

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
FACULDADE DE ENGENHARIA E INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM ENGENHARIA COMPUTACIONAL**

**Mathews Edwirds Gomes Almeida**

**Detecção de faces e regiões de interesse em imagens de bovinos por meio de  
redes neurais convolucionais**

Juiz de Fora

2023

**Mathews Edwirds Gomes Almeida**

**Detecção de faces e regiões de interesse em imagens de bovinos por meio de  
redes neurais convolucionais**

Monografia apresentada ao Curso de Graduação em Engenharia Computacional da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Bacharel em Engenharia Computacional.

Orientador: Prof. Dr. Luiz Maurílio da Silva Maciel

Juiz de Fora

2023

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Almeida, Mathews E. G..

Detecção de faces e regiões de interesse em imagens de bovinos por meio de redes neurais convolucionais / Mathews Edwirds Gomes Almeida. – 2023.

66 f. : il.

Orientador: Luiz Maurílio da Silva Maciel

Trabalho de Conclusão de Curso – Universidade Federal de Juiz de Fora, Faculdade de Engenharia e Instituto de Ciências Exatas. Bacharelado em Engenharia Computacional, 2023.

1. Detecção de objetos. 2. Redes Neurais. 3. Aprendizado Profundo. 4. Pecuária de Precisão. I. Maciel, Luiz Maurílio da Silva, orient. II. Título.

Mathews Edwirds Gomes Almeida

Detecção de faces e regiões de interesse em imagens de bovinos por meio de  
redes neurais convolucionais

Monografia apresentada ao Curso de Graduação em Engenharia Computacional da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Bacharel em Engenharia Computacional.

Aprovada em 15 de Junho de 2023

BANCA EXAMINADORA

---

Prof. Dr. Luiz Maurílio da Silva Maciel - Orientador  
Universidade Federal de Juiz de Fora

---

Dr. Bruno Campos de Carvalho  
Embrapa Gado de Leite

---

Prof. Dr. Marcelo Bernardes Vieira  
Universidade Federal de Juiz de Fora

---

Prof. Dr. Saulo Moraes Villela  
Universidade Federal de Juiz de Fora



Dedico este trabalho a todos que me apoiaram incansavelmente, compartilhando sabedoria e afeto. Essa conquista é fruto do apoio e determinação que floresceu em cada gesto de carinho ao longo desta jornada.

## AGRADECIMENTOS

Aos meus amados pais, que iluminaram o meu caminho desde o início. Agradeço por todo o amor incondicional, incentivo constante e confiança em minhas capacidades. Sem o amor e suporte de vocês, eu não teria chegado até aqui. Sou eternamente grato por tudo o que vocês fizeram e fazem por mim.

Às minhas queridas irmãs, vocês são muito importantes para mim. Agradeço por estarem ao meu lado, encorajando-me com palavras de ânimo e inspirando-me com sua determinação e apoio. Fico muito feliz por poder contar sempre com vocês.

À minha amada Ana, minha luz e maior fonte de inspiração. Sua presença amorosa, compreensão e encorajamento incansável me sustentaram nos momentos de dúvida e desafio. Seu amor incondicional me deu confiança para enfrentar os obstáculos, alcançar meus objetivos e seguir os meus sonhos. Sou grato todos os dias por ter você ao meu lado.

Aos meus orientadores de iniciação científica e monografia, Marcelo Bernardes e Luiz Maurílio, sou imensamente grato pela oportunidade de ser orientado de vocês, agradeço pela paciência, sabedoria e motivação que compartilharam comigo ao longo dos anos. Suas contribuições e críticas construtivas moldaram completamente este projeto, ampliando minha compreensão e incentivando meu crescimento acadêmico e profissional.

A meu melhor amigo, Brian Maia, obrigado por sua presença constante em minha vida. Você é o motivo de grande parte das minhas conquistas nos últimos anos. Você sempre esteve ao meu lado, apoiando minhas decisões, ouvindo minhas ideias e oferecendo conselhos sinceros. Eu não poderia imaginar essa caminhada sem você.

Agradeço também aos meus amigos Gabriel Rezende, Thaís Marins, João Victor e Luiz Paulo. Gabriel e Thaís, nossa parceria no projeto *Happy Cow ID* foi fundamental para minha formação acadêmica. A dedicação e disposição de vocês em trabalhar em equipe foram inestimáveis. João e Luiz, agradeço por serem amigos leais e confiáveis. Suas palavras de incentivo e disposição em ajudar foram inigualáveis.

Aos meus companheiros de trabalho, Felipe Eduardo Fischer e Rafael Lemos, agradeço por estarem presentes em minha vida profissional e acadêmica. Sua amizade, apoio e conselhos tornaram os dias de trabalho mais agradáveis e produtivos. Agradeço por compartilharem seus conhecimentos e experiências e também por sempre me incentivarem na vida pessoal, profissional e acadêmica.

A todos que mencionei e a todos aqueles que posso ter deixado de fora acidentalmente, meu coração transborda de gratidão. Este é um marco importante em minha vida, e vocês são parte essencial dessa conquista. Obrigado por acreditarem em mim, por me impulsionarem a ir além e por estarem ao meu lado em todos os momentos.

*“If we want machines to think, we need to teach them to see.” –  
(Fei-Fei Li)*

## RESUMO

A pecuária é uma atividade de grande importância econômica e social, e a tecnologia tem desempenhado um papel cada vez mais relevante neste setor. Nesse contexto, o presente trabalho se insere na intersecção entre a Pecuária de Precisão e a Visão Computacional. A motivação desse trabalho surge da necessidade de se identificar a face dos animais e as suas principais regiões de interesse para a solução de problemas como a classificação de expressões faciais associadas a dor ou desconforto. Sob essa perspectiva, é proposta a aplicação de modelos de detecção da face e suas regiões de interesse em imagens de bovinos. Para isso, uma das principais etapas deste trabalho foi a organização de um conjunto de dados anotado especificamente para este propósito, contendo imagens de bovinos em diferentes condições de iluminação, ângulos e cenários. Essa organização envolveu a coleta, filtragem e anotação manual de uma grande quantidade de imagens. O conjunto de dados resultante constitui uma importante contribuição deste trabalho. Subsequentemente, foram realizados estudos sobre redes neurais para o treinamento de modelos de detecção de objetos. Dessa forma, dois modelos de detecção foram treinados e avaliados a partir das redes neurais *SSD MobileNet V2 FPNLite 640 × 640* e *YOLOv8*. Em relação aos critérios de avaliação dos modelos, destaca-se que os experimentos atingiram resultados satisfatórios dado o problema abordado. A respeito dos gráficos traçados para cada modelo, foi possível observar que ambas as redes tiveram resultados semelhantes entre si e consistentes com as expectativas de cada gráfico. Por fim, a análise qualitativa reforçou os resultados obtidos com um estudo visual das detecções de ambas as redes em exemplos representativos do conjunto de teste. De forma geral, os resultados das análises demonstraram que ambos os modelos são capazes de detectar a face de bovinos e suas regiões faciais adequadamente, mesmo em condições adversas. Em conclusão, este trabalho demonstra o potencial das técnicas de aprendizado profundo para a detecção de objetos na Pecuária de Precisão. Os resultados obtidos são encorajadores e sugerem que essas redes podem ser utilizadas para melhorar a eficiência e a precisão das atividades de manejo e produção em meio rural. No entanto, ainda há desafios a serem superados, como a melhoria da robustez dos modelos e aumento de imagens em situações desafiadoras no conjunto de dados.

Palavras-chave: Detecção de objetos. Redes Neurais. Aprendizado Profundo. Pecuária de Precisão

## ABSTRACT

Livestock farming is an activity of great economic and social importance, and technology has played an increasingly important role in this sector. In this context, the present work is part of an intersection between Precision Livestock and Computer Vision. The motivation of this work arises from the need to identify the face of animals and its main regions of interest to solve problems such as the classification of facial expressions associated with pain or discomfort. From this perspective, it is proposed the application of models to detect face detection and its regions of interest in bovine images. To this end, one of the main steps of this work was the organization of a dataset annotated specifically for this purpose, containing images of cattle in different lighting conditions, angles and scenarios. This organization involved the manual collection, filtering and annotation of a large amount of images. The resulting dataset constitutes an important contribution of this work. Subsequently, studies were conducted on neural networks for training object detection models. Thus, two detection models were trained and evaluated from the neural networks SSD MobileNet V2 FPNLite  $640 \times 640$  and YOLOv8. Regarding the evaluation criteria of the models, the experiments achieved satisfactory results given the problem addressed. Regarding the plotted graphs for each model, it was possible to observe that both networks had similar results to each other and consistent with the expectations of each graph. Finally, the qualitative analysis reinforced the results obtained with a visual study of the detections of both networks on representative examples of the test set. Overall, the results of the analysis demonstrated that both models are able to detect the face of cattle and their facial regions adequately, even under adverse conditions. In conclusion, this work demonstrates the potential of deep learning techniques for object detection in Precision Livestock. The results obtained are encouraging and suggest that these networks can be used to improve the efficiency and accuracy of management and production activities in rural areas. However, there are still challenges to be overcome, such as improving the robustness of the models and increasing the number of images in challenging situations in the dataset.

Keywords: Object detection. Neural Networks. Deep Learning. Precision Livestock

## LISTA DE FIGURAS

Figura 1 – Exemplos de expressões faciais de bovinos com dor presente ou ausente.	14
Figura 2 – Exemplos de rotulação de objetos em imagens.	18
Figura 3 – Arquitetura da <i>R-CNN</i> .	20
Figura 4 – O sistema de detecção <i>YOLO</i> .	21
Figura 5 – O modelo de detecção <i>YOLO</i> .	22
Figura 6 – Arquitetura da <i>YOLO</i> .	22
Figura 7 – Arquitetura da <i>SSD</i> .	23
Figura 8 – Exemplo de detecção da rede <i>SSD</i> para objetos de diferentes proporções.	24
Figura 9 – Exemplo de cálculo das caixas de ancoragem.	25
Figura 10 – Exemplo de extração das caixas de ancoragem de uma imagem.	26
Figura 11 – Exemplos de imagens descartadas durante a etapa de filtragem do conjunto de dados.	31
Figura 12 – Exemplos de imagens do conjunto de treinamento.	33
Figura 13 – Exemplos de imagens do conjunto de teste.	33
Figura 14 – Exemplo de anotação de imagens do conjunto de dados.	35
Figura 15 – Distribuição das anotações por classe e por conjunto de dados.	37
Figura 16 – Arquitetura da <i>SSD MobileNet V2 300 × 300</i> .	38
Figura 17 – Cálculo da Interseção sobre União (IoU)	42
Figura 18 – Gráficos de Precisão-Confiança dos melhores modelos de ambas as redes.	51
Figura 19 – Gráficos de Revocação-Confiança dos melhores modelos de ambas as redes.	52
Figura 20 – Gráficos de Precisão-Revocação dos melhores modelos de ambas as redes.	53
Figura 21 – Gráficos de F1 pela Confiança dos melhores modelos de ambas as redes.	53
Figura 22 – Comparação das detecções de uma vaca adulta para ambas as redes.	55
Figura 23 – Comparação das detecções de bezerros para ambas as redes.	56
Figura 24 – Comparação das detecções de vacas adultas em uma imagem desfocada para ambas as redes.	57
Figura 25 – Comparação das detecções de um bezerro sob oclusão para ambas as redes.	57
Figura 26 – Comparação das detecções de uma vaca em posição lateral para ambas as redes.	58
Figura 27 – Comparação das detecções de um bezerro com desconforto para ambas as redes.	58

## LISTA DE TABELAS

Tabela 1	– Principais conjuntos de dados de imagens para detecção de objetos. . .	19
Tabela 2	– Palavras-chave utilizadas na coleta de dados para criação do conjunto de dados. . . . .	30
Tabela 3	– Informações detalhadas sobre os conjuntos de dados de treinamento e teste.	32
Tabela 4	– Informações sobre as anotações do conjunto de dados. . . . .	36
Tabela 5	– Configurações das máquinas. . . . .	41
Tabela 6	– Configurações e resultados dos experimentos com a <i>SSD Mobilenet</i> . . .	46
Tabela 7	– Descrição das técnicas de <i>data augmentation</i> da YOLOv8. . . . .	47
Tabela 8	– Configuração dos experimentos com a <i>YOLOv8</i> . . . . .	48
Tabela 9	– Resultados dos experimentos com a <i>YOLOv8</i> . . . . .	49

## LISTA DE ABREVIATURAS E SIGLAS

AP	<i>Average Precision</i>
CNN	<i>Convolutional Neural Network</i>
CUDA	<i>Compute Unified Device Architecture</i>
FN	<i>False Negatives</i>
FP	<i>False Positives</i>
FPS	Frames por segundo ( <i>Frames per second</i> )
GCG	Grupo de Computação Gráfica, Imagem e Visão
IoU	<i>Intesection over Union</i>
mAP	<i>mean Average Precision</i>
R-CNNs	<i>Region Based Convolutional Neural Networks</i>
RoI	<i>Region of interest</i>
SSD	<i>Single Shot MultiBox Detector</i>
TICs	Tecnologias da Informação e Comunicação
TN	<i>True Negatives</i>
TP	<i>True Positives</i>
YOLO	<i>You Only Look Once</i>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>12</b>
1.1	DEFINIÇÃO DO PROBLEMA . . . . .	15
1.2	OBJETIVOS . . . . .	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>17</b>
2.1	DETECÇÃO DE OBJETOS . . . . .	17
2.2	REDES NEURAIS PARA DETECÇÃO DE OBJETOS . . . . .	19
<b>2.2.1</b>	<b>Redes neurais convolucionais baseadas em regiões (<i>R-CNNs</i>) .</b>	<b>19</b>
<b>2.2.2</b>	<b><i>You Only Look Once (YOLO)</i> . . . . .</b>	<b>21</b>
<b>2.2.3</b>	<b><i>Single Shot MultiBox Detector (SSD)</i> . . . . .</b>	<b>23</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS . . . . .</b>	<b>28</b>
3.1	Conjunto de dados . . . . .	28
<b>3.1.1</b>	<b>Coleta de dados . . . . .</b>	<b>28</b>
<b>3.1.2</b>	<b>Anotação . . . . .</b>	<b>32</b>
3.2	MODELOS DE DETECÇÃO DE OBJETOS . . . . .	36
<b>3.2.1</b>	<b><i>SSD MobileNet V2 FPNLite 640 × 640</i> . . . . .</b>	<b>37</b>
<b>3.2.2</b>	<b><i>YOLOv8</i> . . . . .</b>	<b>39</b>
<b>4</b>	<b>EXPERIMENTOS E RESULTADOS . . . . .</b>	<b>41</b>
4.1	CONFIGURAÇÃO DOS EXPERIMENTOS . . . . .	41
4.2	CRITÉRIOS DE AVALIAÇÃO . . . . .	42
4.3	EXPERIMENTOS COM A <i>SSD MOBILENET V2 FPNLITE 640 × 640</i>	45
4.4	EXPERIMENTOS COM A <i>YOLOv8</i> . . . . .	47
4.5	DISCUSSÃO . . . . .	50
<b>4.5.1</b>	<b>Análise Quantitativa . . . . .</b>	<b>50</b>
<b>4.5.2</b>	<b>Análise Qualitativa . . . . .</b>	<b>54</b>
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>60</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>62</b>

## 1 INTRODUÇÃO

A pecuária é uma atividade econômica que consiste na criação e manejo de animais para a produção de alimentos ou matéria-prima, sendo classificada de acordo com a finalidade de produção, tais como leite, corte ou lã. A atividade pecuária, assim como a agricultura, desempenha um papel de extrema importância para a humanidade desde a pré-história. Ao longo do tempo, surgiram diversas técnicas de manejo e inovações tecnológicas que visam aumentar a eficácia da produção e gerenciamento do rebanho, tornando a pecuária uma atividade cada vez mais eficiente e sustentável (CARVALHO; ZEN, 2017).

No Brasil, a pecuária exerce um papel importante desde o período de colonização, tendo sido responsável pela expansão econômica do país tanto na linha de exportação, como também de abastecimento interno (TEIXEIRA; HESPANHOL, 2014). Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), o Brasil possui um rebanho de bovinos superior a 210 milhões de cabeças (IBGE, 2020). Esse cenário posiciona a pecuária de leite e corte como uma das principais atividades econômicas do país, colocando o Brasil como o segundo maior produtor de carne bovina do mundo (USDA, 2022).

Nos últimos anos, tem-se aplicado o conceito de “Precisão” à pecuária brasileira, originando o termo “Pecuária de Precisão”. Essa abordagem tem como objetivo maximizar o retorno econômico, promover o desenvolvimento sustentável e garantir a rastreabilidade da produção animal. Para alcançar esses objetivos, a Pecuária de Precisão tem se beneficiado de ferramentas e tecnologias de gerenciamento da variabilidade espacial e temporal. Essas ferramentas permitem o monitoramento contínuo das condições ambientais, nutricionais e de saúde dos animais ao longo do tempo e do espaço. Com base nessas informações, são implementadas estratégias específicas para cada situação, visando otimizar a qualidade de vida e produtividade dos animais (BASSOI et al., 2014).

As Tecnologias da Informação e Comunicação (TICs) compreendem todos os meios técnicos que podem ser usados para coletar, tratar e transferir informações, incluindo, por exemplo, o uso de computadores, *internet* e dispositivos móveis. De acordo com pesquisa do IBGE (2019), no triênio 2016-2019, houve um crescimento expressivo do acesso à rede e do uso de dispositivos móveis pelos brasileiros. Em linhas gerais, os índices mostram que o crescimento do uso das TICs ocorreu tanto em meio urbano quanto rural, independente da faixa etária ou condição socioeconômica da população. Dado o contexto da Pecuária de Precisão, o aumento da conectividade no campo pode favorecer a inserção das TICs na atividade, permitindo a coleta mais precisa e automatizada de informações de monitoramento dos animais e de fatores ambientais, além de auxiliar na tomada de decisões e no planejamento da produção. Essa interação entre pecuária e tecnologia também pode, portanto, abrir caminho para aprimorar a produtividade e a competitividade da pecuária

brasileira, impulsionando ainda mais a importância dessa atividade na economia do país (FERRAZ; PINTO, 2017).

A Visão Computacional é a área do conhecimento responsável por extrair e processar informações a partir de imagens ou vídeos. Entre as tarefas realizadas nessa área está o reconhecimento de padrões, que consiste em detectar e descrever os padrões de um objeto. Esses padrões podem ser reconhecidos em diversos tipos de dados como sinais com forma de onda, imagens, textos ou qualquer informação que necessite ser classificada/detectada (THEODORIDIS; KOUTROUMBAS, 2008). Na Pecuária de Precisão, o processamento e reconhecimento de padrões em imagens tem se mostrado uma ferramenta extremamente versátil e acessível, capaz de realizar tarefas como detecções, estimativas e classificações (SANTOS et al., 2020). Um exemplo de uso é o estudo realizado por Bezsonov et al. (2021), em que o reconhecimento de padrões em imagens foi usado para estimar a raça e o peso dos bovinos, além de monitorar a nutrição e o bem-estar do rebanho.

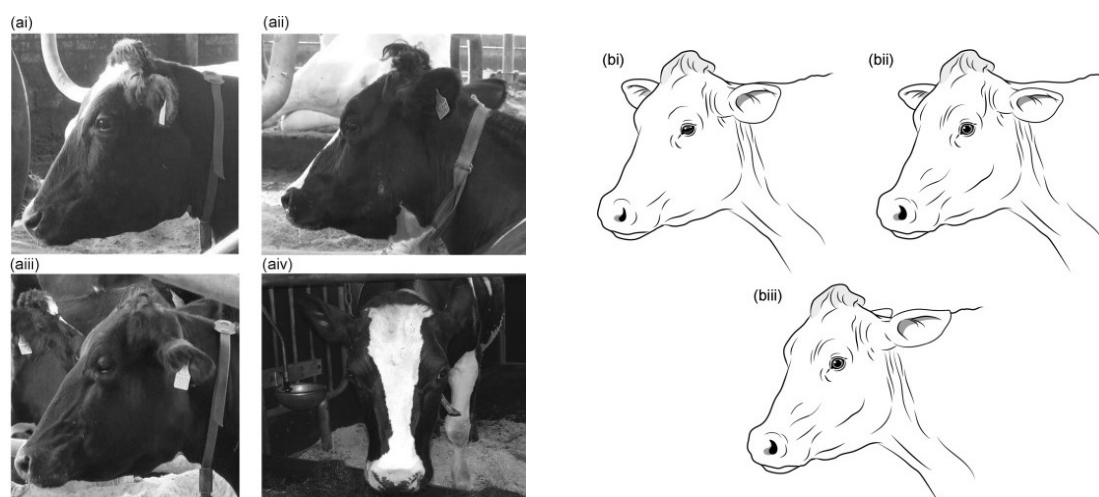
Especificamente, a detecção de objetos é uma tarefa de Visão Computacional que pode ser vantajosa para a Pecuária de Precisão, uma vez que ela pode servir como ferramenta de gerenciamento e monitoramento do rebanho. Por meio dela pode-se, por exemplo, realizar detecções de animais através de câmeras e sensores de movimento (SALAU; KRITTER, 2020), imagens térmicas de infravermelho (MEENA; AGILANDEESWARI, 2021) ou imagens aéreas (SHAO et al., 2020). Sob esse ponto de vista, sistemas de identificação, rastreamento e classificação baseados em redes neurais convolucionais (*Convolutional Neural Networks* – CNNs) podem ser beneficiados pela etapa de detecção de objetos, uma vez que imagens com menos plano de fundo representam mais adequadamente o objeto de interesse e aprimoram o desempenho dessas redes (KAMAL et al., 2021).

