



UNIVERSIDADE
ESTADUAL DE LONDRINA

ANGELICA CRISTINA MANFRIM

**ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE
DETECÇÃO E CONTAGEM DE GADO EM IMAGENS
AÉREAS UTILIZANDO REDES NEURAIAS**

ANGELICA CRISTINA MANFRIM

**ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE
DETECÇÃO E CONTAGEM DE GADO EM IMAGENS
AÉREAS UTILIZANDO REDES NEURAIIS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual de Londrina como parte dos requisitos necessários para a obtenção do Título de Mestre em Engenharia Elétrica.

Orientadora: Profa. Dra. Maria Bernadete de Moraes França

Londrina
2022

Ficha Catalográfica

ANGELICA CRISTINA MANFRIM

ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE DETECÇÃO E
CONTAGEM DE GADO EM IMAGENS AÉREAS UTILIZANDO REDES
NEURAIIS - Londrina - PR, 2022 - 42 p., 30 cm.

Orientadora: Profa. Dra. Maria Bernadete de Moraes França

Pecuária; Dataset; Drone.

I. Universidade Estadual de Londrina. Curso de Engenharia Elétrica. II.
ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE DETECÇÃO E
CONTAGEM DE GADO EM IMAGENS AÉREAS UTILIZANDO REDES
NEURAIIS.

ANGELICA CRISTINA MANFRIM

**ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE
DETECÇÃO E CONTAGEM DE GADO EM IMAGENS
AÉREAS UTILIZANDO REDES NEURAIIS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual de Londrina como parte dos requisitos necessários para a obtenção do Título de Mestre em Engenharia Elétrica.

BANCA EXAMINADORA

Orientadora: Profa. Dra. Maria Bernadete de Moraes
França
Universidade Estadual de Londrina – UEL

Leonimer Flávio de Melo
Universidade Estadual de Londrina – UEL

Flávio José de Oliveira Moraes
Faculdade de Ciências e Engenharia – Câmpus de
Tupã

Londrina, 29 de setembro de 2022

Agradecimentos

Agradeço primeiramente a Deus por tornar tudo isso possível e me ajudar a atravessar esses tempos difíceis de pandemia.

Agradeço a minha família: ao meu pai Antonio, minha mãe Floripes, meu irmão Junior, meus avós: Antenor e Lourdes, Josias e Hilda. Ao meu amado e companheiro Marcelo. A minha gata Luna e minhas cachorras: Choin, Kaira, Lilinus e Paraguaio.

Agradeço a todos os professores que passaram por minha formação na UEL e em especial, a minha orientadora Maria Bernadete que teve toda paciência e todo apoio para que esse trabalho fosse realizado, e aos professores da banca por suas contribuições. Agradeço também ao professor Reginaldo por novamente nos incentivar. Agradeço a instituição UEL e o Laboratório de Automação e Instrumentação (LA2I).

Agradeço aos meus colegas de laboratório Giuliano, João e Sérgio, por todo apoio que me deram durante estes três longos anos, éramos o melhor clã de PUBG Lite e sei que continuaremos amigos além do mestrado. Agradeço também a Andressa e ao Gustavo por terem dividido comigo essa caminhada. Gostaria também de deixar um salve aos meus amigos da UTFPR de Campo Mourão presentes no grupo Shenzhen que torceram por mim e pelo sucesso do meu trabalho.

Este trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Código de Financiamento 001.

"Para um sistema evolutivo, é necessário haver um desenvolvimento contínuo para manter a aptidão relativa aos sistemas com o qual estão a co-evoluir".
(Hipótese da rainha vermelha)

ANGELICA CRISTINA MANFRIM. **ESTUDO E DESENVOLVIMENTO DE UM SISTEMA DE DETECÇÃO E CONTAGEM DE GADO EM IMAGENS AÉREAS UTILIZANDO REDES NEURAIS**. 2022. 42 p. Dissertação do Programa de Mestrado em Engenharia Elétrica - Universidade Estadual de Londrina, Londrina - PR.

Resumo

O monitoramento da quantidade de gado presente em um rebanho é de suma importância para a agropecuária, sendo atualmente realizado de forma invasiva e estressante para os animais, com técnicas como a de brinco RFID. Pesquisas vem sendo realizadas utilizando a aplicação redes neurais e a aprendizagem profunda na solução deste problema. A partir de imagens aéreas de drone e um sistema de detecção de objetos, é possível realizar a contagem à distância, sem necessitar movê-los do local aberto em que estão. Para isso é necessário um *dataset* com imagens dos animais para que a rede neural consiga aprender, identificar e analisar. Com a utilização de um *dataset* disponível, foi possível projetar o sistema para mapear e quantificar. Neste trabalho foram utilizadas as redes Resnet-50 , Resnet-101, SSD-Inception-V2 e YOLOv4, e as métricas de PASCAL VOC, chegou-se aos resultados de precisão da detecção. O *framework* de contagem funcionou como esperado, nota-se a necessidade de repetir os experimentos e a aplicação de novas redes neurais. O *framework* está disponível para utilização em estudos.

