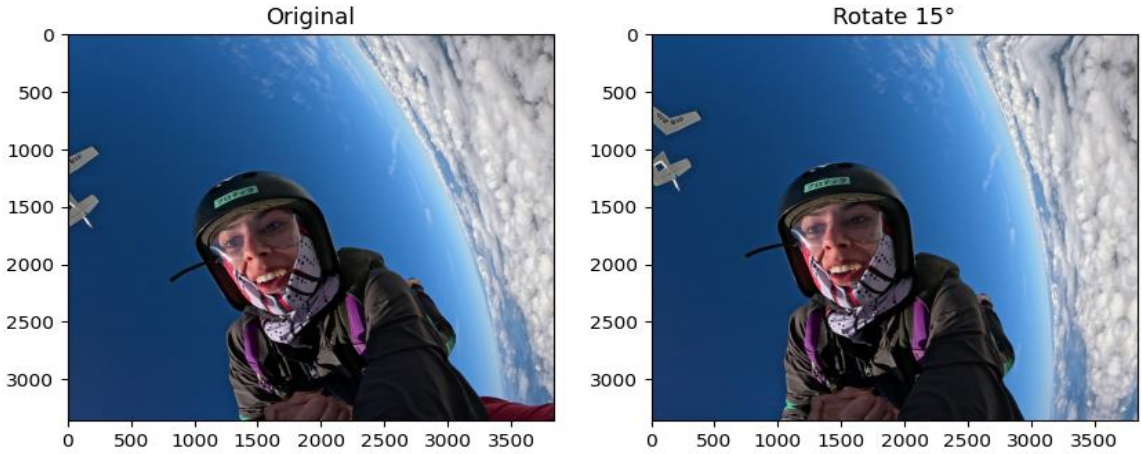
Durante o treinamento, a técnica de aumento de dados foi implementada com várias transformações aplicadas aleatoriamente a cada imagem, visando melhorar a generalização do modelo. As características usadas para o aumento de dados incluíram: rotação de 15 e -15 graus, *shear* de 15 graus, ajustes de matiz, saturação e brilho (matiz de 20 graus, saturação de 25% e brilho de 25%), aplicação de *blur* de 1px e adição de ruído gaussiano com variação de até 2%. Essas transformações tiveram 50% de chance de serem aplicadas, proporcionando maior diversidade ao conjunto de dados e tornando o modelo mais robusto a variações reais. A seguir, são apresentados exemplos de imagens com as transformações aplicadas, acompanhados de uma breve explicação sobre o uso de cada conjunto de dados durante o treinamento dos modelos propostos.

Figura 20 - aumento de dados (*flip* horizontal).



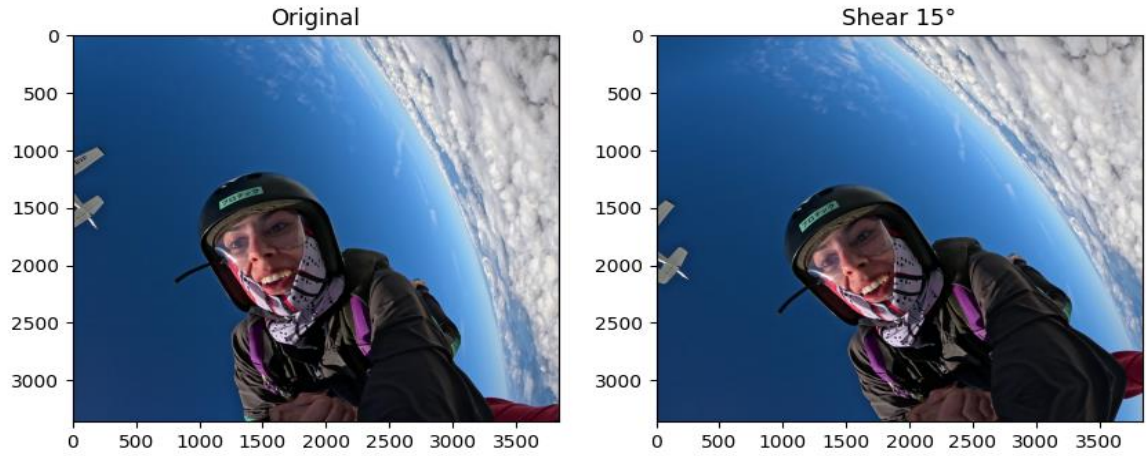
Fonte: imagem do autor.

Figura 21 - Aumento de dados (rotação 15°).



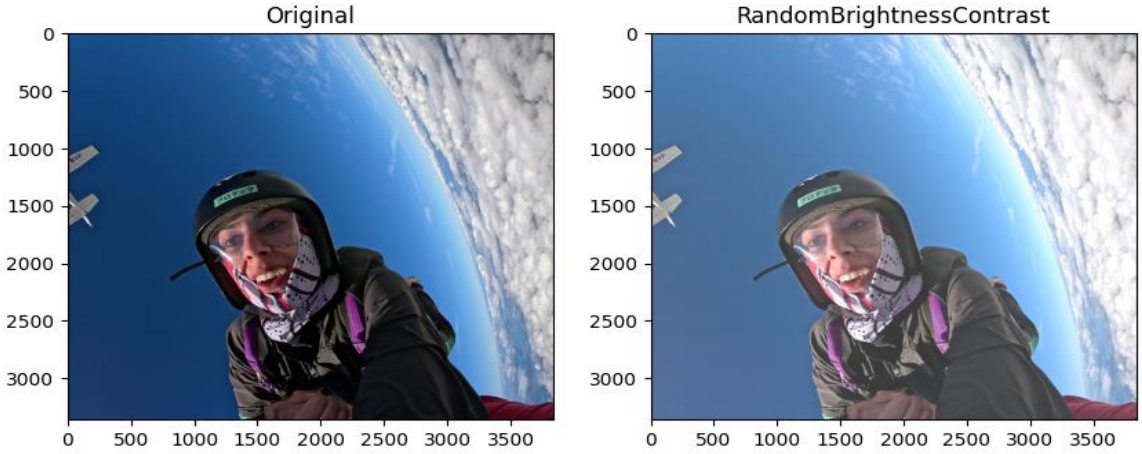
Fonte: imagem do autor (2024).

Figura 22 - Aumento de dados (*shear* 15°).



Fonte: imagem do autor (2024).

Figura 23 - Aumento de dados (*brightness*).



Fonte: imagem do autor.

O conjunto de dados de treinamento é composto por imagens que o modelo utiliza para aprender a identificar padrões e características que correspondem às diferentes expressões faciais – como feliz, neutro e medo. Portanto, o conjunto de dados de treinamento foi utilizado na etapa inicial e principal do treinamento, durante o processo de ajuste dos pesos e parâmetros do modelo.

Tabela 4 - Quantidade total de anotações por classe (treinamento).

Conjunto de dados (Treinamento)	
Classe	**Total de Anotações**
Feliz	800
Neutro	800

Medo	373

Fonte: elaboração própria (2024).

O conjunto de dados de validação é composto por imagens que não são usadas diretamente no ajuste dos pesos do modelo, mas sim para avaliar sua capacidade de generalização enquanto ele ainda está em treinamento. Desse modo, o conjunto de dados de validação foi utilizado ao final de cada época durante o treinamento do modelo, para medir o desempenho.

Tabela 5 - Quantidade total de anotações por classe (validação).

Conjunto de dados (Validação)	
Classe	**Total de Anotações**
Feliz	178
Neutro	210
Medo	75

Fonte: elaboração própria (2024).

O conjunto de dados de teste é composto por imagens que o modelo nunca viu durante o treinamento ou validação, servindo como um conjunto independente para avaliar sua performance em condições reais. Portanto, o conjunto de dados de teste foi utilizado após a conclusão do treinamento, fornecendo uma avaliação definitiva do desempenho do modelo.

Tabela 6 - Quantidade total de anotações por classe (teste).

Conjunto de dados (Teste)	
Classe	**Total de Anotações**
Feliz	132
Neutro	148
Medo	69

Fonte: elaboração própria (2024).

Na subseção seguinte, apresenta-se a relação dos treinamentos realizados, assim como as características de cada treinamento bem como as métricas alcançadas.

4.4.1 Modelo YOLOv8n (dados reduzidos)

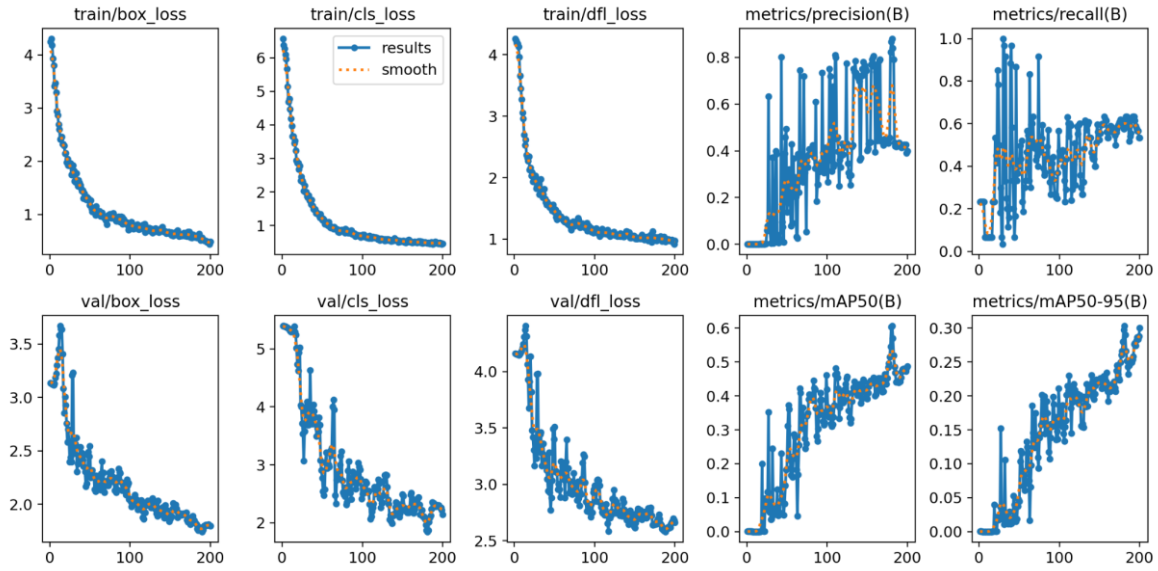
Com base no treinamento realizado com um conjunto de dados reduzido de 129 imagens, sendo 105 destinadas ao treinamento e 24 para validação, o modelo apresentou resultados que refletem seu desempenho limitado devido à quantidade de dados. O valor de mAP50 de 0,4869 indica que o modelo foi capaz de prever corretamente cerca de 48% das caixas delimitadoras, sugerindo que ele está aprendendo as características das classes, mas ainda possui margem para melhorias. A precisão de 0,3999 revela que apenas 40% das predições feitas pelo modelo estavam corretas, enquanto o recall de 0,5333 mostra que ele conseguiu identificar cerca de 53% das ocorrências reais das classes. Esses resultados evidenciam que, embora o modelo apresente uma capacidade moderada de generalização, o uso de um conjunto de dados maior e mais diversificado, melhoraria o desempenho do modelo.