Além disso, a detecção de objetos pode ser aplicada em outras áreas da Pecuária de Precisão, como a identificação individual e o reconhecimento de expressões faciais em bovinos. Nesse sentido, Yao et al. (2019) propuseram uma abordagem baseada em redes neurais profundas que consegue identificar os animais com até 94.1% de precisão. Enquanto Neethirajan (2021) desenvolveu um sistema de reconhecimento de emoções em tempo real, baseado em redes de detecção de objetos, que consegue realizar a detecção de até nove estados emocionais de bovinos e suínos.

Ainda no contexto do reconhecimento de expressões faciais, a detecção de anomalias em áreas específicas da face dos animais pode indicar problemas de saúde ou desconforto. Nessa perspectiva, Glerup et al. (2015) propuseram a análise de vários indicadores físicos e comportamentais de bovinos, incluindo expressão facial, frequência respiratória e comportamento de vocalização. Em síntese, os resultados mostraram que a análise das expressões faciais é mais sensível para detectar dor do que outros parâmetros fisiológicos ou comportamentais, e que a combinação de diferentes indicadores é mais eficaz do que examinar cada um isoladamente. A Figura 1 apresenta imagens e ilustrações que

exemplificam como a expressão facial dos bovinos muda sob diferentes escalas de dor.

Figura 1 – Exemplos de expressões faciais de bovinos com dor presente ou ausente. (ai) Animal relaxado, sem dor. (aii) Animal com dor por manqueira. (aiii) Animal com dor por conta de sistema vascular comprometido, dor no úbere e poucos movimentos peristálticos semanais. (aiv) Animal com dor pós-cirúrgica (fistulação ruminal). (bi) Animal sem dor. (bii) Animal com dor e com orelhas baixas (orelhas de abano). (biii) Animal com dor, com as orelhas tensas e para trás.



(a) Imagens de bovinos sob diferentes escalas de dor e causas distintas.

(b) Ilustrações destacando as mudanças na expressão facial do animal.

Fonte: Adaptada de Gleerup et al. (2015).

A detecção precoce de dor em animais pode levar a um tratamento mais eficiente e, conseqüentemente, reduzir o sofrimento dos animais, além de aumentar a eficiência e produtividade da produção pecuária (KANG; ZHANG; LIU, 2020). Gleerup et al. (2015) evidenciam as 4 principais áreas da face do animal que podem fornecer informações importantes para classificar seu nível de dor ou desconforto:

1. Orelhas: orelhas tensas e para trás ou orelhas baixas;
2. Olhos: os olhos têm uma aparência tensa ou retraída. A tensão dos músculos acima dos olhos pode ser vista como “linhas de sulcos”;
3. Focinho: as narinas têm um aspecto “esticado” e podem haver linhas de expressão ao seu redor. Há aumento do tônus dos lábios;
4. Músculos faciais: tensão dos músculos faciais na lateral da cabeça.

Com base nesses indicadores, é possível considerar a viabilidade de um sistema baseado em Visão Computacional que possa detectar essas regiões e permitir o monitoramento dos animais em tempo real. A utilização desse sistema também poderia contribuir para a redução do estresse dos animais, uma vez que o monitoramento seria menos invasivo e menos estressante em comparação com os métodos tradicionais. Além disso, outra vantagem seria a redução da dependência da avaliação de observadores treinados, uma vez que elas são subjetivas e podem variar de acordo com a o grau de experiência de cada

observador (TSCHARKE; BANHAZI, 2016).

Considerando esse contexto, o presente trabalho busca desenvolver modelos de detecção capazes de identificar as principais regiões da face de bovinos, podendo ser altamente benéfico em sistemas de detecção/classificação de dor a partir de imagens e gravações dos animais, por exemplo. Esses modelos devem ser capazes de realizar detecções com precisão e confiança satisfatórias. Ao mesmo tempo, este trabalho foi conduzido prezando a concepção de modelos leves e rápidos no intuito de que seja possível embarcá-los em sistemas/aplicações de tempo real. Para tanto, fez-se necessária a construção e rotulação de um conjunto de dados (*dataset*) de imagens de bovinos, bem como o treinamento de redes de detecção de objetos. *A priori*, espera-se que este trabalho possa contribuir em diversos problemas relacionados à Pecuária de Precisão e Visão Computacional que necessitem da etapa de detecção e extração dos objetos de interesse a partir de imagens. O que, conseqüentemente, também pode trazer melhorias na produtividade, qualidade de vida e bem-estar dos animais.

## 1.1 DEFINIÇÃO DO PROBLEMA

O problema a ser resolvido neste trabalho consiste em, dada uma imagem, determinar a localização da face de bovinos e as suas principais regiões de interesse, como os olhos, orelhas e focinho. Para cada objeto detectado, é necessário determinar as coordenadas que representam a caixa de delimitação (*bounding box*) e sua respectiva classe. É importante destacar que as imagens onde os objetos serão detectados podem apresentar variações na iluminação, ângulo, posição, plano de fundo e outras características, o que torna a tarefa de detecção ainda mais desafiadora. Para abordar esse problema, serão utilizadas técnicas de detecção de objetos baseadas em redes neurais profundas, visando desenvolver modelos capazes de lidar com esses obstáculos e localizar com precisão as regiões de interesse buscadas.

## 1.2 OBJETIVOS

O objetivo geral deste trabalho é desenvolver modelos de detecção capazes de identificar regiões de interesse (face, olhos, orelhas e focinho) em imagens de bovinos não padronizadas (ambiente não controlado). Para isso, foram utilizadas técnicas de aprendizado profundo em redes neurais de detecção de objetos. Isto posto, os objetivos específicos são:

- Estudar modelos de detecção de objetos baseados em aprendizado profundo;
- Organizar um conjunto de dados de imagens que possuam bovinos e suas principais regiões de interesse;

- Rotular/anotar adequadamente todas as regiões de interesse em cada uma das imagens;
- Treinar diferentes redes de detecção de objetos e gerar modelos de detecção;
- Avaliar os modelos no conjunto de imagens de teste em relação a critérios de avaliação específicos para o problema de detecção de objetos;
- Comparar quantitativa e qualitativamente os resultados dos modelos de detecção.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os principais conceitos e estudos que fundamentam o tema de pesquisa deste trabalho, visando fornecer uma base sólida e consistente para o seu desenvolvimento. É fornecida uma explicação detalhada sobre a tarefa de detecção de objetos em imagens, abordando desde a sua definição até os seus conjuntos de imagens mais relevantes. Além disso, apresenta-se um detalhamento das principais redes de detecção de objetos, incluindo informações sobre o seu funcionamento e dados de entrada e saída. Destaca-se também a eficiência dessas redes, visto que a capacidade de processamento em tempo real é um aspecto crucial em aplicações práticas de detecção de objetos.

### 2.1 DETECÇÃO DE OBJETOS

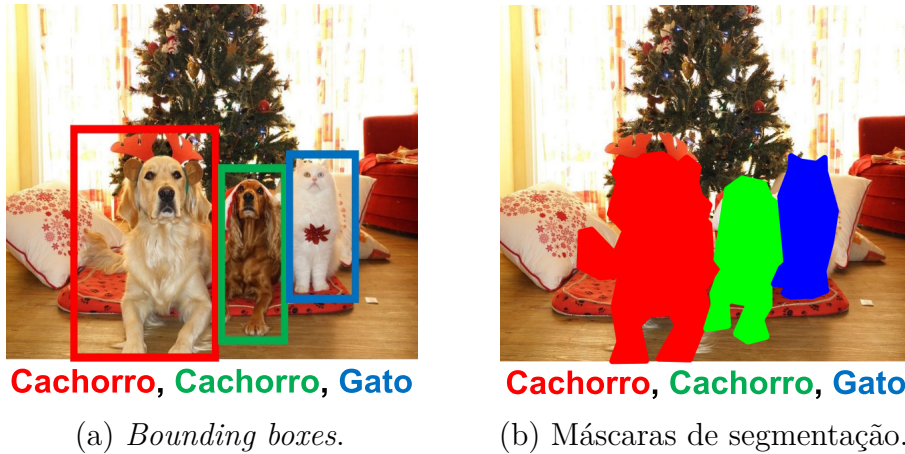
Dada uma imagem digital, o objetivo da detecção de objetos é localizar uma ou mais instâncias de determinados objetos e estimar suas classes corretamente. Essa detecção deve levar em consideração que um mesmo objeto pode ser apresentado em diferentes escalas, orientações, localizações e perspectivas na imagem (AMIT; FELZENSZWALB; GIRSHICK, 2020). Os tipos de objetos a serem detectados podem ser gerais como carros, prédios e pessoas ou específicos como placas de veículos, faces de animais ou pessoas usando máscara, por exemplo.

Para o treinamento de um modelo de detecção, conjuntos de dados de imagens de exemplo contendo os objetos de interesse devem ser fornecidos. Nessas imagens deve-se realizar o processo de rotulação, no qual as informações de localização e classe associada a cada objeto são anotadas/rotuladas (ADHIKARI; HUTTUNEN, 2021). A anotação dessas informações geralmente é representada na forma de caixas de delimitação (Figura 2a) ou de máscaras de segmentação (Figura 2b). As caixas de delimitação são o tipo de representação mais popular de anotação, no qual um retângulo é demarcado com a menor área possível ao redor do objeto. Enquanto as máscaras de segmentação são um conjunto de *pixels* da imagem que formam o objeto de interesse (*Region of Interest* – RoI) (HE et al., 2017). Essas anotações são conhecidas na literatura como *ground truth*.

Segundo Lemoigne e Caner (2008), o *ground truth* pode ser visto como o conceito de se conhecer a verdade sobre determinado tema, sendo também considerado como a solução ideal ou exata que se espera de um problema. Dessa forma, na detecção de objetos, a demarcação dos *ground truths* é responsável não somente por caracterizar a natureza do objeto que se pretende detectar, mas também por possibilitar que as detecções obtidas possam ser comparadas com a demarcação do objeto original. Conseqüentemente, essa comparação permite avaliar a qualidade das detecções e obter perspectivas sobre como aprimorá-las.

Para que a detecção de objetos seja eficiente em uma grande variedade de cenários,

Figura 2 – Exemplos de rotulação de objetos em imagens.



Fonte: Adaptada de Michelucci (2019).

é indispensável que a quantidade de imagens do conjunto de dados de treinamento seja abundante e que as informações nele estruturadas sejam relacionadas ao problema que se deseja abordar. Isso pode ajudar o modelo a capturar diversos aspectos de variabilidade dos objetos, permitindo uma melhor generalização diante de diferentes características como a iluminação, pose, orientação, escala e perspectiva (AMIT; FELZENSZWALB; GIRSHICK, 2020). Nesse sentido, muitos conjuntos de dados vêm sendo construídos para abordar diversos problemas de detecção, alguns dos mais populares são elencados na Tabela 1. Esses conjuntos, em sua maioria, apresentam uma diversificada representatividade em cada uma de suas categorias e uma extensa quantidade de classes e imagens. Dessa forma, acredita-se que eles podem ser capazes de conceber uma melhor generalização e entendimento dos objetos de interesse. No entanto, observa-se que mesmo existindo conjuntos de dados com quantidades tão expressivas de imagens, o número de conjuntos preparados para lidar com problemas mais específicos ainda é pequeno. Na atualidade, esse é um dos principais desafios para o desenvolvimento de projetos ligados à área de Visão Computacional e detecção de objetos (JIAO et al., 2019).

Grande parte das redes neurais para as tarefas de detecção de objetos são previamente treinadas com os conjunto de dados apresentados na Tabela 1. Uma vez pré-treinada, a rede pode então ser modificada e aprimorada para a tarefa final pretendida inicialmente (SZELISKI, 2022). Esse processo é conhecido como transferência de aprendizado (*transfer learning*) e é útil não somente para reduzir o tempo e custo computacional de treinamento de novos modelos de detecção, mas também para permitir que seja possível treinar com um número reduzido de amostras e classes para um determinado problema. Mais detalhes das principais redes de detecção de objetos serão explorados na próxima seção.



Tabela 1 – Principais conjuntos de dados de imagens para detecção de objetos.

Conjunto de dados	Qtd. imagens	Núm. classes	Tipo de anotação	Principais categorias
<i>BDD100K</i> (YU et al., 2020)	100.000	40	<i>bounding boxes</i> e segmentação	Pedestres, veículos e semáforos
<i>Dota</i> (DING et al., 2021)	11.268	18	<i>bounding boxes</i>	Lugares e veículos
<i>ImageNet</i> (DENG et al., 2009)	1.431.137	21.000	<i>bounding boxes</i>	Geral
<i>Google’s Open Images V6</i> (KUZNETSOVA et al., 2020)	9.000.000	6.000	<i>bounding boxes</i> e segmentação	Animais, lugares, objetos e pessoas
<i>Microsoft Coco</i> (LIN et al., 2014)	328.000	91	<i>bounding boxes</i> e segmentação	Animais, objetos e pessoas
<i>PASCAL VOC 2010</i> (EVERINGHAM et al., 2010)	10.103	20	<i>bounding boxes</i> e segmentação	Animais, pessoas e veículos
<i>WIDER FACE</i> (YANG et al., 2016)	32.203	61	<i>bounding boxes</i>	Faces humanas

Fonte: Elaborada pelo autor (2023).

## 2.2 REDES NEURAIAS PARA DETECÇÃO DE OBJETOS

Em contraste com a atualidade, o alicerce da detecção de objetos foi fundado com base em características construídas à mão (*handcrafted features*) devido à carência de dados em forma digital e a limitação dos recursos computacionais da época (ZOU et al., 2019). *Handcrafted features* referem-se ao uso de filtros para detecção de bordas, cores, gradientes ou frequências e que têm a finalidade de extrair as principais características de uma imagem e, a partir disso, realizar tarefas de detecção e reconhecimento de objetos (JIANG et al., 2019). Esses algoritmos são denominados “descritores de características”, sendo *SIFT* (LOWE, 2004), *HOG* (DALAL; TRIGGS, 2005) e *Haar* (VIOLA; JONES, 2004), alguns de seus exemplos mais conhecidos.

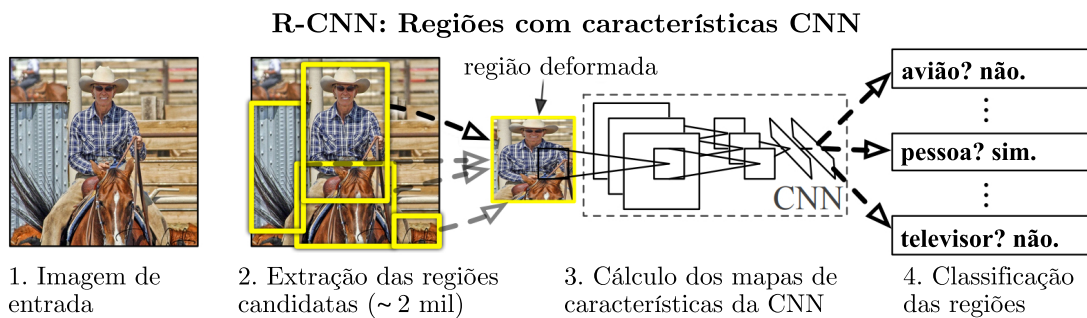
As abordagens atuais se fundamentam no aprendizado profundo (*Deep Learning*), no qual as características mais importantes de cada objeto em uma imagem são “aprendidas” pelo modelo. Essas abordagens levam em consideração dois tipos principais: a detecção em dois estágios (*two-stage*) e a detecção em um estágio (*one-stage*) (ZOU et al., 2019). A primeira é apoiada no uso de modelos de aprendizado profundo e *CNNs* para amostrar e classificar em listas as possíveis regiões de interesse de uma imagem. Enquanto a segunda converte esse problema de amostragem em um problema de regressão, no qual não é mais necessário gerar listas de regiões candidatas (KHAN; KHAN; ANSARI, 2022). Nas subseções seguintes, serão descritos exemplos de redes neurais de ambos os gêneros.

### 2.2.1 Redes neurais convolucionais baseadas em regiões (*R-CNNs*)

Na perspectiva da detecção de objetos em dois estágios, redes da família das *R-CNNs* (GIRSHICK et al., 2014) se destacam por serem pioneiras na utilização de *CNNs* no

contexto da detecção de objetos. Essas redes, de modo geral, foram de grande importância para contornar problemas ligados à presença de múltiplos objetos em uma imagem. Para isso, as *R-CNNs* introduziram a geração de 2.000 regiões de interesse candidatas por amostra, o recorte e deformação de regiões, a classificação com máquinas de vetores de suporte (*Support Vector Machines – SVMs*) e a regressão de *bounding boxes*. Destarte, essas redes se destacaram como estado da arte em comparação com outras redes de detecção desenvolvidas na época. A Figura 3 ilustra a arquitetura da *R-CNN* original.

Figura 3 – Arquitetura da *R-CNN*.



Fonte: Adaptada de Girshick et al. (2014).

De forma geral, o processo de treinamento de uma *R-CNN* envolve cinco etapas: a busca seletiva de todas as regiões candidatas da imagem, a aplicação de cortes/distorções para normalizar as dimensões de cada região, a extração de características com CNNs, a classificação dos objetos detectados com SVMs e, por fim, o ajuste dos *bounding boxes* por meio de um regressor linear. As saídas dessa rede são a classe, a confiança, e o *bounding box* para cada objeto detectado. No entanto, por ser necessário gerar três modelos separadamente e normalizar cada região das imagens, a *R-CNN* original acabou se tornando uma solução lenta e custosa tanto em relação aos elevados recursos computacionais quanto pelo volume de dados necessário para o treinamento. Além disso, observa-se também que a etapa de normalização leva à perda de informações valiosas da imagem (DU, 2018).

Para resolver as limitações de desempenho da *R-CNN*, outras redes como a *Fast R-CNN* (GIRSHICK, 2015) e a *Faster R-CNN* (REN et al., 2015) foram desenvolvidas. Essas novas redes, em suma, focaram em unificar os modelos gerados ao longo do treinamento e em aumentar o compartilhamento de informações aprendidas por suas camadas convolucionais. Assim, em vez de extrair as características de cada região de interesse repetidas vezes, esse novo modelo passou a percorrer cada imagem apenas uma vez (ainda assim gerando centenas de regiões candidatas), o que acelerou o processo de treinamento e detecção. Essas melhorias levaram a *Faster R-CNN* a alcançar uma melhor precisão e a conseguir prever objetos em até 5 *frames* por segundo (FPS) (DU, 2018). Por fim, a *Mask R-CNN* (HE et al., 2017) adaptou a *Faster R-CNN* para a tarefa de segmentação de imagens com máscaras de segmentação. Isso foi possível com a adição de um terceiro ramo de predição em paralelo com os ramos de localização e classificação de regiões candidatas. Assim, a

*Mask R-CNN* foi capaz de prever não somente a classe, o *bounding box* e a confiança para cada região de interesse, mas também a posição a nível de pixel do objeto encontrado na imagem (ZHANG et al., 2021).

### 2.2.2 *You Only Look Once (YOLO)*

Ao invés de selecionar múltiplas regiões candidatas por imagem, as redes de detecção de objetos em um estágio analisam a imagem por completo e tentam detectar um determinado número de objetos e classes em apenas uma execução do algoritmo. Por esse motivo, a *YOLO* (REDMON et al., 2016) e suas variações *YOLOv2* (REDMON; FARHADI, 2017), *YOLOv3* (REDMON; FARHADI, 2018), entre outras, são algumas das redes comumente usadas para detecção de objetos em tempo real. Isso se deve ao fato de que redes desse gênero possuem desempenho muito superior às redes baseadas em dois estágios ao custo de pouca redução na acurácia/confiança das detecções (DU, 2018).