Palavras-Chave: Pecuária; Dataset; Drone.

ANGELICA CRISTINA MANFRIM. **STUDY AND DEVELOPMENT OF A CATTLE DETECTION AND COUNTING SYSTEM IN AERIAL IMAGES USING NEURAL NETWORKS**. 2022. 42 p. Dissertation on a Master's Degree in Electrical Engineering - Londrina State University, Londrina - PR.

Abstract

Monitoring the number of cattle present in a herd is extremely important for agriculture., currently being carried out in an invasive and stressful way for animals, with techniques such as RFID tag. Research has been carried out using the application of neural networks and deep learning to solve this problem. From aerial drone images and an object detection system, it is possible to count remotely, without having to move them from the open place where they are. For this, a dataset with images of the animals is needed so that the neural network can learn, identify and analyze. Using an available dataset, it was possible to design the system to map and quantify. In this work, the Resnet-50, Resnet-101, SSD-Inception-V2 and YOLOv4 networks were used, as well as the PASCAL VOC metrics, reaching the detection accuracy results. The counting framework worked as expected, there is a need to repeat the experiments and the application of new neural networks. The framework is available for use in studies.

Key-words: Cattle raising; Dataset; Drone

Lista de ilustrações

Figura 1 – Ciclo de gerenciamento baseado em informações para agricultura avançada.	18
Figura 2 – Exemplos de Redes Neurais.	20
Figura 3 – Topologia de rede neural multicamadas.	21
Figura 4 – Diferença entre <i>Deep Learning</i> e Redes Neurais simples.	22
Figura 5 – Como funciona uma Rede Neural Convolutacional.	24
Figura 6 – Arquitetura da rede Resnet-101.	24
Figura 7 – Arquitetura da rede SSD-Inception-V2	25
Figura 8 – Arquitetura da rede YOLOv4	25
Figura 9 – Classificação e detecção de objetos.	26
Figura 10 – Fluxo de trabalho de <i>Machine Learning</i>	28
Figura 11 – Exemplos de imagens presentes no <i>dataset</i>	30
Figura 12 – Exemplo de rotulagem no programa labelImg.	31
Figura 13 – Fluxograma simplificado do <i>framework</i>	33
Figura 14 – Resnet-50 <i>Total loss</i>	35
Figura 15 – Resnet-101 <i>Total loss</i>	36
Figura 16 – SSD-Inception-V2 <i>Total loss</i>	36
Figura 17 – YOLOv4 <i>Average total loss</i> e <i>Total loss</i>	37
Figura 18 – <i>Frames</i> do vídeo de saída do <i>framework</i>	38

Lista de tabelas

Tabela 1 – Hiperparâmetros utilizados nas redes Resnet-50, Resnet-101 e SSD-Inception-V2.	32
Tabela 2 – Hiperparâmetros usados na rede YOLOv4.	32
Tabela 3 – Matriz de confusão.	34
Tabela 4 – Métricas e seus valores resultantes do processo de treinamento.	35

Sumário

	Lista de ilustrações	9
	Lista de tabelas	10
	Sumário	11
1	INTRODUÇÃO	13
1.1	Trabalhos relacionados	15
1.2	Objetivos	16
1.3	Organização do trabalho	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Agricultura 5.0 e <i>Smart Farming</i>	17
2.2	Redes Neurais	19
2.3	Visão computacional	21
2.4	Machine Learning	22
2.5	Deep Learning	22
2.6	Redes Neurais Convolucionais	23
2.6.1	Arquitetura das redes	24
2.7	Classificação, Detecção e Rastreamento de Objetos	25
2.7.1	Classificação	25
2.7.2	Detecção de objetos	26
2.7.3	Rastreamento de objetos	26
3	METODOLOGIA	28
3.1	Equipamentos e ferramentas	29
3.2	<i>Dataset</i>	29
3.3	Treinamento	31
3.4	<i>Framework</i>	33
3.5	Métricas de validação	33

4	RESULTADOS E DISCUSSÃO	35
5	CONCLUSÃO	39
5.1	Trabalhos futuros	39
	REFERÊNCIAS	40

1 Introdução

A pecuária desempenha um papel significativo nos meios de subsistência rurais e nas economias dos países em desenvolvimento. São provedores de renda e emprego para produtores e outros em trabalhos relacionados. Fornecem uma importante fonte de nutrição para bilhões de pessoas. Esses papéis socioeconômicos e outros estão aumentando em importância à medida que o setor cresce devido ao crescente aumento da população humana (HERRERO et al., 2013).