Tabela 7 - Resumo das informações do treinamento reduzido.

Conjunto de dados	
Conjunto original	Total de 129 imagens
Treinamento	105 imagens
Validação	24 imagens
Treinamento YOLOv8n	
Modelo	YOLOv8n
Épocas	200
Tamanho do lote	16
Desempenho	

mAP50	0.4869
Precisão	0.3999
Recall	0.53333
F1-score	0.4571

Resultados do Treinamento



Fonte: elaboração própria (2024).

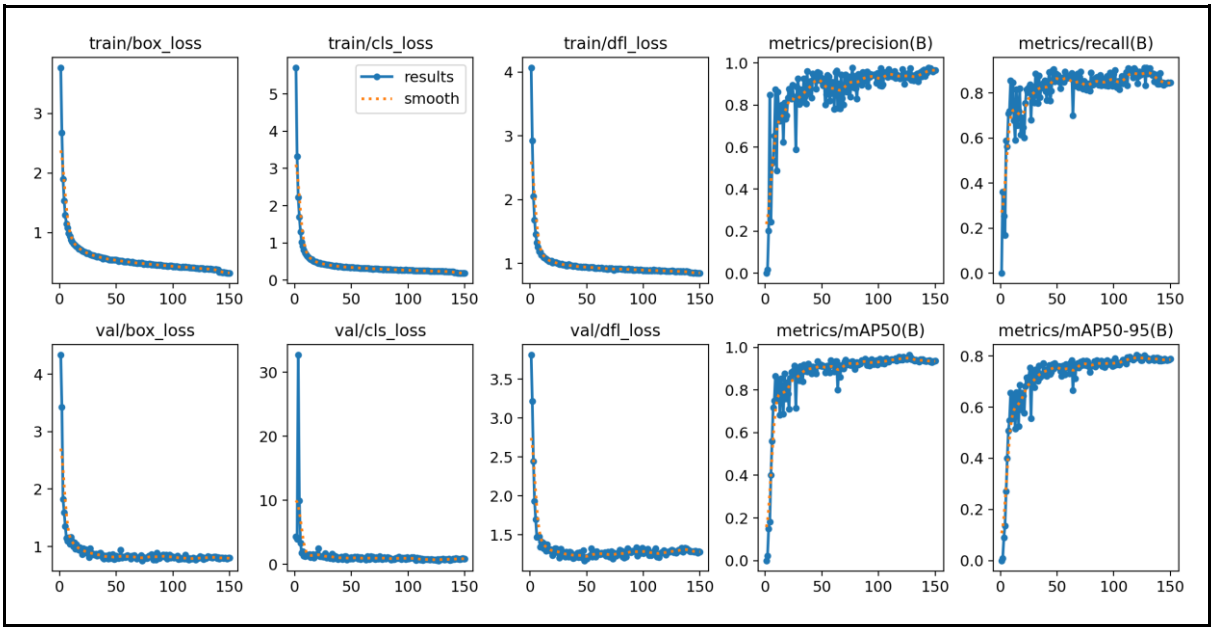
4.4.2 Modelo YOLOv8n

O modelo YOLOv8n[4] obteve um equilíbrio entre precisão e cobertura das classes. A baixa perda de regressão da caixa (*box_loss*) indica que o modelo aprendeu bem as localizações dos objetos, minimizando o erro entre as caixas preditas e as reais. A perda de classificação (*cls_loss*), que mede a precisão na identificação das classes, e a perda focal distributiva (*dfl_loss*), que ajusta a precisão das bordas das caixas delimitadoras, mantiveram-se estáveis, sugerindo um bom ajuste das caixas. Esses resultados, sem queda nas métricas de validação, indicam que não houve *overfitting*, e que o modelo apresenta bom desempenho tanto nos dados de treino quanto nos de validação.

[4] Notebook do treinamento realizado para o modelo YOLOv8n. Disponível em: https://colab.research.google.com/drive/1S83bdQflSNpH9y7FoDkQ3hCHD-TRxG_E?usp=sharing

Tabela 8 - Resumo das informações treinamento YOLOv8n.

Conjunto de dados	
Conjunto original	Total de 2.785 imagens
Treinamento	1973 imagens
Validação	463 imagens
Teste	349 imagens
Treinamento YOLOv8n	
Modelo	YOLOv8n
Épocas	150
Tamanho do lote	32
Desempenho	
mAP50	0.9371
Precisão	0.9651
Recall	0.8448
F1-score	0.9018
Resultados do Treinamento	



Fonte: elaboração própria (2024).

4.4.3 Modelo YOLOv8s

O modelo YOLOv8s[5] alcançou métricas que indicam um desempenho consistente, com alta precisão e capacidade de identificar a maioria das instâncias das classes corretamente. Embora o mAP50 e o F1-*score* sejam inferiores aos do modelo YOLOv8n, o YOLOv8s ainda demonstra um excelente equilíbrio entre precisão e recall, com uma precisão alta que evidencia a habilidade do modelo de evitar falsos positivos. Além disso, a redução consistente das perdas sem um aumento nas métricas de validação indica que o modelo não está sofrendo de *overfitting*, mantendo um bom equilíbrio entre desempenho em dados de treino e generalização em dados de validação

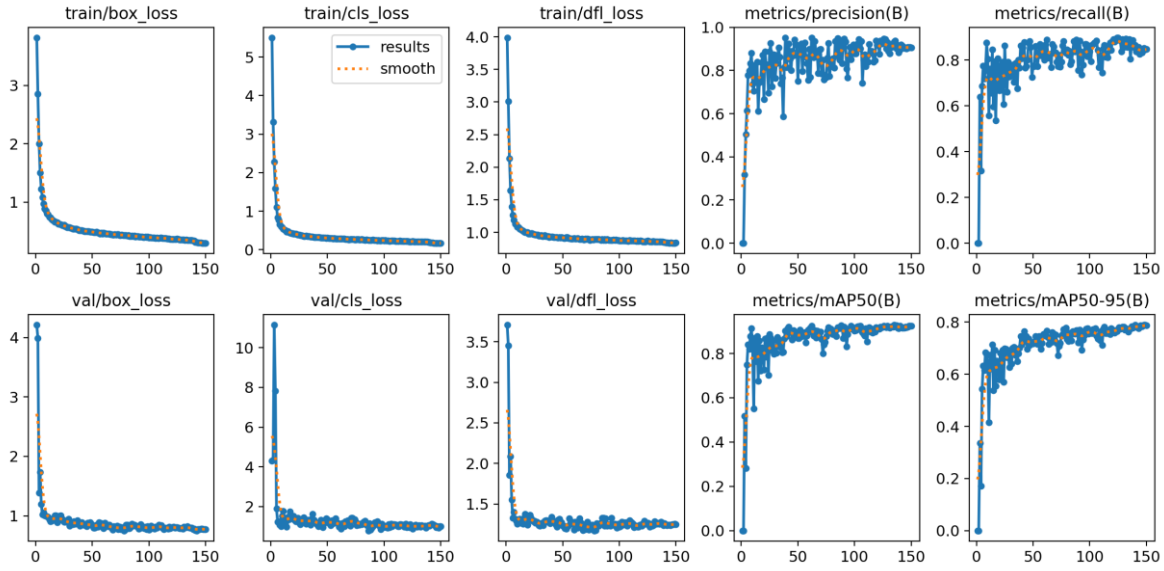
Tabela 9 - Resumo das informações treinamento YOLOv8s.

Conjunto de dados	
Conjunto original	Total de 2.785 imagens
Treinamento	1973 imagens

[5] Notebook do treinamento realizado do modelo YOLOv8s. Disponível em: https://colab.research.google.com/drive/11ZMsBYwIXqNximzNHZlwgkWlm4mazbTo?usp=sharing

Validação	463 imagens
Teste	349 imagens
Treinamento YOLOv8s	
Modelo	YOLOv8s
Épocas	150
Tamanho do lote	32
Desempenho	
mAP50	0.9244
Precisão	0.90706
Recall	0.8483
F1-score	0.8767

Resultados do Treinamento



Fonte: elaboração própria.

4.4.4 Modelo YOLOv10n

O modelo YOLOv10n[6], treinado por 139 épocas devido ao critério de *early stopping*, apresentou um desempenho moderado em comparação com os treinamentos anteriores. O modelo conseguiu identificar e classificar as instâncias de forma aceitável, mas com valores significativamente inferiores aos obtidos pelos modelos YOLOv8. A diminuição das perdas ao longo do treinamento para os conjuntos de treino e teste sugere que o YOLOv10n aprimorou sua capacidade de localizar e classificar objetos, embora a oscilação e tendência de aumento na *cls_loss* na validação possam indicar dificuldades na generalização da classificação das classes. Este aumento na *cls_loss* pode ser um sinal de que o modelo estava começando a se sobreajustar aos dados de treinamento, o que pode ter impactado o desempenho de validação.

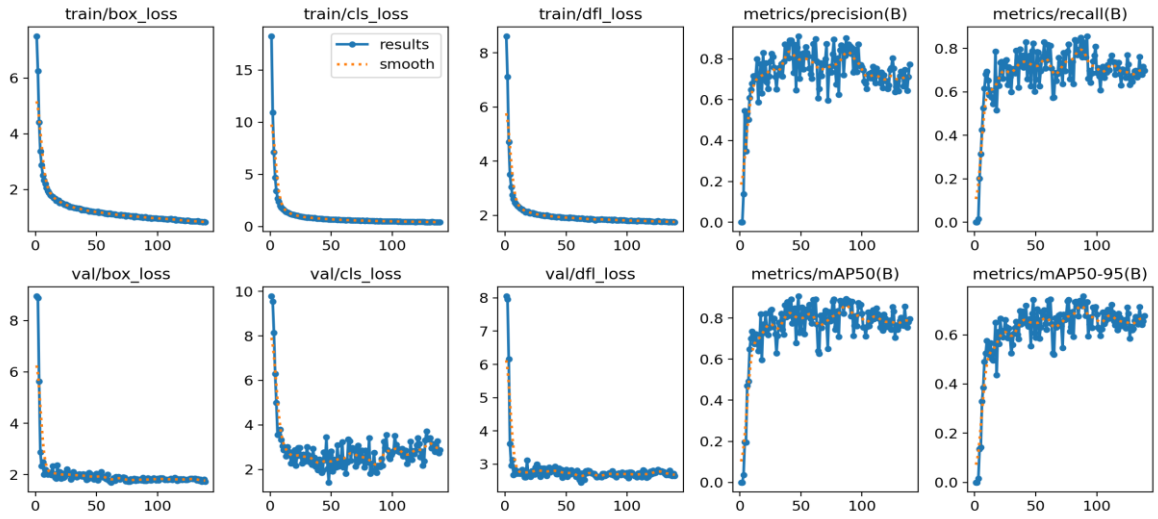
Tabela 10 - Resumo das informações treinamento YOLOv10n.