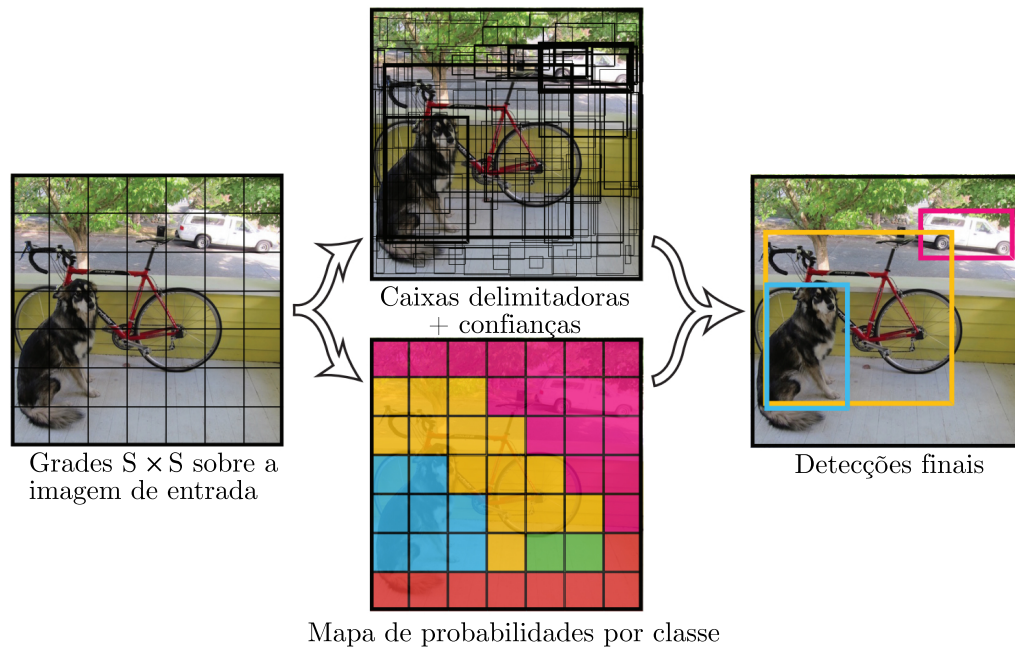
O sistema de detecção da *YOLO* (Figura 4) pode ser resumido em 3 etapas: o redimensionamento das imagens, a detecção e classificação dos *bounding boxes* a partir de CNNs e o pós-processamento das detecções. A primeira etapa é responsável por normalizar as dimensões da imagem para  $448 \times 448$ . Já a segunda etapa é responsável por produzir as caixas delimitadoras que identificam os objetos de interesse na imagem. Por fim, o algoritmo de supressão não-máxima (*non-maximum supression*) tem a função de filtrar *bounding boxes* de mesma classe que estejam sobrepostos acima de um limiar de sobreposição. Por consequência, a precisão do modelo é melhorada, uma vez que apenas as detecções mais confiáveis serão mantidas (REDMON et al., 2016).

Figura 4 – O sistema de detecção *YOLO*.



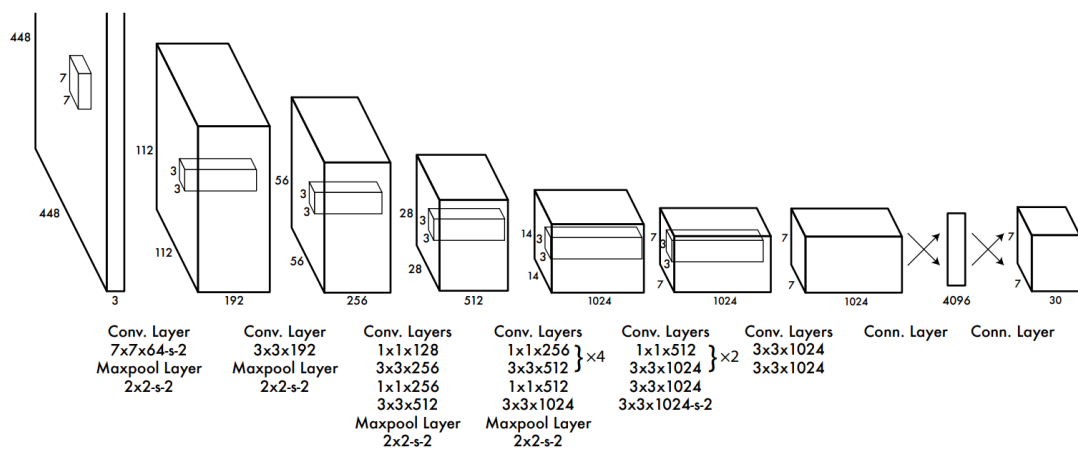
Fonte: Adaptada de Redmon et al. (2016).

O funcionamento do modelo de detecção da *YOLO* se dá pela divisão das imagens em grades de tamanho  $S \times S$ . Cada célula da grade tenta prever o objeto que está dentro dela. Ao encontrar um objeto, são delimitados  $B$  *bounding boxes* ao redor do mesmo, um vetor de probabilidades  $C$  por classe e valores de *offset* para cada *box*. A partir de um limiar (*threshold*), as células que possuem as maiores probabilidades são selecionadas e usadas para delimitar o objeto na imagem (REDMON et al., 2016). Todo esse processo pode ser visualizado na Figura 5.

Figura 5 – O modelo de detecção *YOLO*.

Fonte: Adaptada de Redmon et al. (2016).

Em síntese, a arquitetura da *YOLO* é composta de 24 camadas convolucionais e duas camadas completamente conectadas. Além disso, camadas convolucionais  $1 \times 1$  são aplicadas de forma alternada para reduzir a quantidade de características provenientes das camadas anteriores e melhorar a performance da rede. A Figura 6 mostra a arquitetura dessa rede de forma mais detalhada. A *YOLO* recebe como entrada imagens de tamanho fixado em  $448 \times 448$  *pixels*. Como saída, múltiplos *bounding boxes* com suas respectivas classes e probabilidades são gerados de forma simultânea.

Figura 6 – Arquitetura da *YOLO*.

Fonte: Redmon et al. (2016).

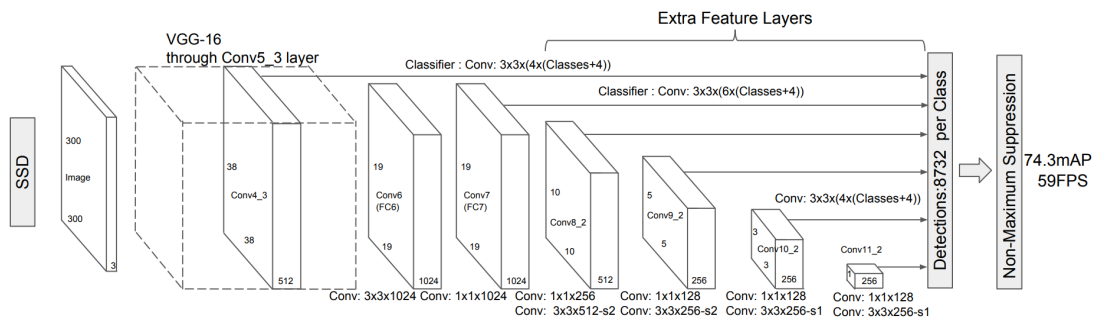
Por fim, destaca-se que a rede *YOLO* em sua versão original apresenta vantagens e desvantagens em relação a outros detectores de objetos da época. Entre as vantagens, pode-se destacar a simplicidade da arquitetura, a divisão das imagens de treinamento em grades

e a aplicação de técnicas de pré e pós-processamento, que contribuíram positivamente para o bom desempenho do detector. Além disso, a *YOLO* original é mais rápida (45 FPS) e produz menos falsos positivos que a *Faster R-CNN*. Porém, sua arquitetura apresenta desvantagens quando se trata de detectar objetos muito pequenos (baixa proporção do objeto em relação ao tamanho da imagem) ou muito próximos (as bordas do objeto não estão presentes na imagem) e também tende a cometer mais erros de localização que a *Fast R-CNN* (REDMON et al., 2016).

### 2.2.3 *Single Shot MultiBox Detector (SSD)*

A *SSD* (LIU et al., 2016) foi a segunda rede de detecção de objetos em um estágio da literatura. Assim como a *YOLO*, ela também é comumente utilizada em aplicações de tempo real por conta de sua velocidade de processamento e boa capacidade de lidar com imagens de alta resolução. Os principais componentes dessa rede são: um modelo de classificação (*backbone*) e o detector *SSD*. Sua arquitetura original (Figura 7) é baseada na rede pré-treinada de classificação *VGG16* (SIMONYAN; ZISSERMAN, 2014) com adição de várias camadas convolucionais de diferentes dimensões em ordem decrescente. Além disso, outras redes como a *MobileNet* (HOWARD et al., 2017) e a *ResNet* (HE et al., 2016) também podem ser usadas como modelos de classificação ao invés da *VGG16*. Para a *SSD*, o papel principal de seu *backbone* é a extração de características enquanto o seu detector é responsável por produzir mapas de características de múltiplas escalas e gerar os *bounding boxes* e suas respectivas classes. As principais contribuições da *SSD* podem ser listadas como: o uso de pequenos filtros convolucionais para predizer classes de objetos e alinhar a posição de seus *bounding boxes*, a separação de *boxes* por sua razão de aspecto (*aspect ratio*) e a geração de representações dos objetos para aprendizado a partir de diversas escalas (ZOU et al., 2019).

Figura 7 – Arquitetura da *SSD*.



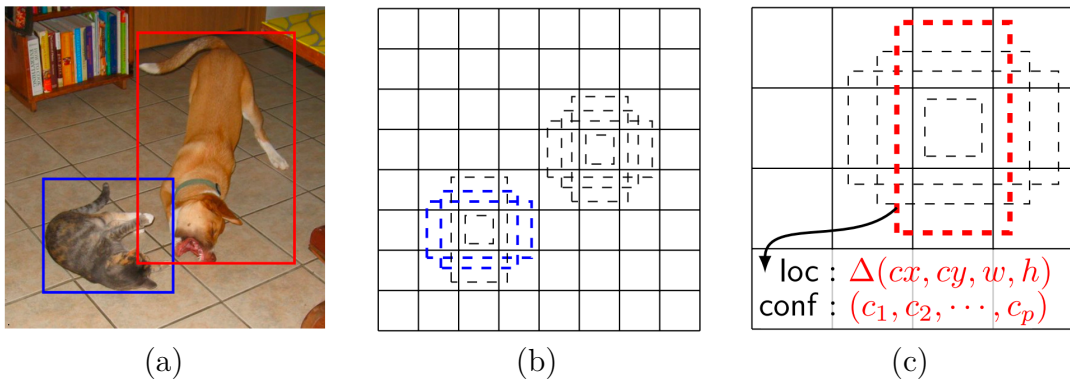
Fonte: Adaptada de Liu et al. (2016).

As camadas convolucionais adicionadas pela *SSD* à *VGG16* podem ser vistas como representações piramidais das imagens de entrada sob diferentes escalas. Essa construção busca garantir que tanto objetos de escala reduzida quanto objetos de grandes proporções possam ser detectados pela rede, o que a torna invariante espacialmente. A ideia de usar

essas representações sob múltiplas escalas vem da intuição de que as camadas iniciais da rede são mais propensas a identificar pequenos objetos enquanto as camadas finais encontram objetos maiores com mais facilidade. Na *SSD*, a detecção ocorre em todas as camadas piramidais, buscando nelas objetos de diversos tamanhos (WENG, 2018).

Em contraste com a *YOLO*, a *SSD* não divide a imagem de entrada em grades de tamanho fixo. Entretanto, ela tenta prever suas caixas de ancoragem (*anchor boxes*) para todas as localizações do mapa de características (*feature map*) da imagem. Resumidamente, as *anchor boxes* nada mais são do que uma coleção de caixas dotadas de informações e dimensões pré-definidas as quais a rede pressupõe serem as mais adequadas para detectar determinado objeto (WENG, 2018). Assim, para cada nível diferente da rede, essas caixas de ancoragem são redimensionadas de modo que mapas de características de tamanho menor (baixa dimensão) são responsáveis por encontrar objetos maiores enquanto mapas de tamanho maior (maior dimensão) encontram objetos menores. Um exemplo de detecção deste método pode ser visto na Figura 8, onde são mostrados dois objetos (Figura 8a) e seus respectivos mapas de características e vetores de localizações e confianças (Figura 8b e Figura 8c).

Figura 8 – Exemplo de detecção da rede *SSD* para objetos de diferentes proporções. (a) Imagem com demarcação de *ground truths*. (b) Mapa de características  $8 \times 8$ . (c) Mapa de características  $4 \times 4$ .



Fonte: Adaptada de Liu et al. (2016).

Como as outras redes de detecção de objetos abordadas anteriormente, a *SSD* também tem como entrada imagens anotadas com dados sobre a posição e classe de cada objeto. Durante o treinamento, a rede é exposta a esses dados e tem seus erros de confiança da detecção (*confidence loss* –  $L_{conf}$ ) e localização dos objetos (*localization loss* –  $L_{loc}$ ) calculados. Esses critérios servem para mostrar, respectivamente: o quão confiante a rede se comporta ao classificar o objeto dentro de um *bounding box* e o quão precisa é a localização da caixa predita em relação ao que ela deveria prever (WENG, 2018). A combinação de ambas as perdas permite a formulação de uma função de custo capaz de minimizar o erro entre as caixas delimitadoras preditas pela rede e as caixas delimitadoras reais (*ground truths*) presentes nas imagens anotadas e usadas na etapa de treinamento.



Assim, essa equação deve considerar os parâmetros:  $x$  que representa a classe do objeto detectado na imagem,  $c$  que representa a confiança da rede na classificação correta do objeto,  $l$  que representa a caixa delimitadora prevista pela rede para o objeto e  $g$  que representa o *ground truth*. Dessa maneira, obtém-se a equação da perda global:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)),$$

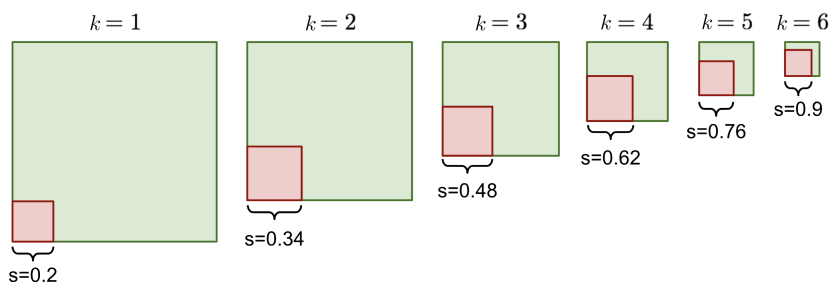
onde  $N$  é a quantidade de caixas de ancoragem correspondente e  $\alpha$  é a constante que estipula o peso da perda de localização sobre essa função. Deste modo, a perda de localização é definida por uma função de perda  $L_1$  suave (*Smooth  $L_1$  Loss* – Girshick (2015)) sobre os parâmetros de localização das caixas previstas com as localizações dos *ground truths* da imagem. Por último, a perda de confiança é calculada a partir de uma função de perda *softmax* sobre múltiplas classes, a qual também é conhecida por “*cross entropy*” ou “entropia cruzada”. Essa perda mede a diferença entre as confianças das classes previstas e as anotações dos *ground truths*.

Antes da detecção propriamente dita, a rede passa pelos processos de normalização de caixas de ancoragem, mineração de negativos difíceis (*hard negative mining*) e aumento de dados (*data augmentation*). O primeiro processo é realizado de modo a tornar as detecções mais precisas e eficientes. Isso é feito através da utilização de uma escala proporcional para cada posição  $(i, j)$  do mapa de características  $(m \times n)$ . Assim, cada escala é determinada com base no nível de profundidade da camada convolucional. Além disso, cada posição do mapa de características pode ter até 5 outras escalas diferentes que devem estar igualmente espaçadas, de acordo com o *aspect ratio* da caixa de ancoragem. A escala para cada uma dessas caixas é calculada a partir da equação:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m - 1}(k - 1), k \in [1, m],$$

onde  $k$  se refere ao índice da camada piramidal que está sendo normalizada e  $s_{min} = 0,2$  e  $s_{max} = 0,9$  representam a menor e a maior escala, respectivamente. A Figura 9 demonstra as escalas dessas caixas para cada camada  $k$  da rede.

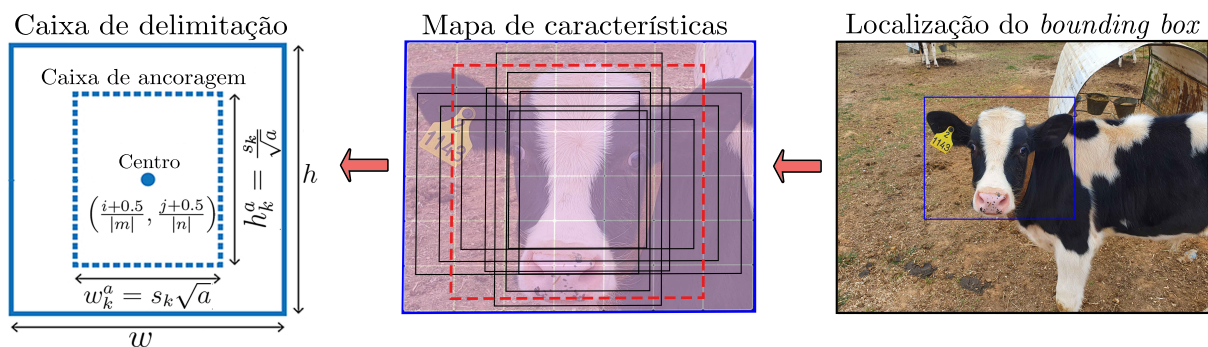
Figura 9 – Exemplo de como as caixas de ancoragem são calculadas para cada camada  $k$ . São consideradas 6 camadas com  $s_{min} = 0,2$ ,  $s_{max} = 0,9$  e *aspect ratio* 1 (caixa quadrada).



Fonte: Adaptada de Weng (2018).

Ademais, para cada camada piramidal, são calculados: a razão de aspecto, altura, largura e coordenadas do ponto central das caixas de ancoragem. O *aspect ratio* é denotado por  $a \in \left\{1, 2, 3, \frac{1}{2}, \frac{1}{3}\right\}$  e seu cálculo parte da relação da altura pela largura das caixas de ancoragem. Por fim, também são denotadas a altura ( $h_k^a = \frac{s_k}{\sqrt{a}}$ ), a largura ( $w_k^a = s_k\sqrt{a}$ ) e as coordenadas dos pontos centrais  $\left(\frac{i+0.5}{|m|}, \frac{j+0.5}{|n|}\right)$  para cada caixa. Esses cálculos são imprescindíveis uma vez que são fundamentais para otimizar a localização das caixas de ancoragem em relação aos objetos na imagem. A Figura 10 ilustra o cálculo das proporções de uma *anchor box* e sua influência sobre o mapa de características.

Figura 10 – Exemplo de extração das caixas de ancoragem de uma imagem. À esquerda estão dispostos os cálculos da altura, largura e coordenadas do ponto central de uma caixa de ancoragem. Essas medidas são fundamentais para determinar a localização e *aspect ratio* dos *bounding boxes* sobre determinado objeto na imagem. Ao centro tem-se o mapa de características, que é o espaço de busca da detecção. Destaca-se de vermelho a caixa de ancoragem escolhida pela rede. Por fim, à direita, existe um bovino com sua face (objeto/classe de interesse) destacada.



Fonte: Elaborada pelo autor (2023).

Durante o processo de treinamento, é comum enfrentar dificuldades na geração de caixas de ancoragem precisas. Isso se deve ao fato de que a quantidade de caixas geradas é sempre significativamente maior do que a quantidade de objetos presentes na imagem. Isso suscita um severo desbalanceamento entre as amostras positivas e negativas, além de influenciar na deterioração das detecções do modelo. Outro problema é que boa parte dessas amostras pode conter objetos de forma parcial ou texturas ruidosas, o que pode levar a confusões da rede de detecção. Para contornar esses problemas, ao invés de serem utilizadas todas as amostras negativas do conjunto de dados, são mantidas apenas as amostras negativas de maior custo/menor confiança de modo a manter uma razão entre negativos e positivos de pelo menos 3:1. Essa técnica é conhecida como *hard negative mining* e é capaz de acelerar a otimização, estabilizar o treinamento da rede e reduzir as confusões do modelo (LIU et al., 2016).

Para que fosse possível treinar um modelo mais robusto e sensível a diferentes tamanhos e formatos de objetos, Liu et al. (2016) propuseram também a aplicação randômica de técnicas de aumento de dados sobre as imagens de treinamento. Algumas delas são: o uso da imagem inteira como entrada da rede, a amostragem de uma região



(*patch*) em torno do objeto sob determinados níveis de sobreposição (0,1; 0,3; 0,5; 0,7 ou 0,9) e a amostragem de *patches* de forma randômica. Esses *patches* são gerados com *aspect ratios* entre  $\frac{1}{2}$  e 2, são redimensionados para tamanhos pré-definidos e também são espelhados horizontalmente com uma probabilidade de 0,5. Além desses *data augmentations* outras técnicas podem ser aplicadas, conforme será discutido nas Seções 4.3 e 4.4.

Por fim, destaca-se que a *SSD* é capaz de detectar objetos em imagens com alta precisão e em tempo real (até 59 FPS), o que a torna adequada para aplicações em que é importante detectar objetos rapidamente, como em sistemas de vigilância (NING et al., 2019) ou em sistemas de assistência à condução (ZHAO et al., 2018). Além disso, a rede é robusta e capaz de detectar objetos em condições adversas ao mesmo tempo em que remove dados redundantes ao longo de suas etapas de processamento (JABBAR et al., 2022). No entanto, a *SSD* é relativamente complexa e requer um grande conjunto de dados para o treinamento, o que pode dificultar sua implementação em algumas situações (LIU et al., 2016). Além disso, dependendo do *backbone* escolhido, a rede pode se tornar exigente em termos de hardware e, portanto, pode não ser adequada para todas as situações. Quando comparada a outras redes de detecção de objetos, como a rede *YOLO* e *R-CNN*, a *SSD* se destaca tanto por detectar objetos de cinco escalas distintas em diferentes camadas quanto por sua alta precisão e velocidade, mesmo em objetos de tamanho reduzido.