Em 2018 a agropecuária ocupou o equivalente a 18,6% do território total do país e somente a produção de bovinos movimentou cerca de 78,5 bilhões de reais em valor bruto. O Brasil possui atualmente o maior rebanho bovino comercial de todo o mundo. Em 2018 foram estimados 213,5 milhões de exemplares, a produção de leite chegou a 33,8 bilhões de litros, tendo um valor aproximado de 39,3 bilhões de reais (IBGE, 2020).

O Ministério da Agricultura, Pecuária e Abastecimento disponibiliza um documento mensal com o número de bovinos, apresentando métricas de quantidade de cabeças, valor bruto e crescimento anual. Com base nesses documentos, notamos o crescimento contínuo da pecuária e criação de gado no país, demonstrando sua importância econômica (MAPA, 2021).

Segundo pesquisas realizadas, há atualmente uma crescente preocupação por parte dos consumidores com o bem-estar animal, além disso existem grandes lacunas no rendimento na maioria dos sistemas de produção. Nota-se a necessidade de práticas sustentáveis, possibilitando aumentando a eficiência da produção visando o social e ambiental, sendo esta uma área com grandes oportunidades de pesquisa e desenvolvimento (HERRERO et al., 2013).

Em casos de rebanhos criados soltos e com grande número de animais, é difícil manter o controle sobre a quantidade exata de animais, além da identificação individual de cada espécime. O principal método atual de controle são os brincos, colocados na orelha dos animais de forma invasiva e estressante, os brincos possuem uma identificação individual que possibilita saber o histórico do animal (SCHMIDEK, 2009).

Existem brincos não apenas com a numeração, mas com tecnologias como RFID (*Radio-Frequency IDentification*, ou Identificação por Rádio Frequência em português),

que possibilitam a leitura a certa distância do animal. Os principais problemas dos brincos são os fatos de poderem quebrar, cair, enroscar e afins. Na contagem de gado utilizando RFID, é necessário realizar o confinamento do gado por meio de deslocamento para a aquisição dos dados, o que gera estresse ao animal (SCHMIDEK, 2009).

Outra forma de contagem do rebanho seria a inspeção visual humana, em campo aberto, porém, esta seria uma tarefa árdua e suscetível a erros, principalmente para grandes rebanhos. Portanto, sistemas automatizados focados em análise de imagens para a contagem do rebanho estão sendo desenvolvidos, tornando a tarefa mais ágil e confiável (BARBEDO et al., 2019).

O reconhecimento de imagens é uma parte importante da visão computacional, baseado na análise de *pixels* e padrões de imagem, utilizado para reconhecer objetos e textos, dentre os quais envolvem uma grande quantidade de matemática e estatística. Contudo, com a utilização de redes neurais, é possível realizar o mapeamento de um volume maior de dados devido à robustez dos algoritmos (CHOLLET, 2017).

A aprendizagem de máquina (*machine learning*) é um ramo da inteligência artificial, utilizado para automatizar a construção de modelos analíticos. Foi baseado na ideia de que as máquinas poderiam identificar padrões, tomar decisões e aprender com dados, com mínima intervenção humana. Uma das maneiras atuais e que estão sendo amplamente utilizadas no aprendizado de uma máquina é por meio de redes neurais, uma ferramenta poderosa que permite reconhecer padrões em dados, agrupá-los, classificá-los e aprender com os mesmos. Para processar informações complexas, ou seja, um grande volume de dados não estruturados, desenvolveu-se os sistemas de aprendizagem profunda (deep learning), que são redes neurais com várias camadas (CHOLLET, 2017).

O processo de trabalho da pesquisa iniciou-se na busca por um *dataset open-source*. Nos *datasets* usuais e com grande volume de dados como o “Open Images Dataset” e “ImageNet”, não foram encontradas imagens aéreas de gado, apenas imagens laterais. O *dataset* Okafor et al. (2019) está disponível e foi utilizado para os testes preliminares. Foi realizado o treinamento de diversas redes utilizando uma plataforma nuvem, aplicando os mesmos hiperparâmetros para se realizar uma comparação entre os resultados obtidos de cada uma. Desenvolveu-se um *framework* em Python que realiza a contagem atual e total dos animais. Foram aplicadas métricas de validação para a comparação entre o desempenho das redes utilizadas, os resultados foram analisados e avaliados.