Conjunto de dados	
Conjunto original	Total de 2.785 imagens
Treinamento	1973 imagens
Validação	463 imagens
Teste	349 imagens
Treinamento YOLOv10n	
Modelo	YOLOv10n
Épocas	139
Tamanho do lote	32
Desempenho	
mAP50	0.7969

[6] Notebook do treinamento realizado para o modelo YOLOv10n. Disponível em: https://colab.research.google.com/drive/1_dYg0DlKDgOfq5l8m5-VV6eIF_2EaUp3?usp=sharing

Precisão	0.7734
Recall	0.6979
F1-score	0.7338

Resultados do Treinamento



Fonte: elaboração própria.

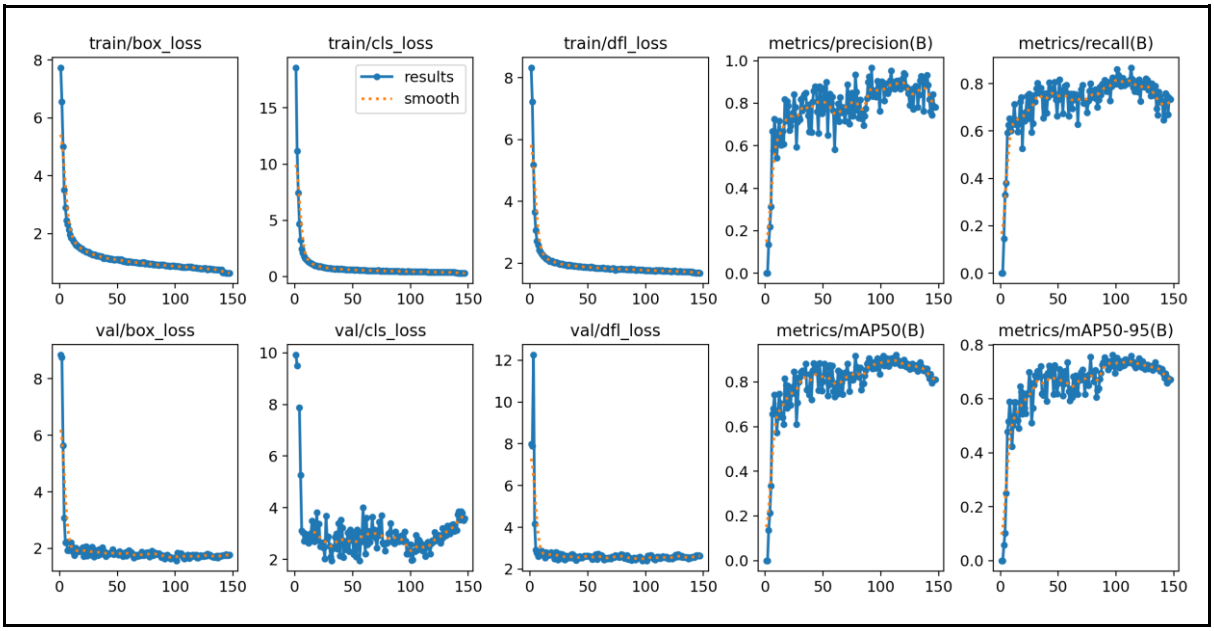
4.4.5 Modelo YOLOv10s

O modelo YOLOv10s[7], treinado por 147 épocas devido ao critério de *early stopping*, apresentou um desempenho um pouco melhor em comparação com o YOLOv10n, mas ainda inferior aos modelos YOLOv8. A diminuição das perdas ao longo do treinamento, tanto nos conjuntos de treino quanto de teste, sugere que o YOLOv10s melhorou sua habilidade de localizar e classificar objetos, mas a oscilação e tendência de aumento na *cls_loss* durante a validação indicam que o modelo pode estar enfrentando desafios na generalização das classificações. Essa oscilação na *cls_loss* pode sinalizar um possível início de *overfitting*, impactando o desempenho nas métricas de validação. Embora o YOLOv10s tenha mostrado um desempenho mais robusto que o YOLOv10n, ainda há espaço para ajustes a fim de alcançar resultados mais consistentes.

Tabela 11 - Resumo das informações treinamento YOLOv10s.

[7] Notebook do treinamento realizado para o modelo YOLOv10s. Disnponível em: https://colab.research.google.com/drive/1QAy2Mm5W_Lctk28UxAoMGYPUqt_Er8v4?usp=sharing

Conjunto de dados	
Conjunto original	Total de 2.785 imagens
Treinamento	1973 imagens
Validação	463 imagens
Teste	349 imagens
Treinamento YOLOv10s	
Modelo	YOLOv10s
Épocas	147
Tamanho do lote	32
Desempenho	
mAP50	0.8112
Precisão	0.7813
Recall	0.7340
F1-score	0.7569
Resultados do Treinamento	



Fonte: elaboração própria (2024).

4.5 EXECUÇÃO DOS TESTES

Foram realizados cinco testes com os modelos previamente treinados. Os quatro primeiros testes utilizaram um conjunto de 349 imagens, sendo 132 da classe feliz, 148 da classe neutro e 69 da classe medo. O quinto e último teste foi realizado com um conjunto reduzido, porém composto de novos rostos e expressões, contendo 10 imagens para cada classe, totalizando 30 imagens. Todas as imagens de teste mantêm as mesmas características dos conjuntos de treinamento e validação, garantindo consistência na avaliação. Os testes foram executados no ambiente *Google Colab*, utilizando os parâmetros da última época do treinamento. Ao final, foram avaliadas as métricas de mAP50, precisão, *recall* e F1-*score*, além da matriz de confusão

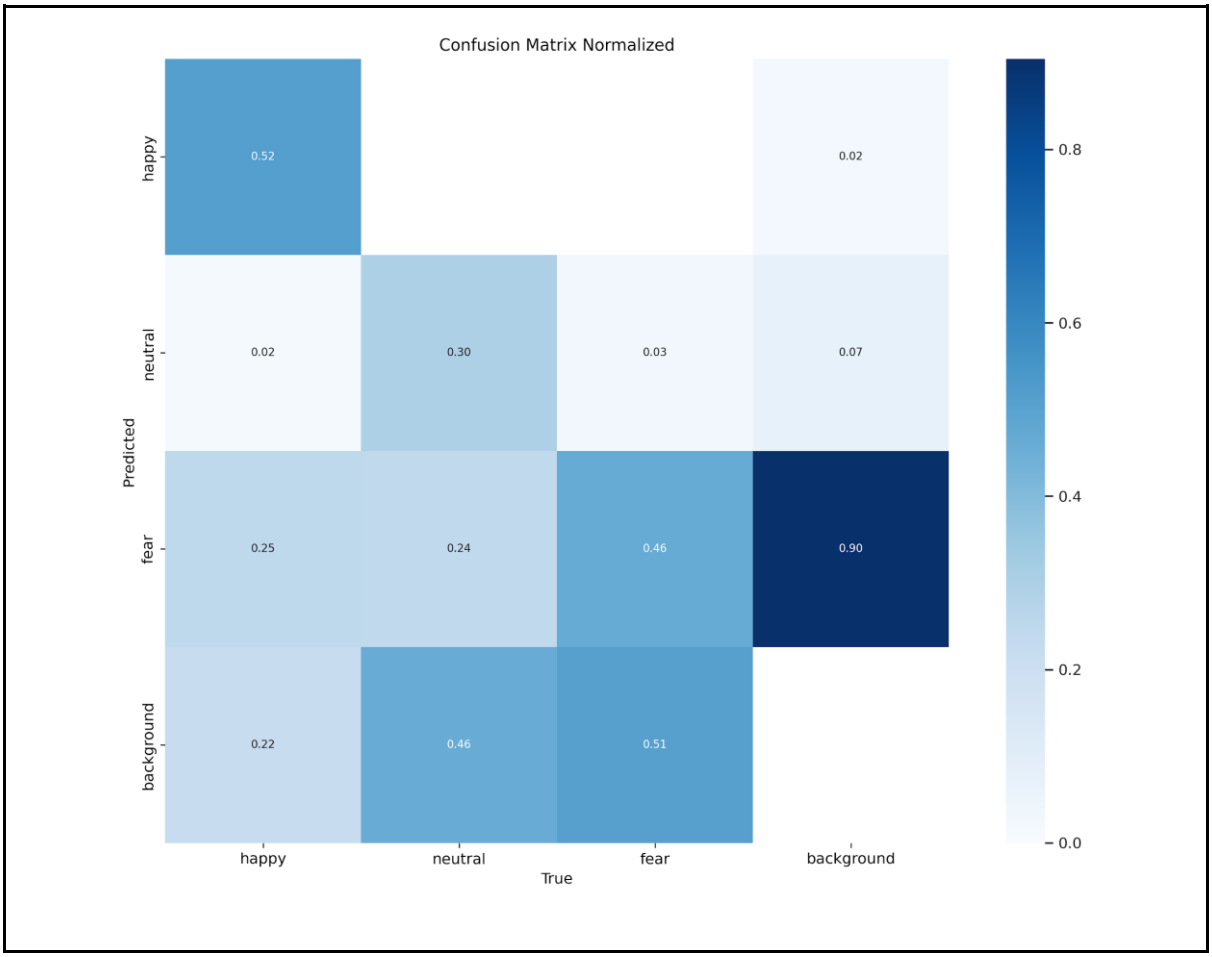
4.5.1 Modelo YOLOv8n

Os resultados do modelo YOLOv8n nos testes mostram uma queda significativa em comparação com as métricas obtidas durante o treinamento. A queda acentuada no desempenho dos resultados de teste pode ter sido influenciada pela reduzida quantidade de dados disponíveis nesse conjunto, o que limita a representatividade e a diversidade das imagens testadas. Analisando a matriz de confusão do modelo, percebe-se dificuldades significativas na classificação das emoções "neutral" e "medo". A classe "neutral" apresentou várias confusões, especialmente com a classe "medo". Da mesma forma, a classe "medo" foi

frequentemente confundida com "neutral", indicando que o modelo teve dificuldades para distinguir claramente entre essas duas emoções. Esses resultados sugerem que as expressões faciais de "medo" e "neutral" não estão sendo suficientemente distintas para o modelo.

Tabela 12 - Resultados do teste para o modelo YOLOv8n.