### 3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados a construção do conjunto de dados para treinamento e teste e os modelos de detecção treinados. Também são explicitados os principais desafios e decisões tomadas durante o desenvolvimento geral do trabalho.

#### 3.1 Conjunto de dados

Um dos principais desafios enfrentados no desenvolvimento deste trabalho foi a criação de um conjunto de dados capaz de atender aos requisitos e objetivos do projeto, visto que não foram encontrados conjuntos significativos e que contassem com os rótulos das regiões de interesse da face de bovinos. Para superar essa limitação, optou-se pela coleta de dados, que se tornou uma etapa crítica da construção deste conjunto. Essa abordagem apresentou diversas vantagens em relação ao uso de conjuntos de dados prontos, como a possibilidade de se adquirir uma grande quantidade de dados e de adaptá-los à medida que o projeto e a metodologia evoluíam.

Sendo assim, após esse processo de curadoria, foi necessário anotar manualmente cada uma das imagens adquiridas. Essa tarefa consistiu em demarcar as regiões de interesse nas imagens e atribuir seus rótulos. Tais anotações são essenciais para a etapa de treinamento dos modelos, uma vez que elas são usadas para associar cada objeto de interesse a uma determinada classe. Embora a tarefa em si seja considerada simples, é necessário ressaltar que ela demanda uma grande quantidade de tempo e é altamente repetitiva, o que a torna bastante custosa em termos de recursos. No entanto, não se pode negligenciar sua importância, uma vez que é essencial para garantir detecções de alta qualidade. Além disso, a anotação também permite uma maior precisão e controle sobre os dados, visto que ela garante que eles sejam estruturados de forma adequada e sigam critérios de qualidade e demarcação como os que serão explicitados na Subseção **3.1.2**.

Destarte, esses passos de coleta, filtragem, anotação e estruturação do conjunto de dados são devidamente detalhados nas subseções seguintes.

##### 3.1.1 Coleta de dados

A coleta de dados é uma etapa importante não só para aplicações das áreas de Aprendizado de Máquina e Visão Computacional, mas também para Ciência de Dados de forma geral. É através dela que se obtém informações valiosas que podem ser transformadas em conhecimento útil para alimentar modelos preditivos e algoritmos. Atualmente, existem diversas abordagens de coleta de dados, incluindo o uso de conjunto de dados de código aberto, *download* da *internet* (*web scraping*), geração de dados artificiais e captura própria de dados (ROH; HEO; WHANG, 2019). Cada uma dessas abordagens tem suas vantagens, desvantagens e casos de uso específicos, e podem ou não ser usadas em conjunto. Por isso

é importante avaliá-las cuidadosamente e escolher as mais adequadas às necessidades de cada aplicação.

Segundo L'heureux et al. (2017), a disponibilidade de dados estruturados é um dos maiores desafios enfrentados na construção de sistemas de Aprendizado de Máquina. Nesse sentido, fatores como a qualidade dos dados coletados, suas anotações e estruturação também refletem diretamente no desempenho desses sistemas. Por conseguinte, encontrar conjunto de dados que cumpram esses requisitos e que já estejam estruturados para o problema que se deseja abordar se torna uma tarefa ainda mais desafiadora. Isto é especialmente verdadeiro em áreas menos estudadas ou em problemas onde a quantidade de dados é limitada.

Regularmente, sistemas de detecção e classificação utilizam três conjuntos de dados, sendo eles: treinamento, validação e teste. O primeiro conjunto serve para ensinar o modelo a extrair características relevantes dos dados e aprender a relação entre as entradas e as saídas esperadas de um determinado dado. Enquanto o segundo conjunto serve para avaliar o desempenho e reajustar os hiperparâmetros da rede a partir de seus erros de generalização. Após o treinamento, o conjunto de teste é usado para avaliar a performance final dos modelos em dados que eles nunca viram antes (dados esses que não fazem parte dos conjuntos de treinamento e validação) (GOODFELLOW; BENGIO; COURVILLE, 2016). Isso é importante para determinar se o modelo é capaz de generalizar em novos dados e comparar a performance de diferentes modelos (BURKOV, 2019). Por fim, é importante salientar que para algumas redes, como as escolhidas neste trabalho, o conjunto de validação é opcional, visto que ele não retroalimenta a rede e serve apenas para gerar gráficos de desempenho dos modelos ao longo do treinamento.

Neste trabalho, a criação do conjunto de treinamento adotou o método de *web scraping*, enquanto que o conjunto de teste beneficiou-se de imagens capturadas em fazendas no contexto do projeto *Happy Cow ID*, que fez parte do projeto/ação gerencial “Residência Zootécnica Digital”, uma iniciativa da Empresa Brasileira de Pesquisa Agropecuária (Embrapa – Gado de Leite) em parceria com a Universidade Federal de Juiz de Fora (UFJF) e outras Instituições Federais de Educação Superior (IFES). A criação desses conjuntos visou: I – Priorizar a quantidade e variedade dos dados disponíveis para as etapas de treinamento e testes dos modelos; II – Criar um conjunto de teste que representasse de forma fidedigna condições do dia-a-dia para uma possível aplicação de detecção de bovinos em tempo real; III – Garantir que o conjunto de teste tivesse mais de uma imagem de um mesmo animal em diferentes ângulos; IV – Assegurar que não houvessem imagens duplicadas, principalmente no conjunto de teste.

Para a etapa de *web scraping*, foram realizadas buscas de palavras-chave em 3 idiomas, como disposto na Tabela 2. As fontes de dados escolhidas foram *sites* como

Pexels<sup>1</sup>, Creative Commons<sup>2</sup>, Pixabay<sup>3</sup>, Unsplash<sup>4</sup>, entre outros, uma vez que eles são livres de direitos autorais, possuem imagens em alta resolução e sem marcas d'água. Além disso, a extensão *Fatkun Batch Download Image*<sup>5</sup> do navegador *web Google Chrome*<sup>6</sup> auxiliou no *download* das imagens desses *sites*. Algumas vantagens dessa ferramenta são: a filtragem da resolução mínima e máxima, pré-visualização, seleção, ordenação e a possibilidade de baixar imagens de uma ou mais abas do navegador simultaneamente.

Tabela 2 – Palavras-chave utilizadas na busca e coleta de dados para criação do conjunto de dados.

Português	Espanhol	Inglês
vaca	<i>vaca</i>	<i>cow</i>
bezerro, novilho	<i>becerro</i>	<i>calf</i>
boi, touro	<i>buey, toro</i>	<i>bull, taurus</i>
bovino	<i>bovino</i>	<i>bovine</i>
gado	<i>vacuno, ganado</i>	<i>cattle, livestock</i>

Fonte: Elaborada pelo autor (2023).

De modo a compor um conjunto de treinamento mais generalista e abundante, a coleta de dados focou em reunir imagens de bovinos que contivessem sua face ou alguma de suas outras regiões de interesse, independentemente da raça, categoria, sexo, tipo, pose ou pelagem do animal. De forma semelhante, imagens em diferentes razões de aspecto e que possuíam múltiplos indivíduos também foram coletadas e mantidas no conjunto de dados.

Após a coleta, a filtragem dos dados obtidos se torna essencial, uma vez que ela é responsável por eliminar informações irrelevantes, duplicadas, inconsistentes e confidenciais. Isso torna os dados mais confiáveis, coerentes e fáceis de serem anotados e estruturados nas etapas posteriores. Nesse âmbito, a filtragem se baseou principalmente na remoção de imagens que tivessem a proporção da face menor do que aproximadamente 10% da largura e altura da imagem, bem como imagens que estivessem muito borradas, distorcidas, com problemas críticos de iluminação (onde não era possível identificar os contornos que delimitam a face do animal) ou que não possuíam animais da família dos bovinos, entre outros fatores. A Figura 11 apresenta exemplos de imagens descartadas.

<sup>1</sup> <https://www.pexels.com>

<sup>2</sup> <https://creativecommons.org/>

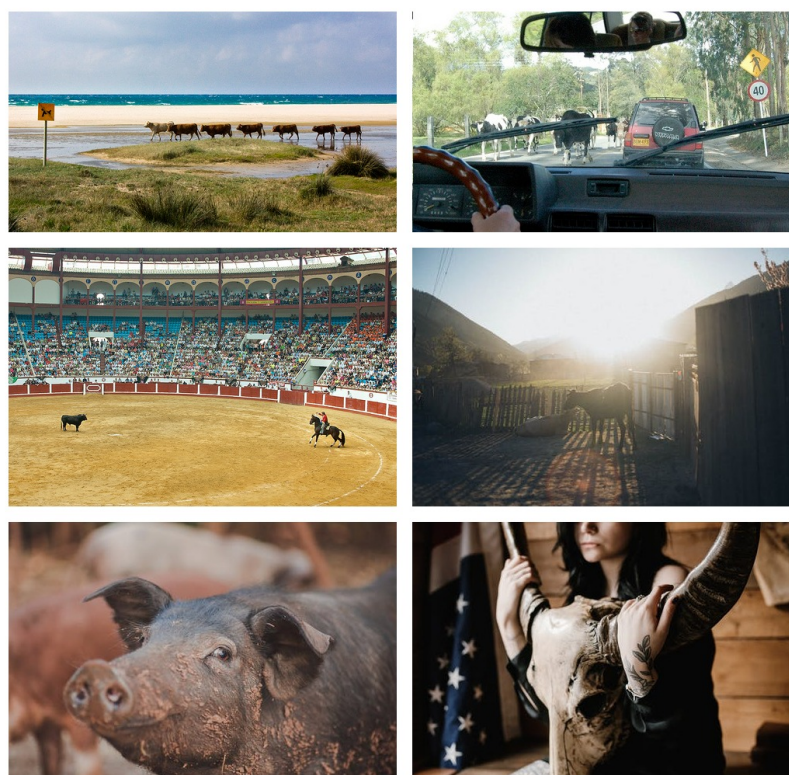
<sup>3</sup> <https://pixabay.com/>

<sup>4</sup> <https://unsplash.com/>

<sup>5</sup> <https://chrome.google.com/webstore/detail/fatkun-batch-download-ima/nnjjahlikiabnchpehcpcckdeckfnohf?hl=pt-br>

<sup>6</sup> <https://chrome.google.com/>

Figura 11 – Exemplos de imagens descartadas durante a etapa de filtragem do conjunto de dados.



Fonte: Elaborada pelo autor (2023).

Destarte, todo esse processo criterioso de curadoria do conjunto de imagens é uma etapa crucial e de grande importância neste trabalho. Através dele, assegura-se a qualidade e relevância dos dados coletados, bem como a estruturação adequada para alimentar os modelos preditivos utilizados. Desse modo, a presença de diferentes raças, categorias, poses e características dos bovinos nos conjuntos de dados permite que os modelos treinados sejam capazes de extrair características relevantes para o problema em questão. Além disso, a criação do conjunto de teste com imagens que representam o cotidiano dos animais em uma ampla variedade de ângulos, poses, iluminação, plano de fundo e nitidez proporciona uma avaliação mais clara do desempenho dos modelos e comparações entre diferentes abordagens. Com isso, o *dataset* construído não apenas oferece suporte à solução do problema em questão, mas também possui um potencial significativo de utilização para outras aplicações, como a geração de imagens de bovinos para treinamento de algoritmos de inteligência artificial em áreas como reconhecimento facial de animais, criação de sistemas de monitoramento de rebanhos e identificação automática de características físicas específicas de bovinos.

Em suma, o conjunto de dados construído está detalhado na Tabela 3. Nela, são mostradas informações gerais sobre as imagens e sobre os indivíduos presentes nos conjuntos de dados de treinamento e teste. Uma vez que as imagens de treinamento foram

coletadas da *internet*, sem qualquer tipo de rotulagem prévia, não foi possível determinar a quantidade de indivíduos e raças nesse conjunto. Além disso, pode-se destacar que as imagens do conjunto de treinamento possuem uma resolução média menor que a do conjunto de teste e que distribuição de imagens por conjunto é de 86,51% para treinamento e 13,49% para teste.

Tabela 3 – Informações detalhadas sobre os conjuntos de dados de treinamento e teste.

Atributo	Treinamento	Teste (Validação)
Qtd. de imagens	6182	964
Resolução mínima	136 × 235	2974 × 2918
Resolução máxima	9799 × 7350	4624 × 3468
Resolução média	2544 × 2157	4317 × 3243
Tamanho mínimo (MB)	0,009	1,82
Tamanho máximo (MB)	46,48	25,72
Tamanho médio (MB)	3,81	13,33
Tamanho total (GB)	23,0	12,5
Qtd. de indivíduos	-	165
Raças	-	Girolando e Holandês
Colorido	Sim	Sim
Formato	JPG	JPG

Fonte: Elaborada pelo autor (2023).

A respeito das imagens dos conjuntos de dados, em ambos os conjuntos foi observada uma grande variação de proximidade, inclinação, plano de fundo, iluminação e nitidez em boa parte de suas imagens. Existem também situações de oclusão e/ou múltiplas faces, categorias, poses, raças, entre outros. Em alguns casos, houve também a presença de múltiplos objetos de interesse que também estavam presentes mesmo quando a face do animal não havia sido completamente identificada na imagem. Outro ponto a se destacar é que algumas imagens do conjunto de treinamento possuem bovinos em ambiente urbano enquanto as imagens do conjunto de teste são apenas em localidades como currais, bezerreiras, pastos ou confinamento. As Figuras 12 e 13 exemplificam, respectivamente, algumas imagens coletadas para o conjunto de treinamento e teste do conjunto de dados.

### 3.1.2 Anotação

O minucioso processo de anotação do conjunto de dados foi indispensável para o treinamento dos modelos de detecção. Nesta etapa, cada objeto que se desejava detectar foi localizado na imagem e recebeu um rótulo correspondente. Esses rótulos, no caso deste trabalho, foram: a face do bovino, suas respectivas orelhas, olhos e focinho. Do ponto de



Figura 12 – Exemplos de imagens do conjunto de treinamento.



Fonte: Elaborada pelo autor (2023).

Figura 13 – Exemplos de imagens do conjunto de teste capturadas em fazendas no contexto do projeto *Happy Cow ID*.



Fonte: Elaborada pelo autor (2023).

vista da aplicação, a anotação dessas regiões é essencial, visto que elas permitem o uso dos modelos em diversos sistemas, uma vez que ele será capaz de fornecer informações importantes para identificação e classificação do animal.

Dado esse contexto, evidencia-se a necessidade da delimitação de *bounding boxes* precisos e consistentes, mesmo quando há a presença de múltiplas pessoas realizando as anotações. Nessa perspectiva, também deve-se garantir que as caixas de delimitação estejam bem posicionadas sobre o alvo e evitar a inclusão de plano de fundo e objetos desinteressantes. Também é importante ressaltar que a omissão na anotação de objetos de uma determinada classe pode resultar em confusão nas detecções por parte dos modelos e, conseqüentemente, diminuir sua precisão e confiabilidade (MA; USHIKU; SAGARA, 2022). Em vista dessas preocupações, tornou-se indispensável a elaboração de um protocolo de controle claro que pudesse garantir a qualidade e a integridade das anotações. As regras a seguir compõem o protocolo elaborado:

1. Anotar todos os bovinos presentes na imagem.
2. Usar os rótulos: *cow* (face da vaca), *left ear* (orelha esquerda), *right ear* (orelha direita), *left eye* (olho esquerdo), *right eye* (olho direito) e *muzzle* (focinho).
3. Respeitar a perspectiva do animal ao anotar as classes “orelha” e “olhos” (esquerda ou direita).
4. Posicionar *bounding boxes* de maneira centralizada sobre os objetos de interesse.
5. Ajustar o rótulo para ocupar a menor área possível ao redor do objeto, minimizando a presença de plano de fundo.
6. Não descartar objetos de interesse mesmo em casos de oclusão parcial da face, rotulando todas as regiões que estiverem visíveis na imagem.
7. Rotular mesmo nos casos em que os olhos estão fechados ou a mandíbula está aberta.
8. Ao realizar a ampliação máxima permitida pela ferramenta em uma imagem, se ainda houver objetos difíceis de reconhecer, eles poderão ser ignorados. Ex: Regiões “embaçadas” ou muito pequenas como os olhos.
9. Evitar rotular os olhos de forma muito justa, sem considerar a região ao redor. Isso pode levar o modelo a confundir os olhos com manchas na pelagem do animal.

Sob esse ponto de vista, a rotulação das imagens foi realizada em colaboração com outro integrante do projeto *Happy Cow ID*. Dessa forma, metade das imagens dos conjuntos de treinamento e teste foi designada para cada anotador. Assim, também foi determinado que, a cada 1000 imagens rotuladas, haveria um processo de revisão das anotações pelos anotadores. A ferramenta de rotulagem escolhida foi o LabelImg (LIN, 2015), por ser de código aberto e multiplataforma, além de possuir uma interface intuitiva e fácil de usar. Outra vantagem importante é a possibilidade de visualização simultânea das imagens e seus respectivos rótulos, facilitando a detecção e correção de possíveis erros de rotulação. A ferramenta também oferece opções para personalização do formato de saída dos rótulos e a exportação das anotações em formatos específicos, como *PASCAL VOC*



ou *YOLO*, tornando o programa flexível e adaptável a diferentes aplicações e necessidades. A Figura 14 mostra exemplos do processo de rotulação através desse *software*.

Figura 14 – Exemplo de anotação de imagens do conjunto de dados com o uso da ferramenta LabelImg.



Fonte: Elaborada pelo autor (2023).

De forma breve, a diferença entre os formatos de anotação se concentra no tipo de arquivo no qual as informações são salvas e na forma como eles estão estruturados. No formato *PASCAL VOC*, cada imagem possui um arquivo *XML* correspondente que contém informações sobre os objetos presentes na imagem, como sua classe, coordenadas do *bounding box* e dificuldade da anotação. Por outro lado, o formato *YOLO* utiliza um arquivo *TXT* por imagem composto de linhas com informações sobre cada objeto, incluindo as classes de forma numérica e as coordenadas centrais, largura e altura do *bounding box* normalizadas entre 0 e 1. Essa normalização é feita a partir da divisão das coordenadas no eixo das abscissas e ordenadas pela largura e altura da imagem, respectivamente. Basicamente, a principal vantagem do formato *YOLO* é que ele é mais simples e fácil de ser implementado, enquanto que o *PASCAL VOC* é mais adequado quando se deseja fornecer mais dados sobre os objetos para a rede de detecção. Por fim, mesmo a ferramenta permitindo a conversão dos dados entre ambos os formatos, pode ser mais prático salvar todas as anotações em um formato e usar um *script* ou biblioteca para convertê-las para outro, se for necessário.

A etapa de rotulação do conjunto de dados demandou aproximadamente cinco meses para ser finalizada. No total, foram anotadas 54.111 instâncias, sendo 45.813 (84,66%) destinadas ao conjunto de treinamento e 8.298 (15,34%) ao conjunto de teste. A anotação dos olhos mostrou-se particularmente desafiadora em virtude de seu tamanho reduzido em relação à resolução das imagens, resultando em uma sub-representação dessa

classe quando comparada às demais. Outrossim, a pose do animal também foi outro fator que afetou a distribuição das anotações, uma vez que, em muitos casos, os animais em posição de perfil não permitiram a visibilidade simultânea dos dois olhos e/ou das duas orelhas.

Em termos quantitativos, observou-se que tanto o comprimento quanto a altura dos rótulos para os olhos tiveram, aproximadamente, até metade do valor das medidas correspondentes para as classes “orelha” e “focinho”. Em contrapartida, a face do animal constituiu o rótulo mais extenso e frequente. A Tabela 4 apresenta essas medidas por classe e conjunto de dados, enquanto a Figura 15 mostra uma representação visual que facilita a compreensão das informações mencionadas anteriormente e mostra também uma semelhança no desbalanceamento das classes dos conjuntos de treinamento e teste.

Tabela 4 – Informações sobre os *bounding boxes* anotados nos conjuntos de dados de treinamento e teste.