1.1 Trabalhos relacionados

Andrew, Greatwood e Burghardt (2017) desenvolveram um *dataset* com aproximadamente 1.000 imagens aéreas de 89 vacas da raça Holandesa. Utilizando um drone em conjunto de redes neurais convolucionais, foram identificadas quantos exemplares existiam em uma determinada área. O sistema funcionou com uma precisão de 99% em campo aberto, porém foi testado apenas para vacas dessa raça, não diferenciando características como: raça, idade, peso, tamanho aproximado e afins.

Em Rivas et al. (2018) foi desenvolvido um sistema baseado em redes neurais convolucionais e utilizando como equipamento um drone, para fazer a contagem de vacas. Foram utilizadas 13.520 imagens, divididas entre as imagens usadas para treinamento (10.816 amostras) e as imagens usadas para teste (2.704 amostras). No processo de treinamento, obteve-se uma precisão média de 97,1% e no processo de teste 95,5%. Os animais em grupos são identificados com uma precisão de 87% o que poderia ser melhorado com o uso de outros algoritmos, e há a necessidade de se evitar que quando o mesmo animal cruze novamente a câmera, seja contado novamente.

Em Barbedo et al. (2019), foram utilizadas imagens aéreas da raça de gado Canchim, no qual foram treinadas 15 diferentes redes neurais convolucionais. Todas estas apresentaram resultados de precisão acima de 95%, todavia a rede que apresentou melhores resultados foi a NasNet Large - com precisão próxima a 100%. Foram analisadas imagens com problemas de iluminação, não se notando diferenças na precisão resultante entre elas, exceto em casos que a cor dos animais se assemelhava em muito a do solo, resultando em erros de detecção. Nota-se a necessidade de testes com outras raças de gado, já que o comportamento das redes pode variar devido a isso.

Shao et al. (2020) realizaram a notação individual de 212 exemplares, gerando um total de 1.886 anotações, em imagens aéreas provenientes de um drone, conseguindo uma precisão de 77%, utilizando o treinamento da rede YOLOv2. Analisou-se que o movimento lento do gado é favorável a aplicação de detecção de imagens para a contagem dos animais. Notou-se também a possibilidade de se aplicar o estudo a outros animais de movimento lento.

Em Soares et al. (2021) desenvolveu-se um método baseado em gráficos para se remover duplicatas de gado na contagem, com dois protocolos de exclusão e comparação das imagens do pasto, utilizando no total 1.337 imagens de gado. A rede utilizada no tra-

balho foi a Faster RCNN Inception Resnet V2, com uma previsão de 92.8%. Chegou-se a conclusão que aplicar o rastreamento ao sistema seria interessante para acompanhar o movimento dos animais.

Em Porto et al. (2021) utilizou-se 90 imagens de gado para testes utilizando a ResNet-50 e VGG16 em imagens sem *data augmentation* e com inversão horizontal e rotação em noventa graus, chegando a um resultado máximo de 89,7% de precisão. Concluindo que é possível a utilização deste sistema porém nota-se necessidade de melhorias do trabalho devido aos resultados não satisfatórios.

Mücher et al. (2022) comparou a utilização de imagens de satélite, drone e aeronave tripulada na detecção de gado, chegando a conclusão que o drone pela proximidade e qualidade das imagens é mais adequado, tendo o porém de necessitar de condições climáticas adequadas, sem chuva ou ventanias muito fortes. Ele utilizou a rede YOLO-3 e conseguiu uma precisão de 95% na detecção do gado utilizando as imagens provenientes do drone.

1.2 Objetivos

O objetivo geral deste trabalho é o desenvolvimento de um *framework*, com base em aprendizagem de máquina e redes neurais para a realização de detecção e contagem dos animais, auxiliando no controle e segurança do rebanho.

Os objetivos específicos são realizar a anotação manual do *dataset open-source*, treinamento de diversas redes neurais, desenvolvimento de um *framework* em Python e analisar dos resultados realizando uma comparação entre o desempenho das redes implementadas.

1.3 Organização do trabalho

Este trabalho está organizado nos próximos capítulos na seguinte estrutura: no capítulo 2 será apresentado o referencial teórico resumido e necessário para o entendimento dos conceitos do trabalho pelo leitor, no capítulo 3 é apresentada a metodologia utilizada, no capítulo 4 estão apresentadas os resultados obtidos durante a condução da pesquisa e no capítulo 5 as conclusões e sugestões de trabalhos futuros.

2 Fundamentação Teórica

Este capítulo apresenta os conceitos teóricos relacionados a redes neurais e detecção de objetos necessários para o entendimento do trabalho.