Modelo YOLOv8n	
Tamanho do lote	32
Tamanho da imagem	640
Avaliação de desempenho	
mAP50	0.5359
Precision	0.6269
Recall	0.5012
F1-score	0.5571
Matriz Confusão	



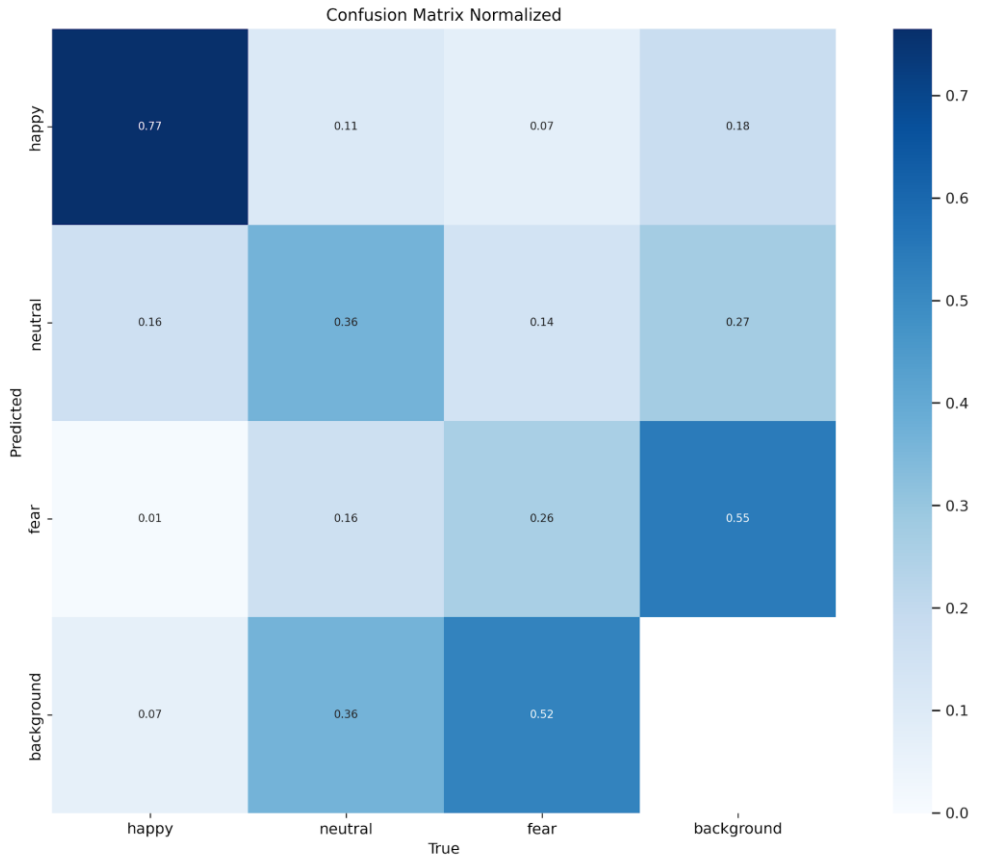
Fonte: elaboração própria (2024).

4.5.2 Modelo YOLOv8s

Os resultados do modelo YOLOv8s nos testes indicam uma queda em comparação com as métricas obtidas durante o treinamento. Essa diferença sugere que, embora o YOLOv8s tenha se ajustado bem aos dados de treinamento, ele não conseguiu generalizar para dados novos. Além disso, os resultados do YOLOv8s foram inferiores aos do modelo YOLOv8n. A redução nas métricas de teste do YOLOv8s pode ser atribuída a um conjunto de dados de teste que pode não ter sido suficientemente diversificado. Para melhorar a robustez do modelo, seria interessante aumentar a diversidade e a quantidade do conjunto de dados de teste. A análise da matriz de confusão nos mostra que, apesar de um bom desempenho na detecção da emoção "feliz", ainda há desafios significativos com as classes "neutral" e "medo". A classe "neutral" apresentou várias confusões, especialmente com "feliz" e "medo", sugerindo que as expressões neutras compartilham características visuais com essas emoções. A classe "medo" teve uma alta taxa de confusão com "neutral", indicando que o modelo encontra dificuldades em separar essas duas emoções de forma clara.

Tabela 13 - Resultados do teste para o modelo YOLOv8s.

Modelo YOLOv8s	
Tamanho do lote	32
Tamanho da imagem	640
Avaliação de Desempenho	
mAP50	0.5264
Precision	0.4769
Recall	0.5244
F1-score	0.4896
Matriz Confusão	



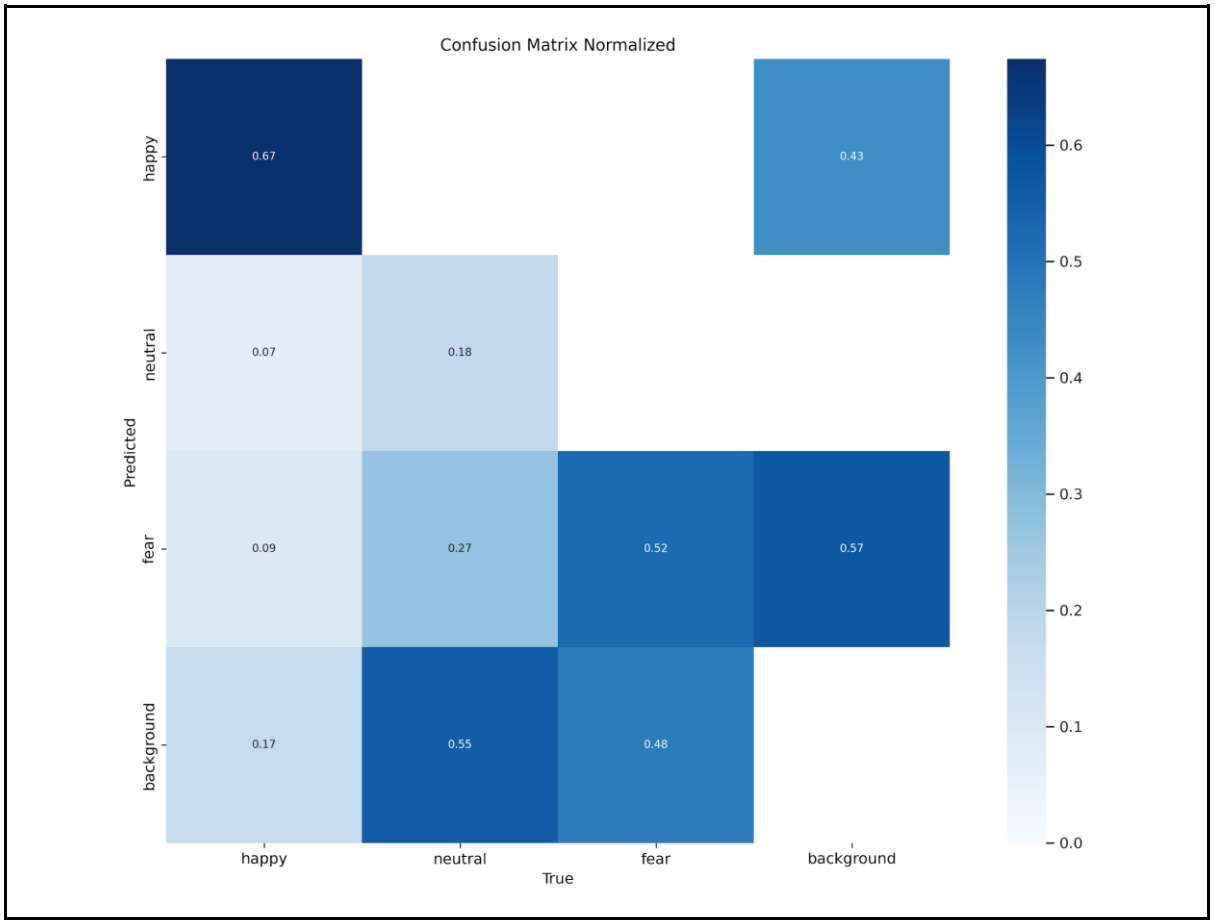
Fonte: elaboração própria (2024).

4.5.3 Modelo YOLOv10n

Os resultados do modelo YOLOv10n nos testes mostram que, embora tenha se ajustado menos aos dados de treinamento em relação aos modelos YOLOv8n e YOLOv8s, o YOLOv10n mostrou um desempenho mais consistente nos dados de teste. A análise da matriz de confusão mostra um bom desempenho na detecção da emoção "feliz". No entanto, houve bastante confusão entre as classes "neutral" e "medo", indicando que o modelo teve dificuldades em distinguir essas duas emoções, sugerindo que o modelo não conseguiu distinguir características distintas para separá-las de forma clara.

Tabela 14 - Resultados do teste para o modelo YOLOv10n.

Modelo YOLOv10n	
Tamanho do lote	32
Tamanho da imagem	640
Avaliação de Desempenho	
mAP50	0.5413
Precision	0.7480
Recall	0.5064
F1-score	0.5789
Matriz Confusão	



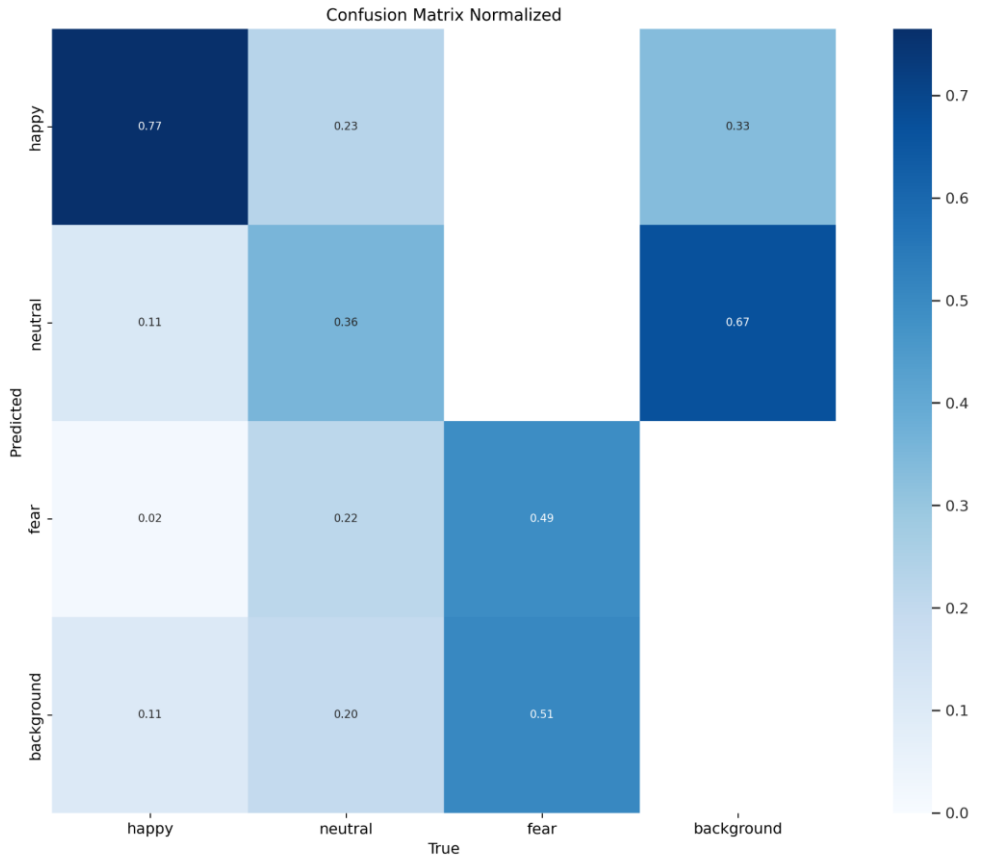
Fonte: elaboração própria (2024).