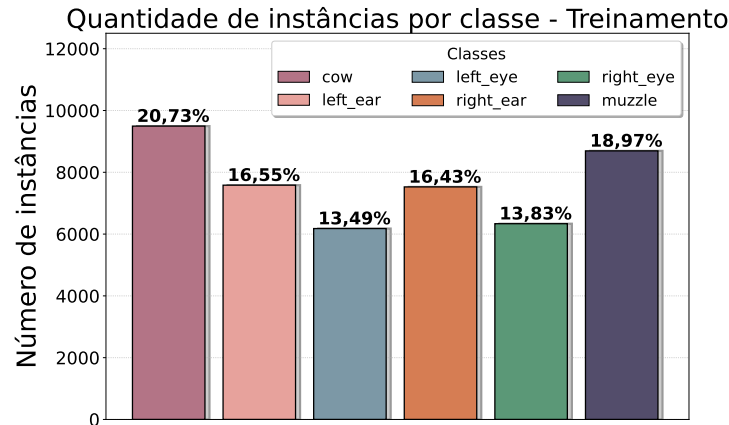
	Classe	Comprimento médio (px)	Altura média (px)	Área média (px <sup>2</sup> )	Número de instâncias
<b>Treinamento</b>	<i>cow</i>	663,39	608,86	1037842,66	9495
	<i>left ear</i>	203,92	182,79	91442,43	7583
	<i>left eye</i>	76,93	88,24	16852,69	6179
	<i>right ear</i>	216,94	190,75	104184,59	7527
	<i>right eye</i>	82,08	93,48	17742,61	6336
	<i>muzzle</i>	215,97	168,69	96583,65	8693
	<b>Teste</b>	<i>cow</i>	652,01	652,69	722617,78
<i>left ear</i>		242,19	205,35	75221,72	1407
<i>left eye</i>		89,33	103,84	12666,33	1010
<i>right ear</i>		256,16	213,22	79643,26	1312
<i>right eye</i>		97,89	113,12	15462,53	980
<i>muzzle</i>		219,01	161,55	54063,64	1656

Fonte: Elaborada pelo autor (2023).

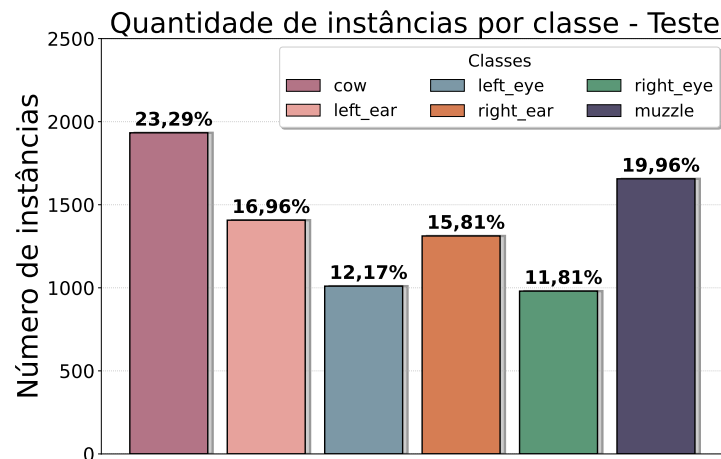
### 3.2 MODELOS DE DETECÇÃO DE OBJETOS

Neste trabalho, foram treinadas e avaliadas duas redes para detecção de objetos: a “*SSD MobileNet V2 FPNLite 640 × 640*” e a “*YOLOv8*”. A primeira rede é uma junção da *SSD* com a rede de classificação *Mobilenet* (CHIU et al., 2020) como seu *backbone*. Enquanto a segunda rede é uma versão recém-lançada da *YOLO*. Ambas possuem como principal vantagem a possibilidade de se utilizar o conjunto de dados

Figura 15 – Distribuição das anotações por classe e por conjunto de dados.



(a) Conjunto de treinamento.



(b) Conjunto de teste.

Fonte: Elaborada pelo autor (2023).

proposto neste trabalho sem a necessidade de grandes ajustes ou retrabalho das etapas de coleta, filtragem ou anotação de dados. Dessa forma, é importante ressaltar também que os treinamentos dessas redes foram realizados utilizando o conjunto de dados descrito na Seção 3.1, sem nenhuma modificação, a fim de possibilitar comparações precisas entre ambas posteriormente.

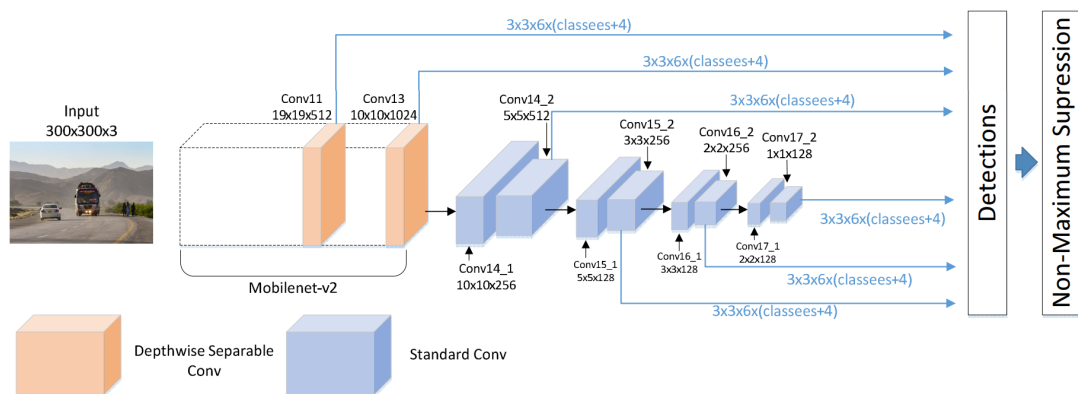
### 3.2.1 *SSD MobileNet V2 FPNLite 640 × 640*

A primeira rede de detecção de objetos utilizada neste trabalho foi a *SSD MobileNet V2 FPNLite 640 × 640*, a qual, por simplicidade, será referida apenas como “SSD Mobilenet” a partir de agora. Essa arquitetura é composta por diversos componentes, como a *MobileNet*, que é responsável por extrair características da imagem, e o *FPNLite (Feature Pyramid Network Lite)*, que gera uma pirâmide de informações com essas características permitindo a detecção de objetos em diferentes escalas espaciais. Além disso, também pode-se destacar

que as imagens de entrada da rede possuem a resolução base de  $640 \times 640$  *pixels* e que podem ser aplicadas técnicas de *data augmentation* ao longo do treinamento.

A *Mobilenet* (HOWARD et al., 2017) é uma rede de classificação popularmente conhecida por ser leve e eficiente em termos de computação, projetada especificamente para ser embarcada em dispositivos móveis. Uma das técnicas utilizadas pela *Mobilenet* para alcançar essa eficiência é a chamada “*factorized convolutions*”, que consiste na decomposição da convolução padrão em duas operações mais simples: a convolução em profundidade (*depthwise convolution*) e a convolução ponto a ponto (*pointwise convolution*). Essa técnica tem como objetivo principal reduzir drasticamente o número de parâmetros da rede, permitindo assim que a *Mobilenet* realize inferências de forma mais rápida e eficiente. A *SSD* utilizando a arquitetura da *MobileNet* como *backbone* é capaz de realizar detecções com boa precisão e, ao mesmo tempo, apresentar um baixo consumo de recursos computacionais. A Figura 16 ilustra a arquitetura de uma rede semelhante a usada no presente trabalho, mas com resolução base menor ( $300 \times 300$  *pixels*) e sem o *FPN Lite*.

Figura 16 – Arquitetura da *SSD MobileNet V2 300 × 300*.



Fonte: Adaptada de Chiu et al. (2020).

Conforme mencionado, o presente trabalho empregou a arquitetura da rede *SSD MobileNet* para gerar modelos capazes de detectar as regiões de interesse em imagens de bovinos. Utilizou-se uma versão dessa rede pré-treinada com o conjunto de dados *COCO* e disponibilizada pelo *TensorFlow Detection 2 Model Zoo*<sup>7</sup>. Essa rede é amplamente utilizada em diversos trabalhos acadêmicos e projetos de desenvolvimento devido à sua eficiência computacional e capacidade de detectar objetos de qualquer tipo, o que a torna uma escolha popular para aplicações que exigem detecção em tempo real. Lamentavelmente, a documentação do *TensorFlow Object Detection API*<sup>8</sup> não fornece informações mais detalhadas sobre essa arquitetura, bem como sobre o processo de pré-treinamento realizado.

Antes de prosseguir para a etapa de treinamento da rede, a base de dados teve de ser reestruturada para um formato binário chamado *TFRecord*, que é utilizado pelo

<sup>7</sup> [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf2\\_detection\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md)

<sup>8</sup> <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/index.html>

*TensorFlow* para o treinamento de modelos de detecção de objetos. Essencialmente, esse formato é responsável por armazenar os dados de forma compacta e otimizada, permitindo mais eficiência em seu acesso e processamento. Ademais, a adoção dessa estrutura também apresenta vantagens tanto em relação à facilidade de uso em ferramentas de computação em nuvem, como é o caso do *Google Colab*<sup>9</sup>, quanto pela sua versatilidade para integração do conjunto de dados com outras ferramentas, redes de detecção e ambientes de treinamento do próprio *TensorFlow*.

Nesse contexto, a rede foi submetida ao treinamento por meio do processo de *fine-tuning*, que consiste em mesclar o conhecimento prévio (*i.e.*, “pesos”) da rede com novos dados, visando melhorar o seu desempenho para a tarefa abordada. Esse processo aprimora o reconhecimento de padrões e a aprendizagem de características de similaridade dos objetos de uma mesma classe, ao mesmo tempo que ensina o modelo a ignorar informações irrelevantes e ruídos das imagens. Cabe ressaltar que, no escopo deste trabalho, o *fine-tuning* foi realizado utilizando-se o conjunto de dados construído na Seção 3.1.

Ao final do treinamento, um modelo de detecção é gerado e, a partir dele, podem ser realizadas inferências tanto no conjunto de teste para avaliar o seu desempenho, quanto em imagens que originalmente não pertençam à base de dados. O modelo pode ser exportado para os formatos do *TensorFlow* e *TensorFlowLite*, permitindo sua utilização em várias plataformas. Cada predição em uma imagem gera um vetor de detecções que inclui informações sobre a localização dos objetos e suas respectivas classes. Para selecionar as detecções mais relevantes, é aplicado a supressão não-máxima e um limiar de confiança mínimo. O resultado final é um vetor contendo as melhores detecções da rede, juntamente com as coordenadas dos seus *bounding boxes* correspondentes.

### 3.2.2 YOLOv8

A *YOLOv8* (JOCHER; CHAURASIA; QIU, 2023) é a versão mais recente da rede de detecção de objetos *YOLO*, lançada de forma gratuita em janeiro de 2023 pela empresa *Ultralytics*. É uma rede neural de última geração que se baseia nos pontos positivos das versões anteriores da *YOLO* enquanto introduz novas funcionalidades e melhorias para aumentar ainda mais o desempenho e a flexibilidade de seus modelos. Um dos principais destaques para essa nova versão é a possibilidade de usá-la para quatro tarefas: o treinamento de modelos de detecção de objetos, a classificação de imagens, a segmentação de instâncias e a estimativa de poses. Assim como a *SSD*, a rede é projetada para ser rápida, precisa e fácil de usar, tornando-a uma excelente escolha para uma ampla gama de aplicações.

Ela foi disponibilizada no Github<sup>10</sup> e possui documentação detalhada na plataforma

<sup>9</sup> <https://colab.research.google.com/>

<sup>10</sup> <https://github.com/ultralytics/ultralytics>

da *Ultralytics*<sup>11</sup>. Os autores são os mesmos da *YOLOv5* e o ambiente disponibilizado pela plataforma permite construir *pipelines* de treinamento, validação e teste com ambas as versões da *YOLO*. Além disso, a plataforma fornece 5 modelos de detecção de objetos pré-treinados no conjunto de dados *COCO* para os quais pode ser realizado um *fine-tuning* com conjuntos de dados personalizados. A principal diferença entre esses modelos é o seu número de parâmetros, os quais variam de 3,2 a 68,2 milhões, que, por consequência, podem ter o potencial de refinar a qualidade das detecções ao custo de mais recursos computacionais para treinamento e inferência dos modelos. Ademais, outras tarefas como classificação e segmentação também possuem seus próprios modelos pré-treinados.

Para instalar a *YOLOv8*, basta ter algum ambiente Python<sup>12</sup> na máquina e executar o comando “`pip install ultralytics`” para instalar todas as dependências do *framework*. Antes do treinamento, deve-se adicionar os conjuntos de dados em uma pasta intitulada “*data*”, sendo que para cada conjunto (treinamento e teste) também devem ser criadas subpastas com suas imagens (*images*) e rótulos (*labels*). Lembrando que esses rótulos devem estar no formato de arquivo *TXT* e no formato de anotação *YOLO*, o qual foi descrito na Seção 3.1. Além disso, deve ser criado um arquivo *YAML* que contenha o mapeamento das pastas e quais são as classes rotuladas nesse conjunto de dados.

Após a estruturação dos dados, basta escolher um dos modelos pré-treinados e realizar estudos de ablação a partir dos experimentos para encontrar as configurações que melhor solucionem determinado problema de detecção. Neste trabalho, o modelo usado é o “*YOLOv8n*”, com resolução de entrada de  $640 \times 640$  *pixels* e 3,2 milhões de parâmetros. Por fim, a rede também permite exportar os modelos em diversos formatos como *Tensorflow Lite*, *PyTorch*, *Keras*, entre outros. Neste trabalho, foi mantido o formato de exportação padrão (*PyTorch*).

---

<sup>11</sup> <https://docs.ultralytics.com/>

<sup>12</sup> <https://www.python.org/>

## 4 EXPERIMENTOS E RESULTADOS

Com o objetivo de avaliar a efetividade dos modelos propostos, foram realizados experimentos utilizando o conjunto de dados e as redes de detecção previamente descritas. Nesta seção, serão apresentados os resultados obtidos e serão discutidos os principais percepções e observações encontrados durante a avaliação das detecções realizadas pelos modelos. Além disso, será realizada uma análise comparativa entre os modelos de detecção treinados, a fim de avaliar o desempenho de cada um deles e discutir suas diferenças. Serão apresentadas discussões sobre os resultados obtidos, com uma análise crítica sobre as limitações e pontos fortes dos modelos propostos, além de uma análise qualitativa das detecções.

### 4.1 CONFIGURAÇÃO DOS EXPERIMENTOS

Para a realização dos experimentos propostos neste trabalho, foram utilizadas duas máquinas distintas, dispostas na Tabela 5. A primeira máquina, pertencente ao autor deste trabalho, foi utilizada para o treinamento dos modelos de detecção com a rede *SSD Mobilenet*, devido à sua disponibilidade e facilidade de acesso. Enquanto isso, a segunda máquina, viabilizada pelo Grupo de Computação Gráfica, Imagem e Visão da UFJF (GCG-UFJF), foi utilizada para o treinamento dos modelos com a rede *YOLOv8*.

Tabela 5 – Configurações das máquinas.

Dispositivo	Máquina 1	Máquina 2
Processador	<i>Intel i5-10300H @ 2.50 GHz</i>	<i>Intel Core i5-10400 2.90GHz</i>
Memória RAM	<i>24GB Dual-Channel</i>	<i>16GB Dual-Channel</i>
Placa de vídeo	<i>GeForce GTX 1650 4GB</i>	<i>GeForce RTX 3060 12GB</i>
Disco Rígido	<i>KINGSPEC NVME SSD 256GB</i>	<i>SAMSUNG HD322HJ 320GB</i>
Sistema	<i>Windows 10 Pro 64-bit</i>	<i>Ubuntu 22.04.2 LTS</i>

Fonte: Elaborada pelo autor (2023).

É importante mencionar que cada máquina foi responsável pelo treinamento e condução de experimentos de forma independente para cada rede. Em virtude da instalação e utilização do CUDA (*Compute Unified Device Architecture* – NVIDIA, Vingelmann e Fitzek (2020)), uma plataforma de computação paralela que acelera o treinamento de redes neurais, ocorreram problemas de compatibilidade para configuração do ambiente de treinamento da rede *SSD Mobilenet*. Isso se deve ao fato de diferentes versões de sistemas operacionais, *drivers*, CUDA e *TensorFlow* não serem compatíveis entre si, o que tem levado a problemas frequentes que vêm sendo relatados em alguns fóruns da *internet*. Tal inconveniente acabou tornando inviável o treinamento dessa rede na Máquina 2. Como resultado, a Máquina 1, mesmo tendo um desempenho computacional relativamente

inferior à Máquina 2, acabou sendo escolhida para treinar essa rede, uma vez que ela não apresentou os problemas de compatibilidade mencionados anteriormente. Por fim, a Máquina 2 foi usada apenas para o treinamento da *YOLOv8*.

Por se tratar de um trabalho extenso, foi necessário utilizar diversos *frameworks* e bibliotecas ao longo das etapas do projeto. De modo geral, é preciso destacar a importância das bibliotecas *Tensorflow* 2.10 (ABADI et al., 2015), *PyTorch* 2.0 (PASZKE et al., 2019) e *Ultralytics* 8.0.72, usadas nos processos de treinamento das redes de detecção de objetos. Além disso, pode-se citar algumas bibliotecas de igual importância como *pillow*, *matplotlib*, *pandas*, *numpy*, *opencv*, dentre outras. Por fim, a linguagem de programação utilizada foi o *Python* 3.10.

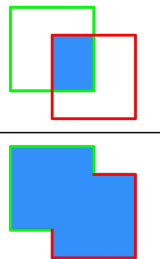
Dessa forma, a utilização das duas máquinas permitiu a execução dos experimentos propostos de maneira eficiente e adequada, garantindo a precisão dos resultados obtidos e a verificação da eficácia das redes de detecção propostas.

## 4.2 CRITÉRIOS DE AVALIAÇÃO

A avaliação do desempenho de modelos de detecção de objetos requer a compreensão dos principais critérios que determinam a eficácia de suas inferências (FAWCETT, 2006). Nesse sentido, serão consideradas os seguintes critérios de avaliação neste trabalho: Interseção sobre União (*Intersection over Union* – IoU), precisão (*precision*), revocação (*recall*), *F-score* (F1), precisão média (*Average Precision* – AP), média das precisões médias (*mean Average Precision* – mAP) e confiança.

A Interseção sobre União é um critério que avalia o quão precisamente um objeto foi detectado em uma imagem. Ela compara a sobreposição entre as caixas delimitadoras detectadas pelo modelo e as caixas delimitadoras anotadas (*ground truths*). Para calcular o IoU, a área de sobreposição das duas caixas é dividida pela área total coberta por ambas, conforme ilustrado na Figura 17. Seu valor varia de 0 a 1, onde o valor máximo representa a situação ideal, na qual as caixas são idênticas, enquanto o valor mínimo indica que as caixas não têm sobreposição alguma e são completamente diferentes.

Figura 17 – Demonstração do cálculo do IoU a partir da divisão da área da intersecção pela área da união entre as caixas de delimitação predita e real.

$$IoU = \frac{\text{área da intersecção}}{\text{área da união}} = \frac{\text{área da intersecção}}{\text{área da união}}$$


Fonte: Adaptada de Padilla, Netto e Silva (2020).



De maneira usual, um valor de IoU acima de um limiar predefinido (geralmente 0,5) indica uma detecção correta, enquanto um resultado abaixo deste valor é visto como uma detecção incorreta. Com base nesse critério, cada caixa delimitadora gerada pelo modelo pode ser classificada em uma das quatro categorias:

- Verdadeiro Positivos (*True Positives* – TP): quando o objeto de interesse existe na imagem e é corretamente detectado. Por exemplo, em uma imagem de um gato e um cachorro, se o gato for detectado e classificado como gato e o cachorro for detectado e classificado como cachorro, então tem-se dois verdadeiros positivos. Normalmente, uma detecção é considerada um verdadeiro positivo quando seu valor de IoU é maior que o limiar definido ( $\text{IoU} > \text{limiar}$ ).
- Falso Positivos (*False Positives* – FP): quando o objeto de interesse **não existe** na imagem mas é erroneamente detectado. Por exemplo, em uma imagem de um gato, se o gato for erroneamente detectado e classificado como cachorro, então tem-se um falso positivo. Normalmente, uma detecção é considerada um falso positivo quando seu valor de IoU é menor ou igual ao limiar definido ( $\text{IoU} \leq \text{limiar}$ ).
- Falso Negativos (*False Negatives* – FN): quando o objeto de interesse existe na imagem mas não é detectado. Por exemplo, em uma imagem de um gato, se nenhum gato for detectado, então tem-se um falso negativo.
- Verdadeiro Negativos (*True Negatives* – TN): quando o objeto de interesse não existe na imagem e não é detectado. Por exemplo, em uma imagem de um cachorro, se nenhum gato for detectado, então tem-se um verdadeiro negativo.

A precisão é o critério responsável por medir a taxa de acerto dos objetos detectados corretamente em relação ao total de objetos detectados pelo modelo. Em outras palavras, ela avalia a capacidade dos modelos de identificarem corretamente os verdadeiros positivos e **minimizar falsos positivos**. Em geral, a precisão é mais relevante quando o objetivo é evitar que instâncias negativas sejam erroneamente classificadas como positivas. Sua fórmula é dada por:

$$\text{Precision} = \frac{TP}{TP + FP},$$

onde TP representa o número de verdadeiros positivos e FP o número de falsos positivos. Uma alta precisão indica que poucos objetos são erroneamente detectados pelo modelo.

No entanto, a precisão sozinha pode não ser suficiente para avaliar adequadamente o desempenho das detecções. Isso ocorre porque um modelo pode ter alta precisão simplesmente ignorando objetos verdadeiros. Nesse contexto, a revocação mede a proporção de detecções corretas em relação ao total de objetos que deveriam ter sido detectados. Assim, esse critério demonstra a capacidade dos modelos de identificarem corretamente os verdadeiros positivos e de **minimizar os falsos negativos**. Dessa forma, a revocação é crucial em situações em que é importante detectar o máximo possível de objetos verdadeiros.