2.1 Agricultura 5.0 e *Smart Farming*

A agricultura está em constante evolução, com o interesse e aplicação de novas tecnologias buscando-se mais produtividade, lucratividade e sustentabilidade. As principais vantagens para os produtores que decidem investir em tecnologia são: economia de dinheiro e trabalho, aumento da produção ou redução de custos com mínimo esforço e produção de alimentos de qualidade com práticas mais ecologicamente corretas. O maior desafio é tornar os dados coletados em aplicações úteis, sendo este o trabalho do sistema de processamento de dados. Tecnologias podem então fornecer sistemas para detecção de problemas imprevistos, difíceis de perceber em uma inspeção visual rotineira do trabalhador (SAIZ-RUBIO; ROVIRA-MAS, 2020), (SIDDHARTH; SAINI; KUMAR, 2021).

Com o avanço da inserção de tecnologias no campo, criou-se o termo Agricultura 4.0, que é uma integração de *software* e equipamentos agrários. Os dois grandes pilares da Agricultura 4.0 são a Internet das Coisas (IoT) e o *Big Data* (análise massiva de dados). Atualmente avançou-se para a Agricultura 5.0, que utiliza inteligência artificial e robótica. As fazendas estão seguindo os princípios da Agricultura de Precisão e utilizando equipamentos que envolvem operações não tripuladas e sistemas autônomos de apoio à decisão, conhecidas então como *smart farmings*. Os robôs agrícolas aumentam a força de trabalho humana e podem colher safras em um volume maior e em ritmo mais rápido do que os trabalhadores humanos. Apesar de ainda serem tecnologias consideradas de valor elevado, a tendência é este custo abaixar ao longo do tempo e se tornar mais acessível (SAIZ-RUBIO; ROVIRA-MAS, 2020).

Na Figura 1 está apresentado um fluxograma de um exemplo de sistema de Agricultura 5.0.

etapa da atuação é a execução física de alguma decisão sobre a safra, podendo ser realizada por meio de equipamentos agrícolas automatizados com comunicação com o sistema (SAIZ-RUBIO; ROVIRA-MAS, 2020).

Os drones são uma importante ferramenta para a aquisição de dados, sendo possível a instalação de uma grande variedade de sensores neles para isto, como câmeras convencionais e infravermelho. Drones estão sendo utilizados para detectar doenças e pragas nas safras e para monitorar seu estado de reprodução e irrigação, sendo atualmente uma ferramenta da Agricultura 5.0 (SIDDHARTH; SAINI; KUMAR, 2021).

2.2 Redes Neurais

A ideia de uso de redes neurais foi baseada no funcionamento do cérebro humano pois, este funciona de maneira totalmente diferente de um computador convencional. O cérebro é um computador altamente complexo, possuindo a capacidade de organizar seus constituintes estruturais, conhecidos como neurônios, de modo a executar certos cálculos (HAYKIN et al., 2009).

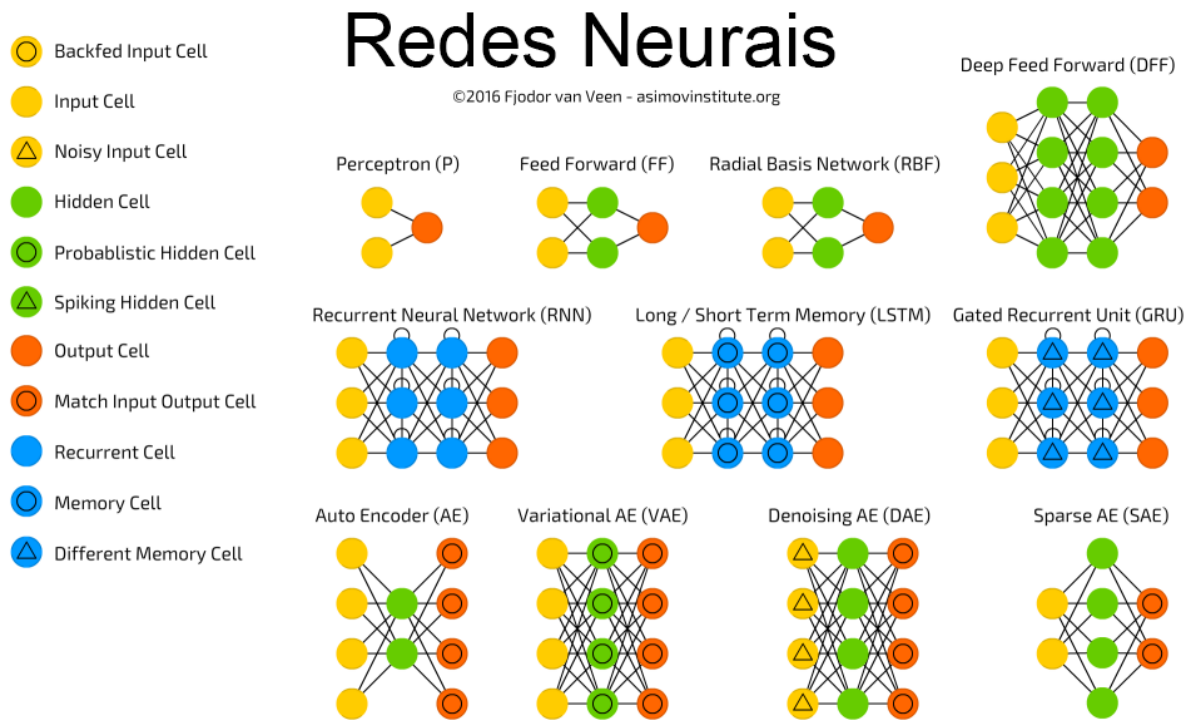
O cérebro realiza rotineiramente tarefas de reconhecimento perceptivo. A visão humana pode ser considerada uma tarefa de processamento de informações, fornecendo uma representação do ambiente para que possamos interagir com o mesmo (HAYKIN et al., 2009).