4.5.4 Modelo YOLOv10s

Os resultados do modelo YOLOv10s nos testes mostram uma queda nas métricas em relação ao treinamento, mas revelam também um desempenho superior em comparação aos outros modelos nos dados de teste, indicando uma melhor capacidade de generalização na localização de objetos, e também maior precisão – que foi de 0,7901. A análise da matriz de confusão revela que o modelo teve um bom desempenho na detecção da emoção "feliz". No entanto, observou-se confusões significativas entre as classes "neutral" e "medo", indicando que o modelo não conseguiu distinguir bem essas duas emoções. Esse padrão sugere que o modelo enfrenta dificuldades para separar expressões neutras das expressões de medo, possivelmente devido a semelhanças visuais entre elas. Para melhorar essa classificação, ajustes no conjunto de dados, com exemplos mais evidentes das diferenças entre "neutral" e "medo", poderiam ajudar o modelo a captar melhor essas distinções.

Tabela 15 - Resultados do teste para o modelo YOLOv10s.

Modelo YOLOv10s	
Tamanho do lote	32
Tamanho da imagem	640
Avaliação de Desempenho	
mAP50	0.5861
Precision	0.7901
Recall	0.5166
F1-score	0.5939
Matriz Confusão	

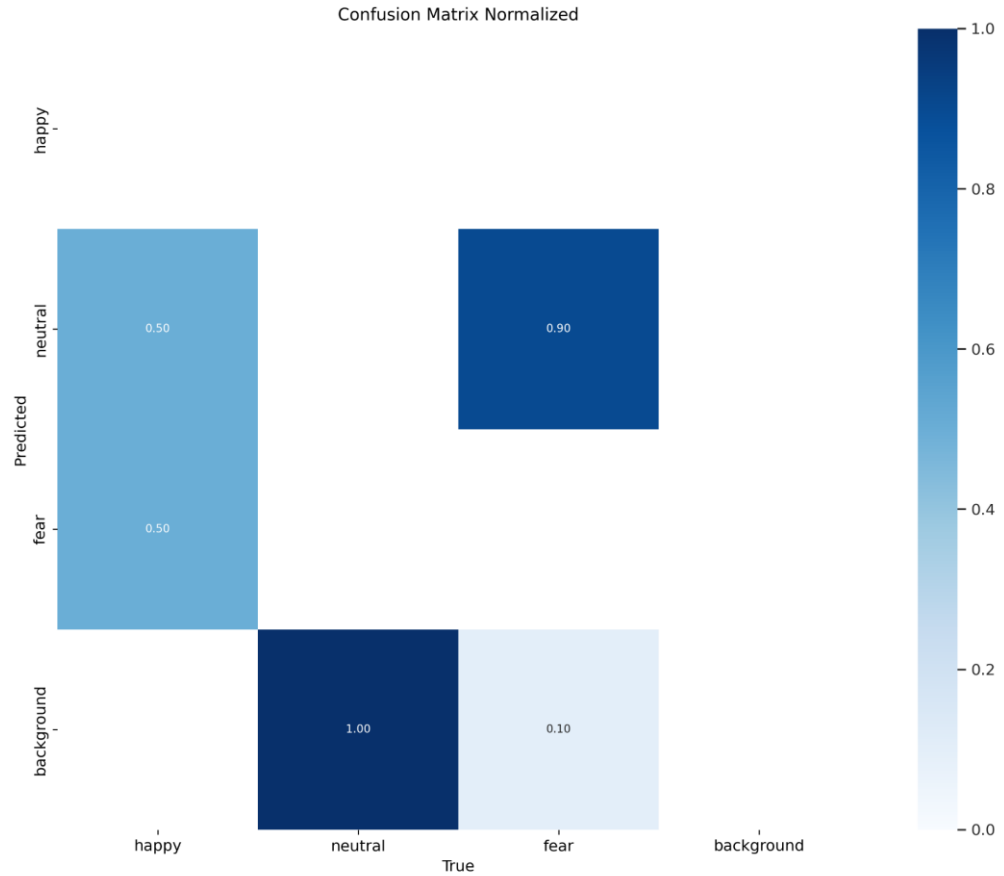


Fonte: elaboração própria.

4.5.5 Resultados YOLOv10s (novo conjunto de testes)

O modelo YOLOv10s foi escolhido para realizar o último teste com o novo conjunto, pelo motivo de apresentar o melhor desempenho nos testes realizados anteriormente com o conjunto de dados de teste original. Através da matriz de confusão, percebe-se que o modelo YOLOv10s apresentou um desempenho muito insatisfatório, não conseguindo fazer nenhuma predição correta para qualquer uma das classes. A ausência de acertos pode ser atribuída, em partes, ao baixo número de imagens presentes no conjunto de dados de teste, limitando o desempenho. O teste também reforça um problema observado anteriormente: as características das imagens das classes “neutral” e “medo” não estão suficientemente distintas, causando uma confusão recorrente entre essas classes.

Figura 24 - Resultados do teste do modelo YOLOv10s para o novo conjunto de teste.



Fonte: elaboração própria (2024).

4.6 Aprimoramento do modelo YOLOv10s

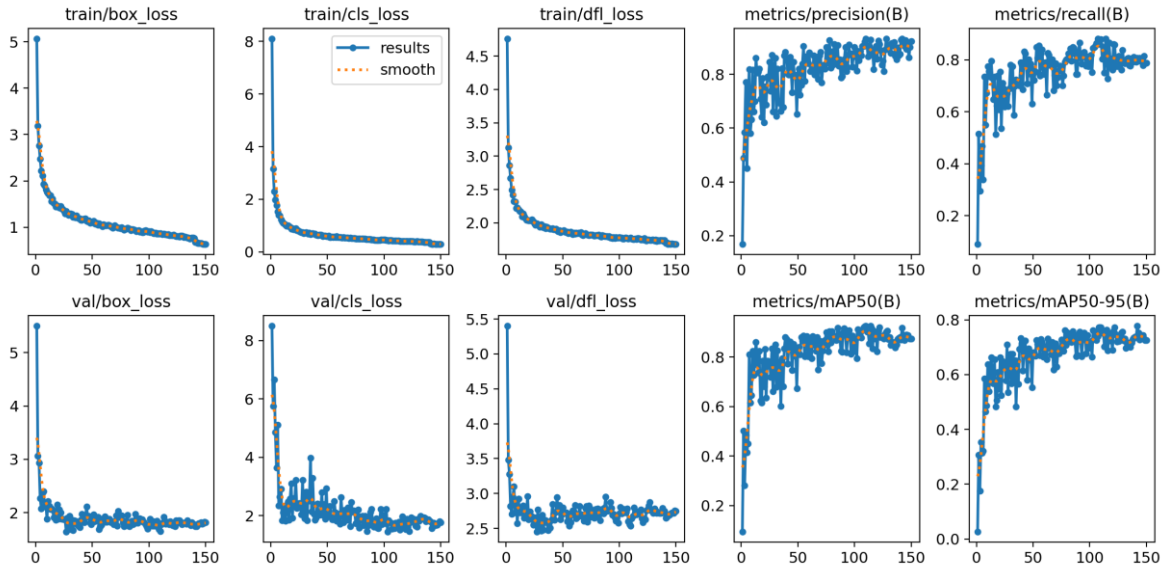
O modelo YOLOv10s[8] foi selecionado com o objetivo de melhorar o desempenho, pois apresentou os melhores resultados nos testes realizados anteriormente. Para essa nova tentativa, alguns parâmetros foram ajustados em relação aos treinamentos anteriores, como o tamanho do lote, o otimizador e o aumento de dados, que foi intensificado. O tamanho do lote foi reduzido de 32 para 16, visando uma melhor generalização, e o otimizador escolhido foi o AdamW, que pode oferecer uma atualização mais eficiente dos pesos. Quanto ao aumento de dados, foram aplicadas diversas transformações, como rotação de 30 e -30 graus, *shear* de 20 graus, e ajustes de matiz (25 graus), saturação (30%) e brilho (30%). Além disso, foi aplicado *blur* de 2px. Todas essas transformações foram configuradas para ocorrer com uma probabilidade de 50%. A seguir, apresenta-se os resultados do treinamento, seguidos dos testes realizados.

Tabela 16 - Resumo das informações treinamento YOLOv10s, otimizador AdamW.

Conjunto de dados	
Conjunto original	Total de 2.785 imagens
Treinamento	1973 imagens
Validação	463 imagens
Teste	349 imagens
Treinamento YOLOv10s	
Modelo	YOLOv10s
Épocas	150
Tamanho do lote	16
Desempenho	
mAP50	0.8744

[8] Notebook do treinamento realizado YOLOv10s. Disponível em: https://colab.research.google.com/drive/1_2cv8ev9mL2A5O9erFIL_nHdgoKm4q6u?usp=sharing

Precisão	0.9236
Recall	0.7876
F1-score	0.8502
Resultados do Treinamento	



Fonte: elaboração própria (2024).

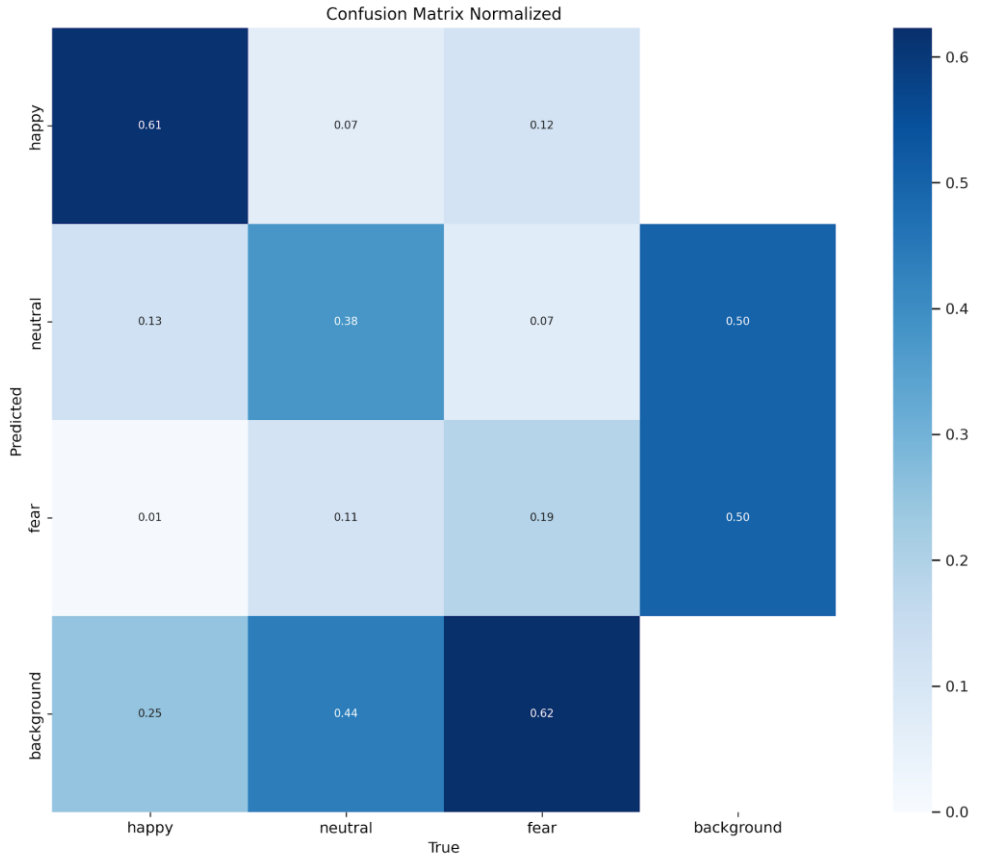
O modelo apresentou um aprimoramento nas métricas em relação ao treinamento anterior, evidenciando que as alterações nos parâmetros e o aumento de dados contribuíram para esse melhor desempenho. A redução nas métricas *box_loss* e *cls_loss* de validação ao longo das épocas indica que o modelo está progredindo tanto na localização quanto na classificação das imagens.