Sua equação é dada por:

$$Recall = \frac{TP}{TP + FN},$$

onde TP representa o número de verdadeiros positivos e FN o número de falsos negativos. Um alto revocação indica que o modelo está detectando a maioria dos objetos de interesse presentes nas imagens.

Portanto, para uma avaliação mais contundente do desempenho dos modelos, é importante considerar tanto a precisão quanto a revocação. Isso é especialmente importante em situações em que é necessário equilibrar os dois valores, como em problemas em que se deseja minimizar os falsos positivos, mas também detectar o maior número possível de objetos presentes na imagem. Para alcançar esse equilíbrio, pode-se utilizar o critério “*F-score*”, ou *F1*, que é a média harmônica da precisão e revocação, definida por:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}.$$

O valor máximo de F1 é 1, indicando uma detecção perfeita, em que todas as detecções são corretas e não há falsos positivos ou falsos negativos. Isso significa que o algoritmo de detecção obteve uma alta precisão e uma alta revocação simultaneamente. Já o valor mínimo é 0, indicando uma detecção muito ruim, em que a precisão e/ou revocação são nulos. Isso ocorre quando todas as detecções estão incorretas ou quando nenhum objeto da imagem foi detectado.

A precisão média (AP), por sua vez, é um critério de avaliação normalmente calculado a partir da integral definida da área sob a curva precisão  $\times$  revocação. Isso significa que ela fornece uma visão mais completa da precisão dos modelos em diferentes níveis de revocação, ajudando a avaliar como eles se comportam em diferentes pontos da curva. Por exemplo, se um modelo tem alta precisão em um determinado ponto, mas baixa revocação, o cálculo do AP levará em conta essa compensação. Sua integral é dada por:

$$AP = \int_0^1 p(r) dr,$$

onde  $p(r)$  é a precisão sob o valor de revocação  $r$ . Quanto maior o valor encontrado, melhor o modelo de detecção está performando, o que indica uma capacidade consistente de equilibrar a precisão e a revocação em diferentes pontos da curva.

O mAP é um critério baseado na média dos valores de AP encontrados para todas as classes de objetos no conjunto de dados. Essa critério é amplamente utilizado em problemas de detecção de objetos, pois oferece uma avaliação mais completa do desempenho dos modelos para a detecção de todas as classes em que ele foi treinado. Seu valor varia entre 0 e 1 e pode ser calculado a partir da equação:

$$mAP = \frac{1}{n} \sum_{k=1}^n AP_k,$$

onde  $n$  é o número total de classes presentes no conjunto de dados e  $AP_k$  é a Precisão Média calculada para cada classe  $k$  individualmente.

Neste trabalho, ambas as redes de detecção foram avaliadas utilizando-se os valores de mAP50 e mAP50-95. O mAP50 avalia o desempenho das detecções com um limiar de sobreposição de pelo menos 50% de IoU, enquanto o mAP50-95 é a média dos valores de AP para cada limiar de sobreposição no intervalo de 50% a 95%, com incrementos de 5%. Dessa forma, o mAP50-95 permite uma análise mais abrangente e detalhada do desempenho dos modelos em diferentes níveis de sobreposição. Por esse motivo, é comum na literatura considerar o modelo com o melhor valor de mAP50-95 como o mais robusto e eficaz, uma vez que ele possui uma maior capacidade de encontrar objetos corretamente alinhados e sobrepostos às regiões verdadeiras da imagem durante a detecção.

Por fim, a confiança indica a probabilidade de um objeto detectado pertencer a uma determinada classe. Dessa forma, um valor de confiança alto indica que o objeto provavelmente pertence àquela classe que lhe foi rotulada, enquanto um valor de confiança mais baixo indica que o modelo teve dificuldades em classificar aquele objeto e pode ter se confundido. Nesse caso, a classe do objeto detectado pode estar incorreta ou pode não haver um objeto naquela localização, resultando em um falso positivo. Com base na confiança, é possível identificar a necessidade de ajustes no modelo e direcionar esforços para diminuir suas principais confusões, tornando-o mais confiável e eficaz na detecção de objetos em diversos cenários.

### 4.3 EXPERIMENTOS COM A *SSD MOBILENET V2 FPNLITE* $640 \times 640$

Os experimentos conduzidos com a arquitetura da rede pré-treinada *SSD Mobilenet* para detecção de objetos são descritos na Tabela 6. Eles foram realizados modificando parâmetros de treinamento e técnicas de *data augmentation*. No total, foram conduzidos 15 experimentos, todos com 200 épocas de treinamento e resolução de imagem padrão de 640 *pixels*.

Nos experimentos 1 e 2 houve apenas alterações no tamanho do lote (*batch size*) para o treinamento, sendo que o *data augmentation* utilizado foi apenas o recorte aleatório de imagem (*random crop image*), que faz parte da configuração padrão da rede. Esses valores de *batch size* foram escolhidos com base nos recursos computacionais da máquina utilizada, uma vez que não foi possível executar experimentos para valores de *batch size* maiores que 4.

A partir do experimento 3 ao 11, o *batch size* foi fixado em 2 por ter um desempenho mais consistente e os experimentos passaram a variar individualmente apenas as técnicas de *data augmentation*. Nesses experimentos, foram exploradas diferentes técnicas, como a inversão horizontal aleatória (*random horizontal flip*), escala de imagem aleatória (*random image scale*), conversão aleatória de RGB para cinza (*random rgb to gray*), ajuste de brilho

Tabela 6 – Configurações e resultados dos experimentos com a *SSD Mobilenet*.

Experimento	Batch size	Data augmentations	Precisão	Revocação	mAP50	mAP50-95
1	2	<i>random crop image</i>	0,7978	0,7297	0,4733	0,2492
2	4	<i>random crop image</i>	0,8030	0,7326	0,4730	0,2443
3	2	-	0,7831	0,7028	0,4335	0,2215
4	2	<i>random pixel value scale</i>	0,7753	0,7049	0,4338	0,2185
5	2	<i>random image scale</i>	0,7824	0,7155	0,4356	0,2219
6	2	<i>random rgb to gray</i>	0,7705	0,7221	0,4214	0,2140
7	2	<i>random adjust brightness</i>	0,7719	0,7178	0,4348	0,2206
8	2	<i>random adjust contrast</i>	0,7869	0,7150	0,4268	0,2186
9	2	<i>random adjust hue</i>	0,7790	0,7047	0,4278	0,2194
10	2	<i>random adjust saturation</i>	0,7604	0,7169	0,4321	0,2203
11	2	<i>random distort color</i>	0,7779	0,7086	0,4332	0,2219
12	2	<i>random image scale, random crop image</i>	0,8063	0,7421	0,4753	0,2457
13	2	<i>random distort color, random crop image</i>	0,8078	0,7398	0,4783	0,2495
14	2	<i>random image scale, random distort color, random crop image</i>	0,7957	0,7312	0,4750	0,2453
15	2	<i>random image scale, random distort color</i>	0,7789	0,7191	0,4315	0,2224

Fonte: Elaborada pelo autor (2023).

aleatório (*random adjust brightness*), escala aleatória de valor de pixel (*random pixel value scale*), ajuste aleatório de contraste (*random adjust contrast*), ajuste aleatório de matiz (*random adjust hue*), ajuste aleatório de saturação (*random adjust saturation*), distorção aleatória de cor (*random distort color*). No experimento 3, mais especificamente, nenhum *data augmentation* foi utilizado com o objetivo de se estabelecer uma base de comparação para os experimentos seguintes e verificar quais técnicas poderiam trazer mais benefícios para os critérios avaliados.

Por fim, os experimentos de 12 a 15 visavam a exploração de diversas combinações de técnicas de *data augmentation* de forma a aprimorar o desempenho dos modelos. Para isso, foram selecionadas as técnicas que apresentaram os três melhores resultados até o experimento 11. Essa abordagem foi adotada com o intuito de maximizar as melhorias no modelo, aproveitando as técnicas previamente identificadas como promissoras e explorando suas combinações de forma variada para investigar se há uma sinergia capaz de proporcionar avanços adicionais.

Os resultados são baseados na avaliação do modelo gerado na última época de cada experimento e são apresentados na Tabela 6. Os critérios de avaliação são: precisão, revocação, mAP50 e mAP50-95. As células em verde destacam os melhores resultados para cada critério. Observando essa tabela, é possível identificar algumas tendências:

- Os experimentos que usaram técnicas de *data augmentation* tiveram, de forma geral, resultados superiores ao que não usou (comparando o experimento 3 com os demais).
- Em linhas gerais, o experimento 13 se destacou em todas os critérios avaliados. Ao combinar as técnicas de *random distort color* e *random crop image*, ele obteve os melhores resultados em termos de precisão (0,8078), mAP50 (0,4783) e mAP50-95 (0,2495), e o segundo melhor revocação (0,7421).

Essas tendências sugerem que a combinação de técnicas de *data augmentation* pode proporcionar um melhor desempenho ao treinar a rede *SSD Mobilenet*. No entanto, a escolha das técnicas deve ser feita com cuidado, já que nem todas as combinações resultam em melhorias, como evidenciado pelo experimento 15. Dessa forma, o experimento 13 foi definido como o melhor modelo de detecção utilizando a rede *SSD Mobilenet*. Portanto, esse modelo foi selecionado para a produção de gráficos e para comparações com o modelo de melhor desempenho treinado com a *YOLOv8*, que será abordado na próxima seção.

#### 4.4 EXPERIMENTOS COM A *YOLOv8*

Os experimentos realizados com a *YOLOv8* foram estruturados com o objetivo de avaliar o impacto de diferentes parâmetros de treinamento na eficácia da detecção de objetos, bem como de possibilitar comparações com o melhor modelo treinado com a rede *SSD Mobilenet*. Antes de aprofundar nos experimentos específicos, vale destacar as diversas técnicas de *data augmentation* que podem ser utilizadas pela rede, como descrito na Tabela 7. Adicionalmente, destaca-se que o uso dessas técnicas depende da definição prévia do parâmetro “*augment=true*”.

Tabela 7 – Descrição de algumas das técnicas de *data augmentation* e seus valores padrão na implementação original da *YOLOv8*. Esses valores são empregados apenas na etapa de treinamento da rede. Para poder usá-los, o parâmetro “*augment=true*” deve ser predefinido.

Técnica	Valor padrão	Descrição
<i>hsv_h</i>	0,015	Aumento de intensidade da Matiz (fração)
<i>hsv_s</i>	0,7	Aumento de intensidade da Saturação (fração)
<i>hsv_v</i>	0,4	Aumento de intensidade do Valor (fração)
<i>degrees</i>	0,0	Rotação (+/- grau)
<i>translate</i>	0,1	Translação (+/- fração)
<i>scale</i>	0,5	Escala da imagem (+/- ganho)
<i>fliplr</i>	0,5	Espelhamento vertical (probabilidade)
<i>mosaic</i>	1,0	Composição de imagens com mosaico (probabilidade)

Fonte: Adaptada de Jocher, Chaurasia e Qiu (2023).

Os experimentos foram divididos em três grupos distintos, como mostra a Tabela 8, cada um com suas particularidades. É importante observar que, para uma melhor visuali-

zação, foram incluídos na coluna de hiperparâmetros apenas os *data augmentations* que foram modificados. Em outras palavras, a técnica mostrada em um experimento foi usada em conjunto com as demais técnicas da Tabela 7 utilizando os valores padrão. Além disso, para evitar confusões nas detecções entre o lado esquerdo e o lado direito das classes dos olhos e orelhas, a técnica de espelhamento horizontal (*fliplr*) foi desativada em todos os experimentos, uma vez que os rótulos não são modificados durante o treinamento.

Tabela 8 – Configuração dos experimentos com a *YOLOv8*.

Experimento	Epochs	Batch size	Resolução da imagem	Hiperparâmetros
1	20	16	640	<i>augment=true fliplr=0,0</i>
2	20	8	1080	<i>augment=true fliplr=0,0</i>
3	40	8	1620	<i>augment=true fliplr=0,0</i>
4	40	8	1920	<i>augment=true fliplr=0,0</i>
5	40	8	1920	<i>augment=true hsv_h=0,015 hsv_s=0,7 hsv_v=0,4 degrees=0,15 translate=0,1 scale=0,5 fliplr=0,0 mosaic=0,0</i>
6	40	8	1920	<i>augment=true hsv_h=0,015 hsv_s=0,7 hsv_v=0,4 degrees=0,0 translate=0,0 scale=0,0 fliplr=0,0 mosaic=0,0</i>
7	40	8	1920	<i>augment=true hsv_h=0,0 hsv_s=0,0 hsv_v=0,0 degrees=0,15 translate=0,1 scale=0,5 fliplr=0,0 mosaic=0,0</i>
8	40	8	1920	<i>augment=false</i>
9	40	8	1920	<i>cos_lr=false lr0=0,001 lrf=0,01 augment=true fliplr=0,0</i>
10	40	8	1920	<i>cos_lr=false lr0=0,001 lrf=0,1 augment=true fliplr=0,0</i>
11	40	8	1920	<i>cos_lr=true lr0=0,001 lrf=0,01 augment=true fliplr=0,0 mosaic=0,0</i>
12	40	8	1920	<i>cos_lr=true lr0=0,001 lrf=0,1 augment=true fliplr=0,0 mosaic=0,0</i>
13	40	8	1920	<i>cos_lr=true lr0=0,1 lrf=0,01 augment=true fliplr=0,0 mosaic=0,0</i>
14	40	8	1920	<i>cos_lr=true lr0=0,1 lrf=0,001 augment=true fliplr=0,0 mosaic=0,0</i>

Fonte: Elaborada pelo autor (2023).

Dessa forma, o primeiro grupo (experimentos 1-4) variou a quantidade de épocas, o tamanho do lote e a resolução da imagem. Além disso, o *batch size* foi diminuído quando a resolução da imagem foi aumentada devido a falta de memória gráfica para o treinamento. O segundo grupo (experimentos 5-8) manteve a mesma quantidade de épocas, o tamanho do lote e a resolução da imagem do melhor experimento do primeiro grupo e fez combinações das técnicas de *data augmentation*. Por fim, o terceiro grupo (experimentos 9-14) utilizou as mesmas configurações do melhor experimento até então (experimento 4) e alterou os parâmetros de taxa de aprendizado.

Explorando com mais profundidade o último grupo de experimentos, os parâmetros *lr0* e *lrf* são ambos utilizados durante o treinamento da rede para controlar a programação da taxa de aprendizagem. O parâmetro *lr0* é a taxa de aprendizagem inicial, e *lrf* é a taxa de aprendizagem final na última época do treinamento. Por padrão, tanto *lr0* quanto

$lrf$  têm o mesmo valor de 0,01, sendo assim, não há mudanças na taxa de aprendizagem ao longo do treinamento.

Em relação ao parâmetro  $cos\_lr$ , se ele estiver definido como verdadeiro, então a programação da taxa de aprendizagem seguirá um padrão de *cosine annealing* ao invés de uma programação linear. O *cosine annealing* é um método de programação de taxas de aprendizagem que reduz gradualmente a taxa de aprendizagem durante as épocas de treinamento, seguindo um padrão cosseno, que é mais suave e pode levar a uma melhor convergência e a estabilidade do modelo de aprendizado de máquina. Usualmente, a taxa de aprendizado inicial  $lr0$  para essa função deve ser um valor máximo, o qual será reduzido gradualmente até atingir a taxa de aprendizagem final  $lrf$  que deve ser o valor mínimo.

Os resultados desses experimentos estão detalhados na Tabela 9. Assim como na rede anterior, a última época de cada experimento foi avaliada com base em quatro critérios principais: precisão, revocação, mAP50 e mAP50-95. Nessa tabela, as células destacadas em verde representam os melhores valores obtidos para cada critério.

Tabela 9 – Resultados dos experimentos com a *YOLOv8*.

Experimento	Precisão	Revocação	mAP50	mAP50-95
1	0,8599	0,6281	0,6746	0,3785
2	0,8667	0,5928	0,6405	0,3697
3	0,8590	0,6933	0,7374	0,4228
4	0,8409	0,7159	0,7591	0,4429
5	0,8628	0,7152	0,7699	0,4464
6	0,8687	0,6700	0,7449	0,4202
7	0,8592	0,7022	0,7497	0,4321
8	0,8514	0,6532	0,7280	0,4159
9	0,8609	0,7016	0,7584	0,4478
10	0,8622	0,7086	0,7680	0,4529
11	0,8561	0,6896	0,7484	0,4424
12	0,8550	0,6998	0,7507	0,4409
13	0,8676	0,6661	0,7364	0,4243
14	0,8687	0,6554	0,7182	0,4165

Fonte: Elaborada pelo autor (2023).

A análise das Tabelas 8 e 9 permite destacar alguns pontos em relação aos experimentos e aos critérios de avaliação:

- Em geral, os experimentos que utilizaram *data augmentations* apresentaram melhores resultados, principalmente quando se compara o experimento 8 (sem *data augmentations*) com os experimentos de 1 a 7 (com *data augmentations*).
- A princípio não foi possível identificar um único experimento que tenha alcançado o melhor desempenho em todas os critérios simultaneamente, uma vez que os melhores resultados para cada critério estão distribuídos entre diferentes experimentos.

- Como o experimento 10 se destacou por obter o melhor mAP50-95 da tabela (0,4529) e valores comparáveis para os outros critérios, ele foi escolhido como o melhor experimento dessa rede.

Em linhas gerais, esses resultados enfatizam a importância de investigar diferentes parâmetros e combinações durante o treinamento de modelos de detecção de objetos, visto que outras configurações podem otimizar ainda mais seu desempenho no futuro. Essa abordagem contínua de experimentação e refinamento é fundamental para impulsionar o avanço e o aprimoramento dos modelos, ao mesmo tempo em que permite a identificação de padrões que possam melhorar seus resultados, dados os critérios de avaliação.

## 4.5 DISCUSSÃO

### 4.5.1 Análise Quantitativa

Ao analisar os critérios de avaliação apresentados nas Tabelas 6 e 9 e comparar o melhor experimento de cada rede. Observa-se que a *YOLOv8* alcançou resultados superiores à *SSD Mobilenet*. Em termos de precisão e revocação, a *YOLOv8* apresentou uma diferença pequena de aproximadamente 6% e 3%, respectivamente. Por outro lado, os critérios de mAP50 e mAP50-95 da *YOLOv8* mostraram uma superioridade mais significativa, com incrementos em torno de 29% e 20%, para cada caso. Acredita-se que a principal causa para esse comportamento tenha sido o uso de resoluções de imagem mais altas nessa rede, visto que isso permitiu uma captura de características mais detalhadas dos objetos. Contudo, é importante realizar outras análises mais detalhadas do desempenho dos modelos, o que será feito na sequência do trabalho.

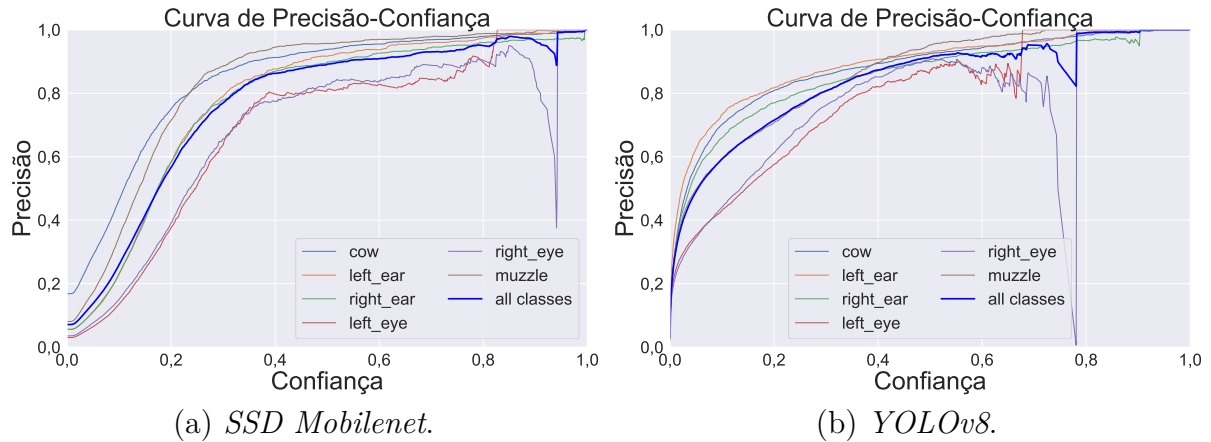
Após a análise dos dados tabelados, é importante prosseguir para a investigação e comparação dos gráficos de Precisão-Confiança, Revocação-Confiança, Precisão-Revocação e F1-Confiança a partir dos melhores modelos gerados para cada rede. Essa análise permite comparar de forma mais clara o desempenho entre as duas. Além disso, ela possibilita uma melhor compreensão de como esses critérios se comportam sob diferentes valores de confiança, fornecendo esclarecimentos adicionais sobre as vantagens e desvantagens de cada rede.