Patterson e Gibson (2017) define que:

"Redes neurais são um modelo computacional no qual muitas unidades simples estão trabalhando em paralelo, sem unidade de controle. Os pesos entre as unidades são o principal meio de definir o armazenamento de informações em redes neurais. Atualizar os pesos é a principal forma de rede neural aprender novas informações".

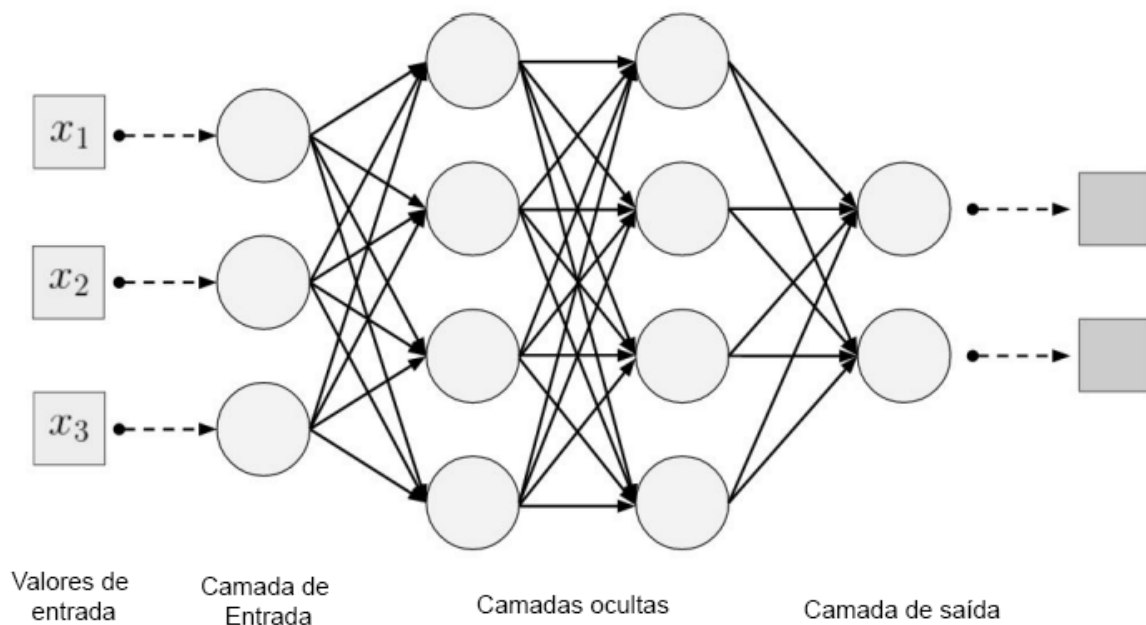
Uma arquitetura de rede neural pode ser definida pela seguinte estrutura: número de neurônios, número de camadas e tipos de conexões entre as camadas. Na Figura 2 estão demonstrados a composição e exemplos de redes neurais.

Figura 2 – Exemplos de Redes Neurais.



Fonte: Adaptado de (THE ASIMOV INSTITUTE, 2016).

Em Patterson e Gibson (2017), é possível compreender que a rede neural mais conhecida e mais simples de entender é a rede neural de multicamadas *feed forward*. Ela tem uma camada de entrada, uma ou várias camadas ocultas, e uma única camada de saída. Cada camada pode ter um número diferente de neurônios e cada camada está totalmente conectada à camada adjacente. As conexões entre os neurônios em as camadas formam um gráfico acíclico, conforme ilustrado na Figura 3.

Figura 3 – Topologia de rede neural multicamadas.