Tabela 17 - Resultados do teste para o modelo YOLOv10s, otimizador AdamW.

Modelo YOLOv10s	
Tamanho do lote	16
Tamanho da imagem	640
Avaliação de Desempenho	
mAP50	0.4884

Precision	0.6685
Recall	0.4248
F1-score	0.5107

Matriz Confusão



Fonte: elaboração própria (2024).

Apesar da melhora observada no desempenho do modelo no conjunto de validação, os resultados no conjunto de teste não apresentaram a mesma evolução, demonstrando uma pior performance comparado ao modelo previamente treinado. Embora o modelo tenha mostrado um bom desempenho ao classificar a classe "feliz", ele continua apresentando dificuldades em distinguir as classes "neutral" e "medo" no conjunto de teste.

5. ANÁLISE DOS RESULTADOS

A tabela abaixo (Tabela 18), apresenta um resumo dos resultados das métricas obtidas durante a etapa de validação dos treinamentos realizados. Essas métricas fornecem uma avaliação detalhada do desempenho dos modelos ao longo do processo de treinamento. Apesar do bom desempenho na validação dos treinamentos, os modelos não tiveram uma boa generalização.

Tabela 18 - Resultados dos testes realizados.

	Resultados da Validação nos Treinamentos			
Modelo	**maAP50**	**Precisão**	**Recall**	**F1-Score**
YOLOv8n	0.9371	0.9651	0.8448	0.9018
YOLOv8s	0.9244	0.9070	0.8483	0.8767
YOLOv10n	0.7969	0.7734	0.6979	0.7338
YOLOv10s	0.8112	0.7813	0.7340	0.7569
YOLOv10s (Otimizador AdamW)	0.8744	0.9236	0.7876	0.8502

Fonte: elaboração própria (2024).

A tabela a seguir (Tabela 19), apresenta um resumo dos resultados das métricas obtidas na etapa de teste dos modelos treinados. Esses resultados refletem o desempenho final dos modelos em dados completamente novos, fornecendo uma avaliação imparcial da sua capacidade de generalização em condições reais de uso.

Tabela 19 - Resultados dos testes realizados.

	Resultados dos Testes			
Modelo	**maAP50**	**Precisão**	**Recall**	**F1-Score**
YOLOv8n	0.5359	0.6269	0.5012	0.5571
YOLOv8s	0.5264	0.4769	0.5244	0.4896
YOLOv10n	0.5413	0.7480	0.5064	0.5789
YOLOv10s	0.5861	0.7901	0.5166	0.5939
YOLOv10s (Otimizador AdamW)	0.4884	0.6685	0.4248	0.5107

Fonte: elaboração própria (2024).

Nos treinamentos dos modelos, todos apresentaram desempenhos satisfatórios com métricas elevadas, indicando um bom ajuste aos dados de treinamento. Os modelos YOLOv8n e YOLOv8s foram superiores aos modelos YOLOv10n e YOLOv10s, destacando-se tanto em precisão quanto em capacidade de generalização.

Nos testes, todos os modelos tiveram quedas de desempenho, revelando desafios em generalizar para dados novos. O YOLOv8n e YOLOv8s apresentaram uma perda significativa de precisão e *recall*, indicando dificuldade em manter a consistência fora dos dados de treino. O YOLOv10n teve um desempenho mais equilibrado e manteve sua precisão acima da média, embora ainda tenha enfrentado uma queda em recall. O YOLOv10s foi o modelo com maior desempenho nos testes, mantendo métricas mais estáveis e apresentando o melhor mAP50.

De modo geral, enquanto os modelos YOLOv8 enfrentaram dificuldades em generalizar, o YOLOv10n e, especialmente, o YOLOv10s, conseguiram resultados mais consistentes nos testes, destacando-se como modelos potencialmente mais robustos para aplicação prática. A diversidade limitada do conjunto de dados de teste, no entanto, pode ter contribuído para as quedas de desempenho observadas, indicando a necessidade de um conjunto de teste mais abrangente para uma análise de generalização mais precisa.

Apesar da melhora do desempenho no conjunto de validação com ajuste de parâmetros, o segundo treinamento do modelo YOLOv10s para o conjunto de teste teve resultados inferiores ao primeiro treinamento, não alcançando o objetivo de melhorar o desempenho do modelo.

6. CONSIDERAÇÕES FINAIS

Com base nos resultados obtidos nos treinamentos e testes dos modelos, foi possível observar variações significativas em seu desempenho, especialmente ao avaliar a capacidade de generalização para dados novos. Os modelos YOLOv8n e YOLOv8s demonstraram excelente desempenho nos dados de treinamento, com mAP50 e precisão elevados, indicando que ambos foram capazes de identificar e classificar corretamente as faces no conjunto de treino. No entanto, esses modelos apresentaram uma queda nas métricas durante os testes, sugerindo dificuldades em manter o mesmo nível de precisão e *recall* ao serem aplicados a novos dados. O

YOLOv8n, por exemplo, que alcançou uma precisão de 0,9651 no treinamento, registrou uma redução para 0,6269 nos testes, revelando um possível ajuste excessivo ao conjunto de treino (*overfitting*), o que limitou sua capacidade de generalização.

Por outro lado, os modelos da série YOLOv10, tanto na versão “n” quanto “s”, apresentaram uma performance mais equilibrada entre treinamento e testes, com uma menor queda nas métricas, especialmente na precisão e mAP50. O YOLOv10s, em particular, obteve o melhor mAP50 nos testes – 0,5861 – entre todos os modelos, além de manter a precisão em 0,7901. Além das métricas mencionadas, o F1-*score* também apresentou variações importantes, sendo mais estável nos modelos YOLOv10 do que nos YOLOv8, especialmente no YOLOv10s, que alcançou um F1-*score* de 0,5939 nos testes.

Vale enfatizar que o conjunto de treinamento estava limitado a um total de 1973 imagens, com cinco indivíduos. Em comparação, o *dataset* FER2013, que é um dos conjuntos de dados mais utilizados nos trabalhos relacionados, possui aproximadamente 30.000 imagens divididas em sete classes de emoções, das quais cerca de 28.000 são destinadas para o treinamento. Portanto, o FER2013 oferece uma diversidade maior, o que contribui para um treinamento mais robusto e uma melhor capacidade de generalização em diferentes rostos e expressões.

Uma das contribuições deste trabalho foi a construção do *dataset*, permitindo que futuros estudos aprofundem a análise e desenvolvam soluções mais eficazes sobre o tema. É importante destacar que os erros observados, principalmente entre as classes "neutral" e "medo", ocorreram devido a semelhanças visuais entre as imagens dessas classes. Para resolver esse problema, aumentar o conjunto de dados com imagens que apresentem distinções mais claras entre essas emoções pode ajudar a aprimorar o desempenho do modelo.

Por fim, os modelos YOLOv10 demonstraram um desempenho mais consistente e adaptado aos testes, enquanto os modelos YOLOv8, embora eficazes no treinamento, apresentaram dificuldades de generalização, especialmente devido à menor quantidade de dados no conjunto de testes. Esses resultados indicam que, para aplicações práticas em cenários variados, os modelos YOLOv10, especialmente o YOLOv10s, podem oferecer uma solução mais robusta. Contudo, para uma avaliação ainda mais precisa, seria recomendável considerar o aumento e a diversificação do conjunto de testes, permitindo uma análise completa da capacidade de generalização de cada modelo.

Como proposta para trabalhos futuros, recomenda-se aumentar o conjunto de dados de treino e teste, garantindo uma diversidade que permita ao modelo generalizar melhor e distinguir com mais clareza as emoções, especialmente para as classes “neutral” e “medo”. Além disso, explorar modelos alternativos à arquitetura YOLO, como ResNet e outras Redes Neurais Convolucionais (CNNs), pode ser valioso, dado seu uso frequente em tarefas de classificação de imagens e reconhecimento de padrões faciais.

REFERÊNCIAS

BETTADAPURA, Vinay. **Face expression recognition and analysis: the state of the art**. Atlanta (EUA), 30 de mar. 2012. Disponível em: <https://doi.org/10.48550/arXiv.1203.6722>. Acesso em: 14 de set. de 2024.

DENG, Jiankang *et al*. **RetinaFace: Single-stage dense face localisation in the wild**. 2 de maio de 2019. Disponível em: <https://doi.org/10.48550/arXiv.1905.00641>. Acesso em: 21 de set. de 2024.

EKMAN, Paul. Facial expression and emotion. **American psychologist**, v. 48, n. 4, p. 384, 1993. Disponível em: <https://scholar.google.com/scholar?hl=pt-BR&as_sdt=0%2C5&q=facial+expression+and+emotion+paul+ekman&btnG=&oq=facial+expression+and+emotion+paul+ekma>. Acesso em: 14 de set. de 2024.

GARG, Dweepna; GOEL, Parth; PANDYA, Sharnil; GANATRA, Amit. KOTECHA, Ketan. **A Deep Learning Approach for Face Detection using YOLO**. Pune (IN), 30 de nov. de 2018. Disponível em: <10.1109/PUNECON.2018.8745376>. Acesso em: 14 de set. de 2024.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning (Adaptive Computation and Machine Learning series). Edição nº 1. Cambridge (EUA): The MIT Press. 18 de nov. de 2016.

KHANZADA, Amil; BAI, Charles; CELEPCIKAY, Ferhat Turker. **Facial expression recognition with deep learning**. Stanford (EUA), 8 de abr. De 2020. Disponível em: <https://doi.org/10.48550/arXiv.2004.11823>. Acesso em: 13 de ago. de 2024.

MELO, Sara L.; MOURA, Fabio F. de; MACEDO, Kely; ALVES, Fabiano S. R; FERNANDES, Márcia A. Estudo comparativo de técnicas computacionais para classificação de emoções. **Anais do Simpósio Brasileiro de Informática na Educação (SBIE)**, São Paulo, v. 25, n. 1, p. 456, 2014. Disponível em: <https://doi.org/10.5753/cbie.sbie.2014.456>. Acesso em: 13 de ago. de 2024.

MITCHELL, Tom M. Machine Learning. Edição nº 1. Nova Iorque: McGraw-Hill Education. 01 de mar. de 1997.

RAJESH, K. M; NAVEENKUMAR, M. **A robust method for face recognition and face emotion detection system using support vector machines**. Mysuru (IN), 10 de dez. de 2016. Disponível em: <10.1109/ICEECCOT.2016.7955175>. Acesso em: 14 de set. De 2024.

REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. **You Only Look Once: Unified, Real-Time Object Detection**. 8 de jun. de 2015. Disponível em: <https://doi.org/10.48550/arXiv.1506.02640>. Acesso em: 14 de set. de 2024.

RIYANTOKO, Prismahardi Aji; SUGIARTO, K. M; HINDRAYIANI, Kartika Maulida. Facial emotion detection using Haar-cascade classifier and convolutional neural networks. **Journal of Physics: Conference Series**, 2021.

RUSSEL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach. Edição nº 3. Londres: Pearson. 8 de maio de 2020.

SHAIKH, Awais *et at*. Comprehensive Study on Emotion Detection with Facial Expression Images Using YOLO Models. **Journal of Information Assurance & Security**, v. 18, n 2, 2023.

URBANOWICK, Ryan J. (@DocUrbs). New proposed field/term Venn diagram for an upcoming talk. My take on illustrating the relationship between #DataScience, MachineLearning, ArtificialIntelligence , Statistics, and DataMining. Twitter, 14 de jun. 2018. Disponível em: <https://x.com/DocUrbs/status/1007375834347376642>. Acesso em: 13 de ago. 2024.

YOLOv8: THE LATEST VERSION OF THE YOLO MODEL. **Ultralytics**, 2023. Disponível em: <https://docs.ultralytics.com/>. Acesso em: 2 de set. de 2024.

ZHAO, Zhong-Qiu *et al*. Object detection with deep learning: A review. 15 de jul. de 2018. Disponível em: <https://doi.org/10.48550/arXiv.1807.05511>. Acesso em: 14 de set. de 2024.

APÊNDICE A - ARTIGO NO FORMATO SBC

Desenvolvimento de modelos de deep learning para detecção de emoções durante saltos de paraquedismo

Vinicius Claudino Antunes

Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis / SC, Brasil.

viniciustuness7@gmail.com

Abstract. *Facial expression recognition has been a subject of study since the pioneering works of Charles Darwin and Paul Ekman. With advancements in computer vision, automatic emotion detection systems have found practical applications in areas such as robotics and human-computer interaction. This study developed deep learning models using the YOLO algorithm to identify emotions during skydiving – fear, neutral, and happy – in a high emotional impact context. The results showed that YOLOv10 models achieved better metric balance, while YOLOv8 models exhibited signs of overfitting. Limitations include the reduced dataset and visual similarities between classes, highlighting the importance of more diverse datasets. Future work will focus on dataset expansion and exploring new architectures to improve model generalization*.

Resumo. *O reconhecimento de expressões faciais tem sido objeto de estudo desde os trabalhos pioneiros de Charles Darwin e Paul Ekman. Com os avanços em visão computacional, sistemas automáticos para detecção de emoções ganharam aplicações práticas em áreas como robótica e interação humano-computador. Este estudo desenvolveu modelos baseados em deep learning, utilizando o algoritmo YOLO, para identificar emoções durante saltos de paraquedismo – medo, neutro e feliz – em um contexto de alta carga emocional. Os resultados mostraram que os modelos YOLOv10 apresentaram melhor equilíbrio de métricas, enquanto os modelos YOLOv8 indicaram sinais de overfitting. As limitações incluem o conjunto de dados reduzido e similaridades visuais entre classes, destacando a importância de datasets mais diversificados. Trabalhos futuros focarão na ampliação do dataset e na exploração de novas arquiteturas para melhorar a generalização dos modelos*.

1. Introdução

O estudo das expressões faciais tem sido uma área de interesse desde os tempos de Aristóteles, atraindo a atenção de filósofos, artistas, psicólogos e cientistas ao longo dos séculos. A relação entre as emoções humanas e a aparência do rosto foi explorada por estudiosos como John Bulwer, Charles Darwin e Paul Ekman, cujas contribuições estabeleceram os fundamentos para a compreensão moderna das expressões faciais. Esses avanços formaram a base para os atuais sistemas automáticos de reconhecimento de emoções, amplamente aplicados em áreas como interação humano-computador, robótica e inteligência artificial (BETTADAPURA, 2012).

Nas últimas décadas, o reconhecimento de expressões faciais deixou de ser um tema exclusivo da psicologia, expandindo-se para a ciência da computação. Avanços em visão computacional permitiram o desenvolvimento de sistemas capazes de detectar e classificar expressões em tempo real. O trabalho seminal de Ekman (1993), que identificou seis expressões faciais universais – alegria, tristeza, raiva, medo, surpresa e nojo –, foi fundamental para o desenvolvimento de algoritmos que classificam emoções com precisão.

Esses sistemas têm aplicações em diversas áreas, como robótica, onde robôs precisam interpretar e responder a estados emocionais, e na interação humano-computador, que busca interfaces mais intuitivas e sensíveis ao estado emocional dos usuários. Além disso, setores como medicina, segurança e entretenimento digital utilizam o reconhecimento facial para aprimorar diagnósticos, melhorar a experiência do usuário e criar interações mais naturais em ambientes virtuais (BETTADAPURA, 2012).

Neste contexto, o presente trabalho propõe o desenvolvimento de modelos de deep learning para detecção de emoções durante saltos de paraquedismo. A escolha desse cenário se deve à intensidade emocional da atividade, na qual predominam as emoções medo, neutro e feliz. Utilizando o algoritmo *You Only Look Once* (YOLO) para detecção e classificação, os modelos visam identificar expressões faciais com precisão em um ambiente desafiador e dinâmico. Este estudo busca contribuir para o avanço do reconhecimento de emoções em situações de alto impacto emocional, atendendo à crescente demanda por sistemas capazes de interpretar emoções humanas de forma eficiente em contextos não controlados, caracterizados por reações genuínas.

2. Trabalhos correlatos

Os trabalhos selecionados abordam diferentes arquiteturas para reconhecimento de emoções faciais. O primeiro estudo comparou técnicas tradicionais como Redes Neurais *Multi-Layer Perceptron* (MLP), Redes *Basis Functions Networks* (RBFN) e Redes Bayesianas para classificar emoções a partir de movimentos faciais, com a Rede MLP alcançando uma precisão de 75,72%, a Rede RBF obtendo 71,76% e as Redes Bayesianas atingindo uma precisão de 86%. O segundo trabalho utilizou *deep learning*, aplicando uma arquitetura de rede neural convolucional (CNN) com técnicas como *transfer learning* e *data augmentation* para melhorar a precisão em conjuntos de dados limitados. Já o terceiro estudo propôs o uso do algoritmo YOLO para detecção facial em tempo real, destacando-se pela sua eficiência e velocidade, com os autores alcançando uma acurácia de 92,2% no modelo proposto. No entanto, não foi encontrado nenhum trabalho relacionado ao estudo das emoções faciais especificamente no contexto do paraquedismo, o que destaca a originalidade da presente pesquisa.

3. Solução

3.1 Objetivos

O objetivo deste trabalho é desenvolver modelos baseados em *deep learning* para identificar e classificar expressões faciais em imagens capturadas durante saltos de paraquedismo e nos momentos que antecedem o salto, dentro da aeronave. Os modelos serão

treinados para prever, com precisão, se as expressões detectadas pertencem a uma das três classes emocionais – feliz, neutro ou medo –, contribuindo para a análise de emoções em situações extremas e contextos de alta carga emocional.

3.2 Preparação de dados

A coleta de dados para este estudo baseou-se na análise de 33 vídeos de saltos de paraquedismo, que serviram como principal fonte para a análise das expressões faciais. Esses vídeos capturaram os paraquedistas em diferentes momentos do salto, apresentando expressões faciais sob variadas condições de iluminação, ângulos e distâncias da câmera. Para preparar os dados para o treinamento dos modelos, foi utilizado um *script* em Python para extração de frames específicos. Esse processo foi realizado de forma semiautomática, pois exigiu a seleção manual de frames em que as faces apareciam claramente visíveis, frontais, próximas à câmera e sob condições adequadas de iluminação. A identificação desses frames foi feita manualmente, observando os momentos nos vídeos que atendiam a esses critérios. Os instantes selecionados foram então indicados no script, que realizou a extração dos frames correspondentes.

Tabela 1 – Conjunto de imagens rotulados (elaboração própria, 2024).

Conjunto de dados	
Classe	**Total de Anotações**
Feliz	1.110
Neutro	1.158
Medo	517

(A)	(B)	(C)

Figura 1 – Exemplo de imagens do conjunto (elaboração própria, 2024). (A) representa a classe medo, (B) representa a classe neutro e (C) representa a classe feliz.

Nas etapas de treinamento, foram aplicadas técnicas como aumento de dados para diversificar o conjunto de imagens e melhorar a generalização dos modelos, além de *early stop* para evitar o *overfitting* durante o treinamento. Essas abordagens contribuíram para aprimorar o desempenho dos modelos em um contexto de alta variabilidade.

Durante o treinamento, o conjunto de treinamento foi usado para ensinar o modelo a generalizar os padrões e características presentes nos dados. A cada época de treinamento, o conjunto de validação foi empregado para avaliar a capacidade do modelo de generalizar para

dados não utilizados no treinamento direto. Por fim, o conjunto de testes foi reservado para medir o desempenho final do modelo, garantindo uma avaliação imparcial após o término do processo de treinamento e validação.

3.3 Treinamento

O treinamento foi realizado utilizando os modelos YOLOv8n, YOLOv8s, YOLOv10n e YOLOv10s, diferenciados pelo tamanho e complexidade. As versões "nano" são menores e mais rápidas, adequadas para dispositivos com limitações de hardware, enquanto as versões "s" possuem maior capacidade de aprendizado, oferecendo melhor desempenho em cenários com mais recursos computacionais.

Foram aplicadas técnicas como aumento de dados, otimizador *Stochastic Gradient Descent* (SGD) e *early stop*. O aumento de dados gerou variações nas imagens, como rotações, ajustes de brilho e recortes, para melhorar a capacidade de generalização. O SGD otimizou os pesos da rede de forma eficiente, e o *early stop* evitou *overfitting,* interrompendo o treinamento quando o desempenho estabilizou.

O desempenho foi monitorado por métricas como mAP50, precisão e recall, com análise das métricas de perda ao longo das épocas. Ao todo, seis treinamentos foram realizados, incluindo um inicial com dados reduzidos, quatro com as diferentes arquiteturas e um final para otimizar os resultados. O conjunto de dados incluiu 2.785 imagens, divididas em 1.973 para treinamento, 463 para validação e 349 para teste. Os resultados da validação dos treinamentos estão na Tabela 2.

Tabela 2 – Resultados da validação dos treinamentos (elaboração própria, 2024).

	Resultados da Validação nos Treinamentos			
Modelo	**maAP50**	**Precisão**	**Recall**	**F1-Score**
YOLOv8n	0.9371	0.9651	0.8448	0.9018
YOLOv8s	0.9244	0.9070	0.8483	0.8767
YOLOv10n	0.7969	0.7734	0.6979	0.7338
YOLOv10s	0.8112	0.7813	0.7340	0.7569
YOLOv10s (Otimizador AdamW)	0.8744	0.9236	0.7876	0.8502

Nos treinamentos dos modelos, todos apresentaram desempenhos satisfatórios com métricas elevadas, indicando um bom ajuste aos dados de treinamento. Os modelos YOLOv8n e YOLOv8s foram superiores aos modelos YOLOv10n e YOLOv10s, destacando-se tanto em precisão quanto em capacidade de generalização.