Os gráficos de Precisão-Confiança (Figura 18) retratam a proporção de objetos corretamente detectados em diferentes níveis de confiança, demonstrando a relação entre a precisão dos modelos e esses valores. Usualmente, é encontrada uma curva ascendente nestes gráficos, sugerindo que a precisão aumenta à medida que a confiança dos modelos aumenta. Isso indica que a maioria das detecções com valor alto de confiança tendem a estar corretas.

Para o gráfico da *SSD Mobilenet* (Figura 18a), observa-se que todas as curvas das classes são ascendentes, com um aumento acentuado de precisão em torno de 0,2



Figura 18 – Gráficos de Precisão por confiança dos melhores modelos para as redes *SSD Mobilenet* e *YOLOv8*.



Fonte: Elaborada pelo autor (2023).

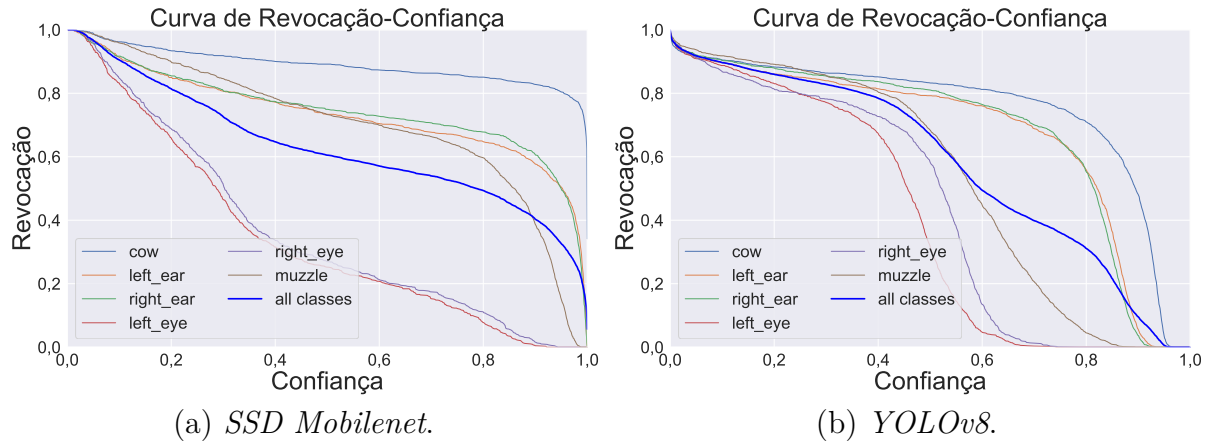
de confiança. No entanto, a classe “*right eye*” apresenta uma precisão relativamente menor quando a confiança está entre 0,8 e 1,0. Por outro lado, o gráfico da *YOLOv8* (Figura 18b) revela um rápido aumento da precisão para todas as classes mesmo em valores de confiança inferiores a 0,2. Após esse ponto, a taxa de crescimento diminuiu. Similarmente à *SSD Mobilenet*, a classe “*right eye*” também exibe uma menor precisão próximo a 0,8 de confiança. Esse comportamento em ambos os gráficos pode ser explicado pelo baixo número de detecções para valores de confiança nesse intervalo. No entanto, para uma compreensão mais aprofundada e abrangente dessas quedas, estudos adicionais são necessários no futuro.

Comparando os dois gráficos, o melhor modelo da *YOLOv8* demonstra ser capaz de identificar corretamente um maior número de objetos em valores de confiança mais baixos. Embora seu comportamento geral ao longo da curva seja bem semelhante ao da *SSD Mobilenet*.

Em seguida, tem-se os gráficos de Revocação-Confiança (Figura 19), que são similares aos gráficos anteriores, contudo, eles focam na revocação em relação à confiança. Em suma, estes gráficos demonstram a proporção de objetos detectados em relação a todos os objetos que deveriam ter sido encontrados sob diferentes valores de confiança. Na literatura, esse gráfico normalmente apresenta uma curva descendente suave, indicando que a revocação tende a diminuir à medida que a confiança aumenta.

Ao analisar o gráfico da *SSD Mobilenet* (Figura 19a), observa-se que, à medida que a confiança aumenta, ocorre uma diminuição da quantidade de objetos detectados pelo modelo. Esse comportamento é particularmente evidente nas classes “*left eye*” e “*right eye*”, as quais apresentam valores de revocação mais baixos a partir do valor de confiança de 0,1, se tornando insuficientemente detectáveis a partir do valor de 0,8 de confiança. Por outro lado, destaca-se especialmente a classe “*cow*” por demonstrar resultados satisfatórios em toda a faixa do gráfico.

Figura 19 – Gráficos de Revocação por confiança dos melhores modelos para as redes *SSD Mobilenet* e *YOLOv8*.



Fonte: Elaborada pelo autor (2023).

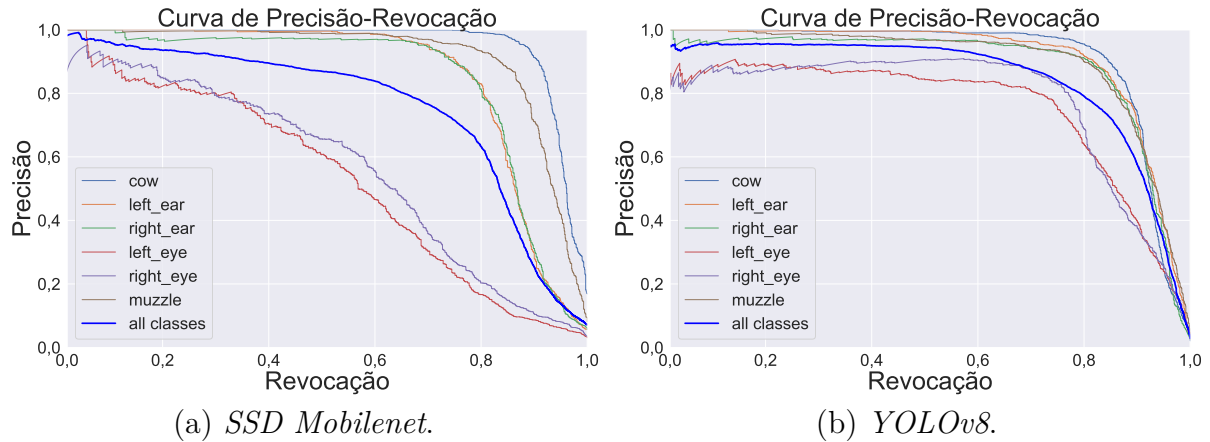
Quanto ao gráfico da *YOLOv8* (Figura 19b), é possível notar que as curvas das classes “*left eye*” e “*right eye*” começam a apresentar uma tendência de queda somente a partir do valor de 0,4 de confiança. Infelizmente, essas classes também se tornam insuficientemente detectáveis em torno do valor de 0,8 de confiança. Em relação às demais classes, seu comportamento é semelhante e quase tão satisfatório quanto o gráfico da *SSD Mobilenet*.

O gráfico Precisão-Revocação (Figura 20), demonstra como a capacidade dos modelos de fazer detecções corretas (precisão) e de identificar todos os objetos reais (revocação) varia, evidenciando a relação inerente entre esses critérios. Por via de regra, quando a precisão é melhorada, geralmente ocorre uma queda na revocação, o que faz com que detecções reais sejam ignoradas. Por outro lado, quando a revocação é melhorada, geralmente ocorrem detecções falsas, o que reduz a precisão. Visualmente, um modelo perfeito teria uma reta localizada na borda superior, com alta precisão e revocação. Entretanto, na literatura, esse gráfico costuma ser uma curva descendente suave, indicando a dificuldade de se encontrar um equilíbrio entre esses critérios.

No gráfico da *SSD Mobilenet* (Figura 20a), nota-se que as curvas correspondentes às classes “*left eye*” e “*right eye*” apresentam um decréscimo relativamente mais acentuado em relação às demais à medida que a revocação aumenta. Esse padrão tem um impacto negativo na curva geral representada pela cor azul, destacando como o baixo desempenho de determinadas classes pode afetar o comportamento geral do modelo. No entanto, é importante ressaltar que as outras classes exibem resultados satisfatórios na maior parte do gráfico, mostrando uma boa performance, mas apresentando uma queda de desempenho vertiginosa após atingirem uma revocação de 0,8.

Em compensação, o gráfico da *YOLOv8* (Figura 20b), mostra as curvas de todas as classes em uma posição mais próxima à borda superior, indicando um desempenho melhor ao longo de todo o gráfico em comparação com a *SSD Mobilenet*. Isso significa

Figura 20 – Gráficos de Precisão pela Revocação dos melhores modelos para as redes *SSD Mobilenet* e *YOLOv8*.

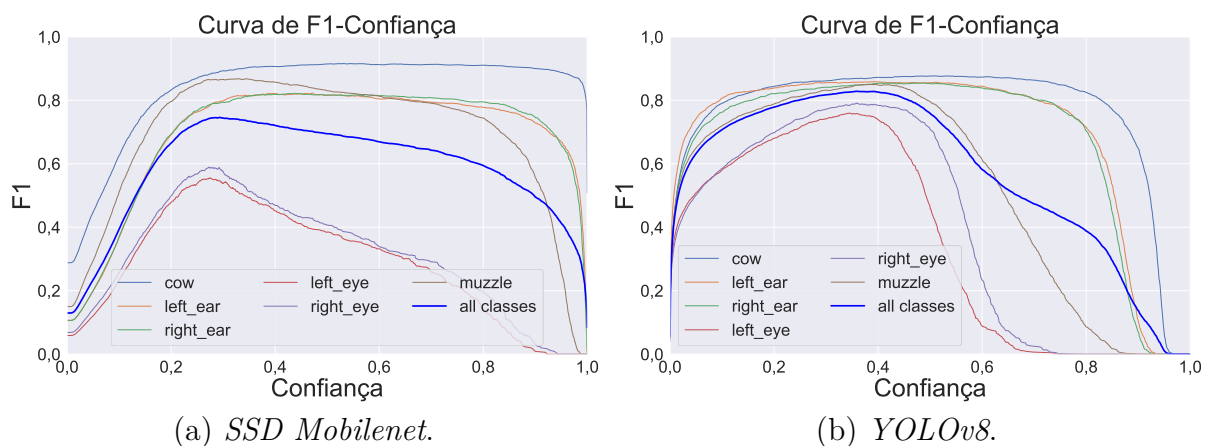


Fonte: Elaborada pelo autor (2023).

que a *YOLOv8* consegue manter uma alta precisão em todas as classes até um nível de revocação mais alto (0,8), mesmo em classes que normalmente possuem um desempenho inferior, como é o caso das classes “*left eye*” e “*right eye*”.

Dado o o comportamento encontrado nos gráficos de Precisão-Revocação, é preciso buscar o ponto de equilíbrio que ofereça a melhor combinação desses critérios. Esse ponto normalmente não é fácil de encontrar, principalmente em gráficos com centenas de pontos. Para isso, a curva de F1-Confiança normalmente é calculada e traçada (Figura 21). Ela é responsável por medir o equilíbrio entre a precisão e a revocação. Quando seu valor é alto, ambas estão equilibradas em valores mais altos. Já valores mais baixos significam que há um desequilíbrio maior entre elas. Em um modelo perfeito, este valor deveria aumentar à medida que a confiança aumenta, indicando que ele mantém um bom equilíbrio para esses critérios em qualquer valor de confiança.

Figura 21 – Gráficos de F1 pela Confiança dos melhores modelos para as redes *SSD Mobilenet* e *YOLOv8*.



Fonte: Elaborada pelo autor (2023).

Observando o gráfico da *SSD Mobilenet* (Figura 21a), pode-se notar que a curva de

$F1$  atinge seu ponto máximo para todas as classes entre 0,2 e 0,4 de confiança. A partir desse ponto, o valor de  $F1$  começa a diminuir para as classes “*left eye*” e “*right eye*”. É interessante notar que a classe “*cow*”, seguida das classes “*muzzle*”, “*left ear*” e “*right ear*” apresentam alto valor de  $F1$  na maior parte do gráfico, o que aponta um bom desempenho para essas classes.

Em contrapartida, ao analisar o gráfico da *YOLOv8* (Figura 21b), é possível ver inicialmente que os valores de  $F1$  são mais consistentes desde os valores mais baixos de confiança para todas as classes. No entanto, as classes “*left eye*”, “*right eye*” e “*muzzle*” começam a decair entre 0,4 e 0,6 de confiança. Portanto, isso sugere que essa rede, ao contrário da *SSD Mobilenet*, tem mais dificuldade em detectar objetos de menor tamanho.

Ao analisar todos gráficos desta seção, pode-se tomar nota de alguns aspectos importantes. Primeiro, observa-se que as classes “*left eye*” e “*right eye*” apresentam um desempenho inferior nas medições, em comparação com as outras classes. Por outro lado, todas as outras classes obtêm resultados satisfatórios, destacando-se especialmente a classe “*cow*”.

Existem duas razões plausíveis para o baixo desempenho das classes que representam os olhos. A primeira é a sua sub-representação no conjunto de dados, como explicado na Subseção 3.1.2. Essa questão é evidenciada na Tabela 4, na qual pode-se observar que tanto a área dos *bounding boxes* quanto o número de exemplos anotados para cada classe são menores para a classe dos olhos em comparação com as demais. Além disso, a Figura 15 representa de forma visual a distribuição da quantidade de anotações por classe. Enquanto a segunda explicação diz respeito à homogeneidade das características dos olhos. Os olhos dos animais são pequenos, tendem a ser predominantemente pretos e suas pálpebras costumam ter tons semelhantes. Em animais de pelagem escura, os olhos se apresentam pouco salientes em relação ao plano de fundo. Como resultado, o interior dos *bounding boxes* anotados para essas classes têm pouca variação de cor.

Por outro lado, a razão para o bom desempenho das demais classes pode ser a heterogeneidade de seus exemplos. Essas classes apresentam uma variedade maior de características, dimensões e plano de fundo, o que torna mais fácil para o modelo aprender a distingui-las corretamente. Além disso, essas classes geralmente possuem uma quantidade maior de *bounding boxes* disponíveis para treinamento, o que contribui para o aprendizado do modelo.

#### 4.5.2 Análise Qualitativa

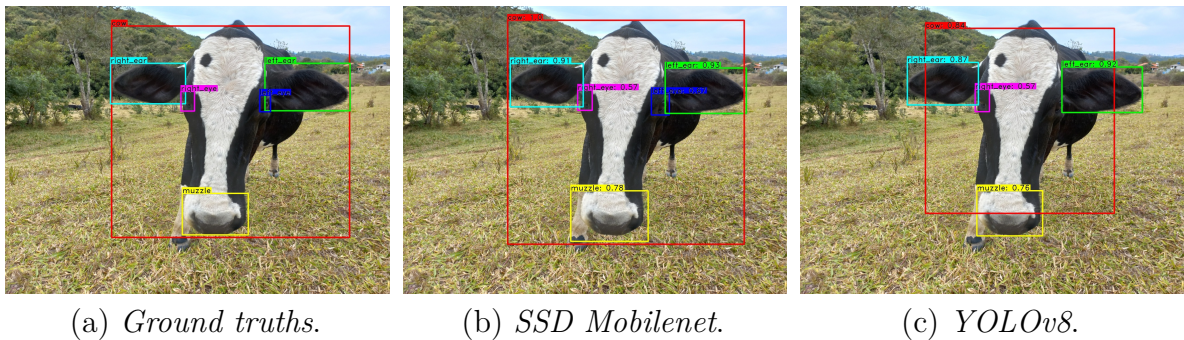
Agora será realizada uma análise qualitativa das detecções dos modelos, utilizando imagens consideradas representativas do conjunto de teste. O objetivo é comparar o comportamento dos modelos em cenários reais. É importante ressaltar que o limiar de confiança utilizado para desenhar as caixas delimitadoras foi 0,5 para ambas as redes.

Dessa forma, pode haver objetos que foram detectados corretamente mas que não foram mostrados nas figuras, por terem confiança inferior ao limiar. Essa abordagem foi adotada para evitar falsos positivos na imagem, o que poderia prejudicar a clareza das análises.

Na primeira imagem de cada figura, são apresentadas as anotações reais das imagens (*ground truth*), seguidas das detecções para a *SSD Mobilenet* e *YOLOv8*, respectivamente. Além disso, vale destacar que não foram utilizadas técnicas de filtragem de *bounding boxes*, como a supressão não-máxima. O objetivo é analisar de forma mais imparcial e pura o desempenho das detecções dos modelos.

No primeiro exemplo (Figura 22), é apresentada uma vaca adulta em uma pastagem. Observa-se que, para ambas as redes, as orelhas do animal, o olho direito e o focinho são detectados com valores de confiança similares e boa localização em relação aos *ground truths*. A face do animal também foi detectada por ambas as redes, com destaque para a *SSD Mobilenet*, que obteve um *bounding box* mais justo em relação ao objeto real. Por fim, evidencia-se que a *YOLOv8*, em comparação à *SSD Mobilenet*, apresentou ao mesmo tempo um enquadramento bom e outro ruim para o focinho e face do animal, respectivamente, além de não ter conseguido identificar o olho esquerdo na imagem.

Figura 22 – Comparação dos *ground truths* e detecções de uma vaca adulta para ambas as redes.

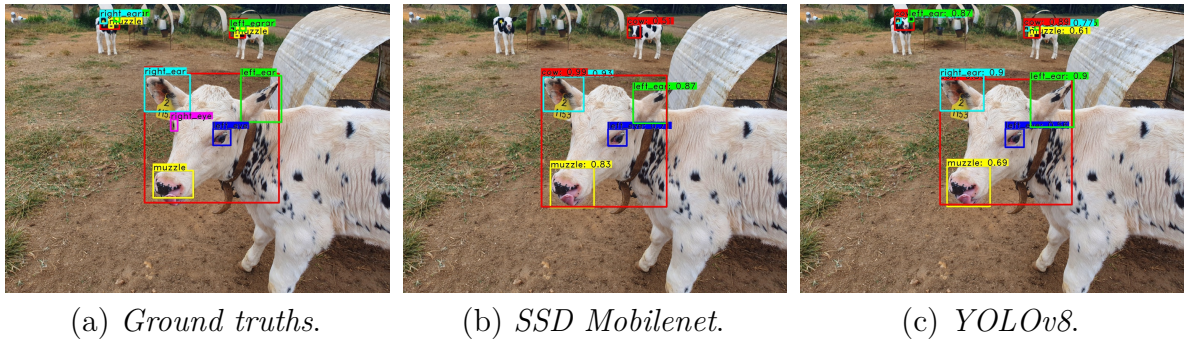


Fonte: Elaborada pelo autor (2023).

No segundo exemplo (Figura 23), existem bezerros próximos aos seus abrigos, conhecidos como “bezerreiras”. Essas instalações são utilizadas em propriedades rurais para manter os bezerros separados dos animais adultos, permitindo que recebam cuidados especializados e um manejo adequado durante os primeiros estágios de vida, incluindo alimentação, higiene e monitoramento de saúde. Sobre os *ground truths*, destaca-se que os olhos dos animais mais distantes foram difíceis de identificar visualmente por conta de sua baixa resolução, por isso eles não foram anotados, conforme protocolo descrito na Subseção 3.1.2.

Em relação ao animal mais próximo da câmera (ao centro), as detecções das duas redes parecem ser semelhantes. Ambas conseguiram detectar corretamente todas as classes com bons valores de confiança, e as áreas delimitadas pelos *bounding boxes*

Figura 23 – Comparação dos *ground truths* e detecções de bezerros para ambas as redes.



Fonte: Elaborada pelo autor (2023).

estão bem posicionadas em relação aos objetos reais. No entanto, ao observar o animal mais distante e à direita, destacam-se as detecções da *YOLOv8*. Nesse indivíduo pelo menos quatro classes foram detectadas, entretanto, não é possível verificar visualmente se elas estão corretamente localizadas. A face foi identificada com uma confiança maior em comparação à face do mesmo animal para a *SSD Mobilenet*. Por fim, é importante destacar que o bezerro mais distante e à esquerda só foi detectado pela *YOLOv8*, com 3 classes identificadas corretamente.

No exemplo apresentado na Figura 24, há a representação de três vacas adultas em um sistema de confinamento chamado “*Compost Barn*”. Esse sistema consiste em um galpão com uma cama comum para todos os animais, feita geralmente de serragem. Essa cama é separada do corredor de alimentação por um beiral de concreto e é iluminada artificialmente por meio de lâmpadas ou luminárias. Na imagem, é possível observar que as faces dos animais estão desfocadas, o que pode representar um desafio para a detecção dos objetos pelas redes. Além disso, ressalta-se que os olhos dos animais mais distantes nem foram anotados por conta da dificuldade de se determinar sua real localização dado o alto nível de desfoque nessas regiões.

Dentre os três animais retratados, a *SSD Mobilenet* conseguiu detectar bem as orelhas, focinho e face do animal ao centro. Entretanto, essa rede foi capaz de detectar apenas a face e o focinho do animal mais próximo (último à direita), deixando de identificar qualquer classe do bovino mais distante (primeiro à esquerda). Em contrapartida, a *YOLOv8* conseguiu detectar todas as faces dos bovinos da imagem, além de identificar corretamente as orelhas e focinhos dos animais mais próximos (o do centro e o da direita). Embora nenhuma das redes tenha conseguido detectar os olhos, os resultados das demais classes demonstram o potencial das redes em detectar objetos em imagens desfocadas.