Fonte: Adaptado de (PATTERSON; GIBSON, 2017).

2.3 Visão computacional

Segundo Elgendy (2020):

"A visão computacional é a ciência de perceber e compreender o mundo por meio de imagens e vídeos construindo um modelo físico do mundo para que um sistema de IA possa então tomar as ações apropriadas. A percepção visual é a busca visualmente por padrões em um objeto".

Os algoritmos de visão computacional são normalmente empregados como dispositivos de interpretação. O intérprete é o cérebro do sistema de visão. Sua função é obter a imagem de saída do dispositivo de detecção e aprender recursos e padrões para identificar objetos. Os cientistas se inspiraram em como nossos cérebros funcionam e tentaram projetar o sistema nervoso central para obter *insights* sobre como construir um cérebro. Assim, nasceram as redes neurais artificiais (RNAs) (ELGENDY, 2020).

2.4 Machine Learning

No aprendizado de máquina, em vez de ensinar a um computador uma lista enorme de regras para resolver o problema, dá-se a ele um modelo com o qual ele pode avaliar exemplos, e um pequeno conjunto de instruções para modificar o modelo quando comete um erro. Espera-se que com um modelo bem adequado, seria capaz de resolver o problema com extrema precisão (BUDUMA; LOCASCIO, 2017).

À medida que avança-se em problemas mais complexos, como reconhecimento de objetos e análise de texto, nossos dados tornam-se mais robustos, e as características se tornam não lineares entre si, para isso foi desenvolvido o *Deep Learning*, que teve sucesso na resolução de problemas de visão computacional (BUDUMA; LOCASCIO, 2017).

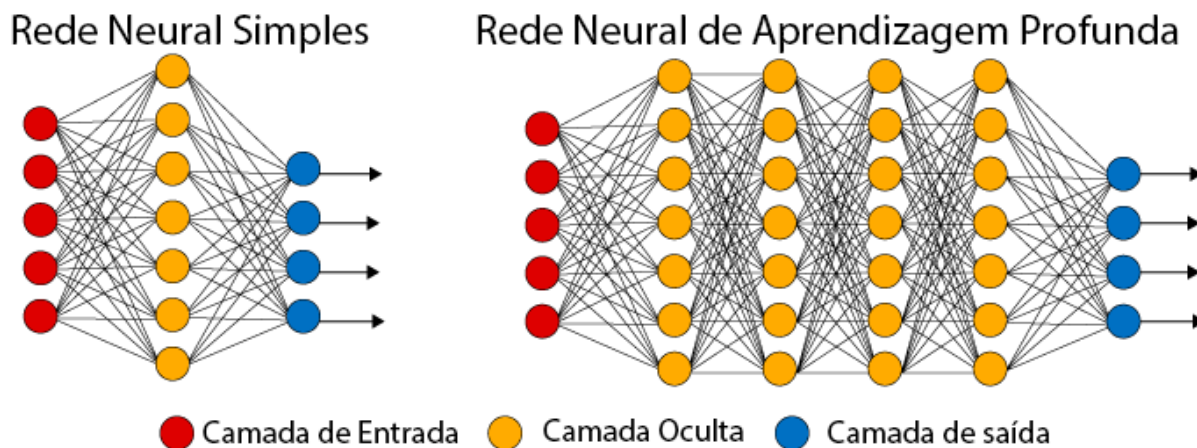
2.5 Deep Learning

Definido por Patterson e Gibson (2017):

"A aprendizagem profunda é um conceito de rede com mais neurônios, formando conexões mais complexas de camadas e com um uso maior do poder da computação para treinar e extrair automaticamente os recursos".

Na Figura 4, é ilustrada a diferença entre uma rede neural simples e uma rede neural profunda. As quatro principais arquiteturas de redes profundas são: redes pré-treinadas não supervisionadas, redes neurais convolucionais, redes neurais recorrentes e as redes neurais recursivas.

Figura 4 – Diferença entre *Deep Learning* e Redes Neurais simples.



Fonte: Adaptado de (KAUR et al., 2021).