3.4 Testes de desempenho

Cinco testes foram realizados com os modelos previamente treinados. Os quatro primeiros utilizaram um conjunto de 349 imagens, composto por 132 da classe feliz, 148 da classe neutro e 69 da classe medo. O quinto teste foi realizado com um conjunto reduzido, mas contendo novas imagens e expressões faciais, com 10 imagens para cada classe, totalizando 30

imagens. Todas as imagens de teste foram selecionadas para manter as características dos conjuntos de treinamento e validação, garantindo consistência na avaliação.

Os testes foram conduzidos no ambiente Google Colab, utilizando os parâmetros da última época do treinamento. Ao final, foram analisadas as métricas de mAP50, precisão, recall, F1-score e a matriz de confusão para avaliar o desempenho do modelo. Os resultados dos testes estão na Tabela 3.

Tabela 3 – Resultados dos testes (elaboração própria, 2024).

	Resultados dos Testes			
Modelo	**maAP50**	**Precisão**	**Recall**	**F1-Score**
YOLOv8n	0.5359	0.6269	0.5012	0.5571
YOLOv8s	0.5264	0.4769	0.5244	0.4896
YOLOv10n	0.5413	0.7480	0.5064	0.5789
YOLOv10s	0.5861	0.7901	0.5166	0.5939
YOLOv10s (Otimizador AdamW)	0.4884	0.6685	0.4248	0.5107

Nos testes, todos os modelos tiveram quedas de desempenho, revelando desafios em generalizar para dados novos. O YOLOv8n e YOLOv8s apresentaram uma perda significativa de precisão e recall, indicando dificuldade em manter a consistência fora dos dados de treino. O YOLOv10n teve um desempenho mais equilibrado e manteve sua precisão acima da média, embora ainda tenha enfrentado uma queda em recall. O YOLOv10s foi o modelo com maior desempenho nos testes, mantendo métricas mais estáveis e apresentando o melhor mAP50.

4. **Discussão**

Durante o treinamento dos modelos, todos apresentaram desempenhos satisfatórios com métricas elevadas, indicando um bom ajuste aos dados de treinamento. Os modelos YOLOv8n e YOLOv8s se destacaram em comparação com os modelos YOLOv10n e YOLOv10s, demonstrando maior precisão e capacidade de generalização.

Nos testes, foi observada uma queda no desempenho de todos os modelos, o que revelou dificuldades em generalizar para dados não vistos anteriormente. O YOLOv8n e o YOLOv8s apresentaram uma queda significativa em precisão e recall, indicando dificuldades em manter consistência fora do conjunto de treinamento. O YOLOv10n apresentou um desempenho mais equilibrado, mantendo uma boa precisão, mas ainda com uma queda em recall. O YOLOv10s foi o modelo com melhor desempenho nos testes, mantendo métricas mais estáveis e alcançando o melhor mAP50.

De maneira geral, os modelos YOLOv8 enfrentaram dificuldades de generalização, enquanto o YOLOv10n e, especialmente, o YOLOv10s, exibiram resultados mais consistentes, destacando-se como modelos mais robustos para aplicações práticas. No entanto, a diversidade limitada no conjunto de dados de teste pode ter influenciado as quedas de desempenho

observadas, apontando para a necessidade de um conjunto de dados de teste mais abrangente para uma avaliação de generalização mais precisa.

Embora tenha ocorrido uma melhora no desempenho no conjunto de validação com ajustes nos parâmetros, o segundo treinamento do modelo YOLOv10s, aplicado ao conjunto de teste, não conseguiu superar o desempenho do primeiro treinamento, não atingindo o objetivo de melhorar os resultados.

5. Conclusão

A partir dos resultados obtidos nos treinamentos e testes dos modelos, observou-se variações significativas em seu desempenho, especialmente em relação à capacidade de generalização para novos dados. Os modelos YOLOv8n e YOLOv8s apresentaram excelente desempenho durante o treinamento, com mAP50 e precisão elevados, indicando boa capacidade de identificação e classificação das faces no conjunto de treino. No entanto, esses modelos apresentaram queda nas métricas durante os testes, o que sugere dificuldades em manter a precisão e recall quando aplicados a dados inéditos. Por exemplo, o YOLOv8n, que obteve precisão de 0,9651 no treinamento, caiu para 0,6269 nos testes, sugerindo um possível *overfitting* que comprometeu a capacidade de generalização.

Em contrapartida, os modelos da série YOLOv10 (tanto nas versões “n” quanto “s”) demonstraram um desempenho mais equilibrado entre treinamento e testes, com menor queda nas métricas, especialmente em precisão e mAP50. O modelo YOLOv10s, em particular, obteve o melhor mAP50 nos testes (0,5861) e manteve uma precisão de 0,7901, destacando-se entre os modelos testados. Além disso, o F1-score foi mais estável nos modelos YOLOv10, especialmente no YOLOv10s, que atingiu 0,5939 nos testes.

Cabe destacar que o conjunto de treinamento utilizado era limitado, com apenas 1.973 imagens de cinco indivíduos. Em comparação com o *dataset* FER2013, amplamente utilizado em estudos semelhantes, que contém cerca de 30.000 imagens divididas em sete classes de emoções, a diversidade de nosso conjunto de dados foi bem mais restrita. Isso pode ter influenciado a capacidade dos modelos de generalizar, destacando a importância de um conjunto de dados maior e mais diversificado para melhorar o desempenho.

Uma contribuição importante deste trabalho foi a criação do *dataset*, que oferece uma base valiosa para estudos futuros sobre o reconhecimento de emoções faciais. É importante ressaltar que os erros observados, especialmente entre as classes “neutral” e “medo”, ocorreram devido às semelhanças visuais entre essas emoções. Uma possível solução seria aumentar o conjunto de dados com imagens que apresentem distinções mais claras entre essas expressões.

Em resumo, os modelos YOLOv10 se mostraram mais consistentes e adaptáveis aos testes, enquanto os modelos YOLOv8, embora eficazes no treinamento, enfrentaram desafios de generalização, principalmente devido à limitação do conjunto de dados de teste. Esses resultados sugerem que os modelos YOLOv10, especialmente o YOLOv10s, podem ser mais robustos para aplicações práticas. Contudo, para uma avaliação mais precisa, é recomendável expandir e diversificar o conjunto de testes, permitindo uma análise mais completa da capacidade de generalização dos modelos.

Como sugestões para trabalhos futuros, propõe-se aumentar tanto o conjunto de dados de treinamento quanto o de teste, visando proporcionar uma maior diversidade e melhorar a capacidade de generalização do modelo. Além disso, explorar arquiteturas alternativas, como

ResNet e outras Redes Neurais Convolucionais (CNNs), pode ser promissor, dada sua frequência em tarefas de classificação de imagens e reconhecimento facial.

REFERÊNCIAS

BETTADAPURA, Vinay. **Face expression recognition and analysis: the state of the art**. Atlanta (EUA), 30 de mar. 2012. Disponível em: <https://doi.org/10.48550/arXiv.1203.6722>. Acesso em: 14 de set. de 2024.

DENG, Jiankang *et al*. **RetinaFace: Single-stage dense face localisation in the wild**. 2 de maio de 2019. Disponível em: <https://doi.org/10.48550/arXiv.1905.00641>. Acesso em: 21 de set. de 2024.

EKMAN, Paul. Facial expression and emotion. **American psychologist**, v. 48, n. 4, p. 384, 1993. Disponível em: <https://scholar.google.com/scholar?hl=pt-BR&as_sdt=0%2C5&q=facial+expression+and+emotion+paul+ekman&btnG=&oq=facial+expression+and+emotion+paul+ekma>. Acesso em: 14 de set. de 2024.

GARG, Dweepna; GOEL, Parth; PANDYA, Sharnil; GANATRA, Amit. KOTECHA, Ketan. **A Deep Learning Approach for Face Detection using YOLO**. Pune (IN), 30 de nov. de 2018. Disponível em: <10.1109/PUNECON.2018.8745376>. Acesso em: 14 de set. de 2024.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning (Adaptive Computation and Machine Learning series). Edição nº 1. Cambridge (EUA): The MIT Press. 18 de nov. de 2016.

KHANZADA, Amil; BAI, Charles; CELEPCIKAY, Ferhat Turker. **Facial expression recognition with deep learning**. Stanford (EUA), 8 de abr. De 2020. Disponível em: <https://doi.org/10.48550/arXiv.2004.11823>. Acesso em: 13 de ago. de 2024.

MELO, Sara L.; MOURA, Fabio F. de; MACEDO, Kely; ALVES, Fabiano S. R; FERNANDES, Márcia A. Estudo comparativo de técnicas computacionais para classificação de emoções. **Anais do Simpósio Brasileiro de Informática na Educação (SBIE)**, São Paulo, v. 25, n. 1, p. 456, 2014. Disponível em: <https://doi.org/10.5753/cbie.sbie.2014.456>. Acesso em: 13 de ago. de 2024.

MITCHELL, Tom M. Machine Learning. Edição nº 1. Nova Iorque: McGraw-Hill Education. 01 de mar. de 1997.

RAJESH, K. M; NAVEENKUMAR, M. **A robust method for face recognition and face emotion detection system using support vector machines**. Mysuru (IN), 10 de dez. de 2016. Disponível em: <10.1109/ICEECCOT.2016.7955175>. Acesso em: 14 de set. De 2024.

REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. **You Only Look Once: Unified, Real-Time Object Detection**. 8 de jun. de 2015. Disponível em: <https://doi.org/10.48550/arXiv.1506.02640>. Acesso em: 14 de set. de 2024.

RIYANTOKO, Prismahardi Aji; SUGIARTO, K. M; HINDRAYIANI, Kartika Maulida. Facial emotion detection using Haar-cascade classifier and convolutional neural networks. **Journal of Physics: Conference Series**, 2021.

RUSSEL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach. Edição nº 3. Londres: Pearson. 8 de maio de 2020.

SHAIKH, Awais *et at*. Comprehensive Study on Emotion Detection with Facial Expression Images Using YOLO Models. **Journal of Information Assurance & Security**, v. 18, n 2, 2023.

URBANOWICK, Ryan J. (@DocUrbs). New proposed field/term Venn diagram for an upcoming talk. My take on illustrating the relationship between #DataScience, MachineLearning, ArtificialIntelligence , Statistics, and DataMining. Twitter, 14 de jun. 2018. Disponível em: <https://x.com/DocUrbs/status/1007375834347376642>. Acesso em: 13 de ago. 2024.

YOLOv8: THE LATEST VERSION OF THE YOLO MODEL. **Ultralytics**, 2023. Disponível em: <https://docs.ultralytics.com/>. Acesso em: 2 de set. de 2024.

ZHAO, Zhong-Qiu *et al*. Object detection with deep learning: A review. 15 de jul. de 2018. Disponível em: <https://doi.org/10.48550/arXiv.1807.05511>. Acesso em: 14 de set. de 2024.