Na Figura 25, é possível observar um bezerro com a face próxima à câmera, em uma situação de oclusão. A oclusão ocorre quando um ou mais objetos de interesse estão parcial ou completamente obstruídos por outros objetos na cena. Assim como o desfoque, essa situação também representa um desafio para modelos de detecção, em razão de dificultar a



Figura 24 – Comparação dos *ground truths* e das detecções de vacas adultas em uma imagem desfocada para ambas as redes.



(a) *Ground truths*.

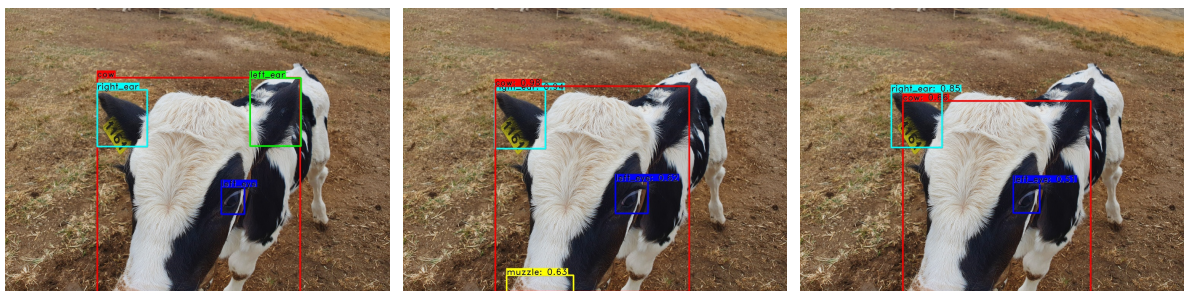
(b) *SSD Mobilenet*.

(c) *YOLOv8*.

Fonte: Elaborada pelo autor (2023).

identificação completa dos objetos. Apesar das circunstâncias, ambos os modelos tiveram desempenho satisfatório para a maioria das detecções dessa imagem, com exceção da orelha esquerda do animal. Em linhas gerais, é evidente o desempenho superior da *SSD Mobilenet* em comparação com a *YOLOv8*, visto que a primeira obteve detecções com maior confiança e enquadrou melhor os *bounding boxes* em relação aos *ground truths*. Além disso, a *SSD Mobilenet* também detectou parte do focinho que está presente na imagem mas que não havia sido originalmente anotado.

Figura 25 – Comparação dos *ground truths* e das detecções de um bezerro sob oclusão para ambas as redes.



(a) *Ground truths*.

(b) *SSD Mobilenet*.

(c) *YOLOv8*.

Fonte: Elaborada pelo autor (2023).

Na Figura 26, evidencia-se a presença de uma vaca adulta em um ambiente de pastagem. Essa imagem é relevante devido a pose lateral do animal e a orientação de sua face, que está voltada para o lado, ou seja, “de perfil”. Nesse ângulo, a orelha direita da vaca acaba obstruindo a identificação do olho direito na imagem. Além disso, observa-se que a face do animal é menos saliente do restante do corpo nessa posição. Conseqüentemente,

o formato triangular da face torna-se mais desafiador de identificar. Essa dificuldade de identificação é ainda mais evidente quando se observa o desempenho de ambas as redes. A *SSD Mobilenet* não conseguiu detectar nenhuma classe, enquanto a *YOLOv8* detectou apenas o focinho.

Figura 26 – Comparação dos *ground truths* e das detecções de uma vaca em posição lateral para ambas as redes.



(a) *Ground truths*.

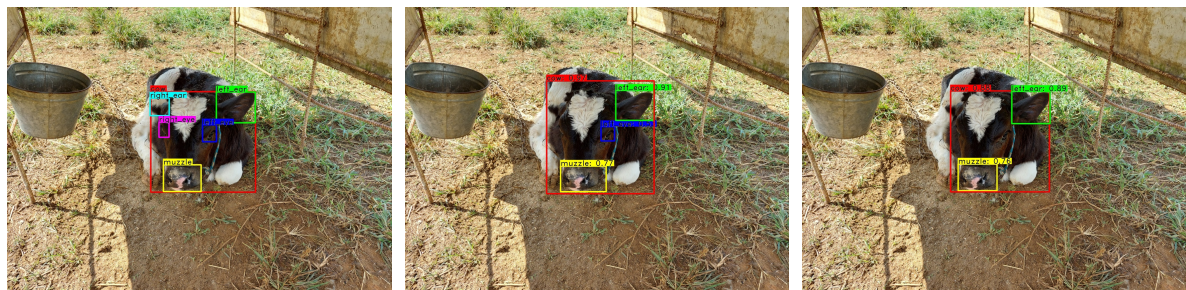
(b) *SSD Mobilenet*.

(c) *YOLOv8*.

Fonte: Elaborada pelo autor (2023).

Na última imagem de exemplo (Figura 27), é apresentado um bezerro manifestando desconforto relacionado à ocorrência de diarreia, cujo diagnóstico foi realizado por um profissional médico veterinário. Esse cenário remete à motivação do trabalho, que consiste em identificar as regiões faciais de bovinos capazes de indicar algum tipo de desconforto. Na referida imagem, podem ser observadas “linhas de sulcos” ao redor dos olhos, muco nas narinas e secreções lacrimais. Em relação às detecções, ambos os modelos conseguiram identificar a maioria das classes presentes na imagem. Ademais, o modelo da *SSD Mobilenet* se sobressaiu por detectar também o olho esquerdo do animal e apresentar confianças ligeiramente mais altas para todas as classes.

Figura 27 – Comparação dos *ground truths* e das detecções de um de bezerro com desconforto para ambas as redes.



(a) *Ground truths*.

(b) *SSD Mobilenet*.

(c) *YOLOv8*.

Fonte: Elaborada pelo autor (2023).

Em geral, a análise qualitativa realizada revelou o panorama de desempenho das redes *SSD Mobilenet* e *YOLOv8*, demonstrando suas vantagens e limitações. As comparações mostraram que ambas foram capazes de detectar características importantes



dos animais, como os olhos, orelhas e focinhos, com bom grau de confiança e localização, apesar das variações na composição das imagens.

A *SSD Mobilenet* se destacou em cenários de maior detalhamento e proximidade dos objetos, proporcionando um enquadramento de caixas delimitadoras mais justo em relação aos *ground truths* e um valor de confiança levemente mais elevado que a *YOLOv8* na maioria dos casos. Por outro lado, a *YOLOv8* mostrou-se eficaz na detecção de objetos em imagens com maior diversidade e complexidade, como aquelas com animais em diferentes poses e ângulos, ou com desfoque. Além disso, a *YOLOv8* demonstrou uma capacidade melhor de detectar objetos mais distantes da câmera, quando comparada à *SSD Mobilenet*.

Os resultados obtidos ressaltaram o potencial das redes em detectar objetos em condições reais, evidenciando a viabilidade prática de sua aplicação em sistemas que requerem a detecção dessas regiões. Isso inclui sistemas de identificação individual de animais, rastreamento do gado ou até mesmo na classificação do estado de saúde do animal. No entanto, é preciso levar em consideração que as imagens escolhidas para ilustrar essa seção podem não refletir adequadamente o desempenho geral dos modelos, uma vez que em outras imagens do conjunto de dados, as detecções de cada rede poderiam ter comportamentos diferentes dos apresentados nesses exemplos.. Assim, as imagens escolhidas serviram meramente para analisar alguns exemplos de detecções boas e ruins em diferentes situações, possibilitando um estudo do comportamento dessas redes nesses casos específicos.

Por fim, vale ressaltar que os resultados das detecções estão diretamente relacionados à qualidade e quantidade de dados utilizados no treinamento. Dessa forma, se torna fundamental garantir o aprimoramento contínuo do conjunto de dados a partir de imagens representativas e de alta qualidade, contendo uma variedade de cenários e objetos a serem detectados. Isso pode envolver a coleta de mais dados relevantes, a anotação cuidadosa dos objetos de interesse e a inclusão de mais casos desafiadores, como imagens com oclusões, desfoque ou iluminação variável.

## 5 CONCLUSÃO

Esse trabalho se propôs a utilizar redes de detecção de objetos em imagens para determinar a localização da face de bovinos e as suas principais regiões de interesse, como os olhos, orelhas e focinho, mesmo em condições adversas de iluminação, ângulo, posição, plano de fundo, entre outras.

Nesse cenário, o presente trabalho propôs a organização de um conjunto de dados de imagens que possuíssem faces de bovinos e suas regiões de interesse. Após o processo de curadoria e anotação, produziu-se um conjunto de dados constituído por 7.146 imagens e 54.111 rótulos. Esse conjunto pode ser usado por outros trabalhos, o que constitui uma importante contribuição desta pesquisa. A partir desse conjunto, foram treinados e analisados modelos de detecção de objetos para duas redes neurais: a *SSD MobileNet V2 FPNLite 640 × 640* e a *YOLOv8*.

A análise quantitativa dos modelos revelou que a *YOLOv8* obteve um desempenho superior à *SSD Mobilenet* nos critérios de avaliação escolhidos. Essa superioridade da *YOLOv8* também é evidenciada no gráfico de Precisão-Revocação. Nos demais gráficos, ambas as redes obtiveram resultados similares entre si e bem alinhados com o que era esperado para cada gráfico. Essa análise demonstrou que a maioria das classes (faces, orelhas e focinho) alcançaram resultados satisfatórios, provavelmente por conta de serem objetos mais salientes e maior disponibilidade de dados de treinamento. Contudo, as classes dos olhos obtiveram desempenho inferior em ambas as redes, possivelmente em razão de sua sub-representação no conjunto de dados, similaridade de características e/ou seu tamanho reduzido em relação às outras classes.

Por sua vez, a análise qualitativa também mostrou que ambas as redes foram capazes de detectar com qualidade as faces, orelhas e focinhos dos animais. Ao contrário da análise quantitativa, nesse caso houve mais destaque para a *SSD MobileNet*, pelo fato dela ter detectado e enquadrado melhor objetos mais próximos, até mesmo em casos de oclusão. Ainda assim, a *YOLOv8* também apresentou ótimos resultados, principalmente em imagens que os objetos estavam desfocados ou mais distantes da câmera.

Em linhas gerais, os resultados destacaram o potencial dessas redes na detecção da face e de suas regiões de interesse em imagens de bovinos, em cenários reais. Isso provou a possibilidade concreta de contribuir para problemas relacionados à Pecuária de Precisão e Visão Computacional que necessitem da etapa de detecção e extração de objetos a partir de imagens, como a classificação de estado de saúde e bem-estar dos animais.

Embora este trabalho tenha demonstrado o potencial das redes de detecção de objetos dado o problema proposto, ainda há amplo espaço para inovações e melhorias. Algumas delas podem contemplar a coleta e anotação de mais dados, a execução de novos experimentos com outras redes e/ou parâmetros de treinamento e a combinação das

saídas dos modelos de ambas as redes (*ensemble*). Outro ponto focal pode ser um estudo para entender o motivo pelo qual as classes dos olhos obtiveram resultados inferiores às demais, e com isso, desenvolver estratégias que possam aprimorar o modelo para lidar com essa situação. Dessa forma, acredita-se que é possível alcançar resultados ainda mais promissores em trabalhos futuros.

## REFERÊNCIAS

- ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>.
- ADHIKARI, B.; HUTTUNEN, H. Iterative bounding box annotation for object detection. In: IEEE. *2020 25th International Conference on Pattern Recognition (ICPR)*. Milano, Italy, 2021. p. 4040–4046.
- AMIT, Y.; FELZENSZWALB, P.; GIRSHICK, R. Object detection. *Computer Vision: A Reference Guide*, Springer, p. 1–9, 2020.
- BASSOI, L. et al. Agricultura de precisão: resultados de um novo olhar. Brasília, DF: Embrapa, 2014. 596 p., 2014.
- BEZSONOV, O. et al. Breed recognition and estimation of live weight of cattle based on methods of machine learning and computer vision. *Eastern-European Journal of Enterprise Technologies*, v. 6, n. 9, p. 114, 2021.
- BURKOV, A. *The hundred-page machine learning book*. Quebec: Andriy Burkov Quebec City, QC, Canada, 2019. v. 1.
- CARVALHO, T. B. de; ZEN, S. D. A cadeia de pecuária de corte no brasil: evolução e tendências. *Revista iPecege*, v. 3, n. 1, p. 85–99, 2017.
- CHIU, Y.-C. et al. Mobilenet-ssdv2: an improved object detection model for embedded systems. In: IEEE. *2020 International conference on system science and engineering (ICSSE)*. Kagawa, Japan, 2020. p. 1–5.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. San Diego, California, USA, 2005. v. 1, p. 886–893.
- DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. Miami, Florida, USA, 2009. p. 248–255.
- DING, J. et al. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2021.
- DU, J. Understanding of object detection based on cnn family and yolo. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. Hong Kong, China, 2018. v. 1004, n. 1, p. 012029.
- EVERINGHAM, M. et al. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, v. 88, n. 2, p. 303–338, jun. 2010.
- FAWCETT, T. An introduction to roc analysis. *Pattern recognition letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006.

- FERRAZ, C. de O.; PINTO, W. F. Tecnologia da informação para a agropecuária: utilização de ferramentas da tecnologia da informação no apoio a tomada de decisões em pequenas propriedades. *Revista Eletrônica Competências Digitais para Agricultura Familiar*, v. 3, n. 1, p. 38–49, 2017.
- GIRSHICK, R. Fast r-cnn. In: . Santiago, Chile: Proceedings of the IEEE international conference on computer vision, 2015. p. 1440–1448.
- GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Columbus, Ohio, USA: [s.n.], 2014. p. 580–587.
- GLEERUP, K. B. et al. Pain evaluation in dairy cattle. *Applied Animal Behaviour Science*, Elsevier, v. 171, p. 25–32, 2015.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. Cambridge, Massachusetts, USA: The MIT Press, 2016.
- HE, K. et al. Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision*. Honolulu, Hawaii, USA: [s.n.], 2017. p. 2961–2969.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas, Nevada, USA: [s.n.], 2016. p. 770–778.
- HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- IBGE. *Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal 2019*. Rio de Janeiro, RJ, Brasil: IBGE Rio de Janeiro, 2019.
- IBGE. Produção da pecuária municipal. v. 47, p. 1–8, 2020.
- JABBAR, M. et al. *Machine Learning Methods for Signal, Image and Speech Processing*. New York, NY, USA: CRC Press, 2022.
- JIANG, X. et al. *Deep learning in object detection and recognition*. Springer, 2019.
- JIAO, L. et al. A survey of deep learning-based object detection. *IEEE access*, IEEE, v. 7, p. 128837–128868, 2019.
- JOCHER, G.; CHAURASIA, A.; QIU, J. *YOLO by Ultralytics*. 2023. Disponível em: <<https://github.com/ultralytics/ultralytics>>.
- KAMAL, K. et al. Impacts of background removal on convolutional neural networks for plant disease classification in-situ. *Agriculture*, MDPI, Open Access Journal, v. 11, n. 9, p. 1–17, 2021.
- KANG, X.; ZHANG, X.; LIU, G. Accurate detection of lameness in dairy cattle with computer vision: A new and individualized detection strategy based on the analysis of the supporting phase. *Journal of dairy science*, Elsevier, v. 103, n. 11, p. 10628–10638, 2020.
- KHAN, M. A.; KHAN, R.; ANSARI, M. A. (Ed.). *Application of Machine Learning in Agriculture*. Academic Press, 2022. i-iii p. ISBN 978-0-323-90550-3. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780323905503000229>>.

- KUZNETSOVA, A. et al. The open images dataset v4. *International Journal of Computer Vision*, Springer, v. 128, n. 7, p. 1956–1981, 2020.
- LEMOIGNE, Y.; CANER, A. *Molecular Imaging: Computer Reconstruction and Practice*. Dordrecht, The Netherlands: Springer Science & Business Media, 2008.
- LIN, T. *LabelImg*. 2015. Disponível em: <<https://github.com/heartexlabs/labelImg>>.
- LIN, T.-Y. et al. Microsoft coco: Common objects in context. In: SPRINGER. Zurich, Switzerland, 2014. p. 740–755.
- LIU, W. et al. Ssd: Single shot multibox detector. In: SPRINGER. Amsterdam, The Netherlands: European conference on computer vision, 2016. p. 21–37.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Springer, v. 60, n. 2, p. 91–110, 2004.
- L'HEUREUX, A. et al. Machine learning with big data: Challenges and approaches. *Ieee Access*, IEEE, v. 5, p. 7776–7797, 2017.
- MA, J.; USHIKU, Y.; SAGARA, M. The effect of improving annotation quality on object detection datasets: A preliminary study. In: . New Orleans, Louisiana, USA: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. p. 4850–4859.
- MEENA, S. D.; AGILANDEESWARI, L. Smart animal detection and counting framework for monitoring livestock in an autonomous unmanned ground vehicle using restricted supervised learning and image fusion. *Neural Processing Letters*, Springer, v. 53, n. 2, p. 1253–1285, 2021.
- MICHELUCCI, U. *Advanced applied deep learning: convolutional neural networks and object detection*. New York, NY, USA: Springer, 2019.
- NEETHIRAJAN, S. Happy cow or thinking pig? wur wolf—facial coding platform for measuring emotions in farm animals. *AI*, MDPI, v. 2, n. 3, p. 342–354, 2021.
- NING, K. et al. An intelligent surveillance system based on lightweight object detection network l-ssd. In: IEEE. Chengdu, China: 2019 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA), 2019. p. 156–157.
- NVIDIA; VINGELMANN, P.; FITZEK, F. H. *CUDA, release: 10.2.89*. 2020. Disponível em: <<https://developer.nvidia.com/cuda-toolkit>>.
- PADILLA, R.; NETTO, S. L.; SILVA, E. A. D. A survey on performance metrics for object-detection algorithms. In: IEEE. Niteroi, Rio de Janeiro, Brasil.: 2020 international conference on systems, signals and image processing (IWSSIP), 2020. p. 237–242.
- PASZKE, A. et al. Pytorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: <<http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>>.
- REDMON, J. et al. You only look once: Unified, real-time object detection. In: . Las Vegas, Nevada, USA: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. p. 779–788.

- REDMON, J.; FARHADI, A. Yolo9000: better, faster, stronger. In: . Honolulu, Hawaii, USA: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. p. 7263–7271.
- REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, v. 28, 2015.
- ROH, Y.; HEO, G.; WHANG, S. E. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 33, n. 4, p. 1328–1347, 2019.
- SALAU, J.; KRIETER, J. Analysing the space-usage-pattern of a cow herd using video surveillance and automated motion detection. *Biosystems Engineering*, Elsevier, v. 197, p. 122–134, 2020.
- SANTOS, T. T. et al. Visão computacional aplicada na agricultura. *Embrapa Agricultura Digital-Capítulo em livro científico (ALICE)*, In: MASSRUHÁ, SMFS; LEITE, MA de A.; OLIVEIRA, SR de M.; MEIRA, CAA . . . , 2020.
- SHAO, W. et al. Cattle detection and counting in uav images based on convolutional neural networks. *International Journal of Remote Sensing*, Taylor & Francis, v. 41, n. 1, p. 31–52, 2020.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- SZELISKI, R. *Computer vision: algorithms and applications*. Cham, Switzerland: Springer Science & Business Media, 2022.
- TEIXEIRA, J. C.; HESPANHOL, A. N. A trajetória da pecuária bovina brasileira. *Caderno Prudentino de Geografia*, v. 2, n. 36, p. 26–38, 2014.
- THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition*. 4. ed. San Diego, CA: Academic Press, 2008.
- TSCHARKE, M.; BANHAZI, T. M. A brief review of the application of machine vision in livestock behaviour analysis. *Agrárinformatika/Journal of Agricultural Informatics*, Hungarian Association of Agricultural Informatics, v. 7, n. 1, p. 23–42, 2016.
- USDA, F. Livestock and poultry: World markets and trade. usa department of agriculture. *Foreign Agriculture Service*, 2022.
- VIOLA, P.; JONES, M. J. Robust real-time face detection. *International journal of computer vision*, Springer, v. 57, n. 2, p. 137–154, 2004.
- WENG, L. Object detection part 4: Fast detection models. *lilianweng.github.io*, 2018. Disponível em: <<https://lilianweng.github.io/posts/2018-12-27-object-recognition-part-4/>>.
- YANG, S. et al. Wider face: A face detection benchmark. In: . Las Vegas, Nevada, USA: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

YAO, L. et al. Cow face detection and recognition based on automatic feature extraction algorithm. In: *Proceedings of the ACM Turing Celebration Conference - China*. New York, NY, USA: Association for Computing Machinery, 2019. (ACM TURC '19).

YU, F. et al. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: . Seattle, Washington, USA: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

ZHANG, A. et al. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

ZHAO, Q. et al. Cfenet: An accurate and efficient single-shot object detector for autonomous driving. *arXiv preprint arXiv:1806.09790*, 2018.

ZOU, Z. et al. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.