2.6 Redes Neurais Convolucionais

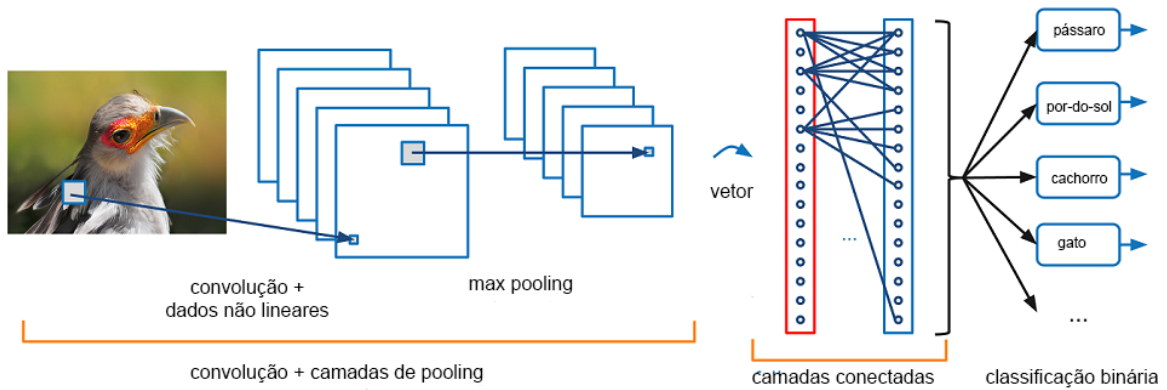
LeCun et al. (1998) desenvolveu um reconhecedor para dígitos chamado LeNet. A arquitetura apresentada foi futuramente então denominada de redes neurais convolucionais e consistia em um *backpropagation* em uma rede *feed-forward* com muitas camadas ocultas e mapas de unidades replicados em cada camada, agrupando as saídas de unidades próximas, se tornando uma rede que pode lidar com vários caracteres ao mesmo tempo.

O objetivo de uma CNN (*Convolutional Neural Network*) é aprender recursos de ordem superior nos dados por meio de convoluções. Elas são consideradas adequadas para o reconhecimento de objetos com imagens e classificação de imagem, podendo identificar rostos, indivíduos, placas de rua, e muitos outros aspectos dos dados visuais. CNNs também podem ser aplicadas a arquivos de áudio quando estes são representados visualmente como um espectrograma (PATTERSON; GIBSON, 2017).

As redes convolucionais percebem as imagens como objetos tridimensionais, ou seja, como volumes. As imagens digitais possuem codificações em cores e a rede convolucional recebe as imagens como estratos dessas cores empilhadas. Os canais são as camadas de profundidade que recebem a imagem como uma caixa retangular, cuja largura e altura são medidas pelo número de pixels das dimensões, a profundidade da caixa é de três camadas profundas (DATA SCIENCE ACADEMY, 2021).

As imagens vão se movendo por meio da rede convolucional, de camada em camada, gerando uma série de probabilidades de classes na camada de saída e a classe com maior probabilidade será definida para a imagem de entrada. Descreve-se em termos de volumes de entrada e saída, expressando-as matematicamente como matrizes de múltiplas dimensões (por exemplo, 30x30x3), como na Figura 5 (DATA SCIENCE ACADEMY, 2021).

Figura 5 – Como funciona uma Rede Neural Convolutional.

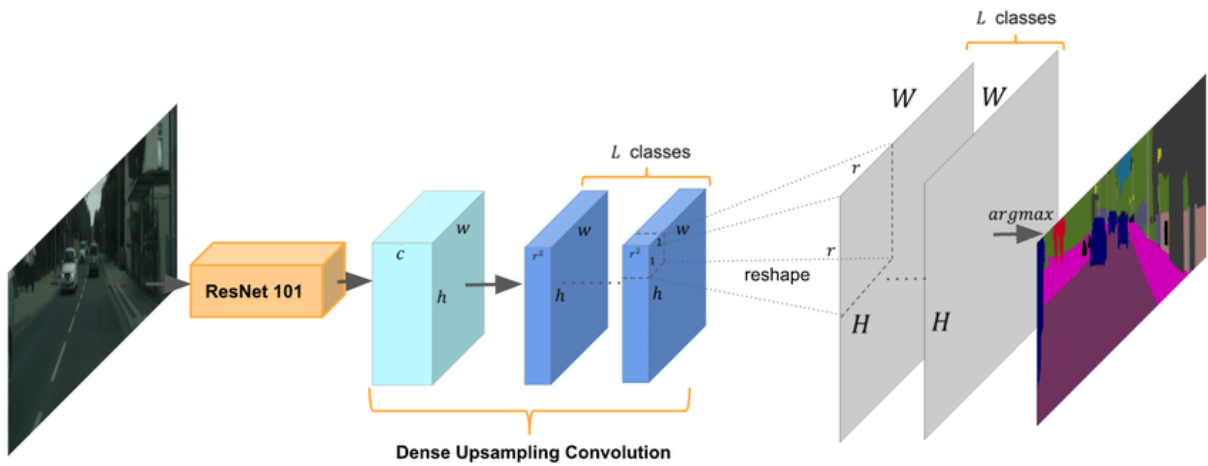


Fonte: Adaptado de (ADIT DESHPANDE, 2016).

2.6.1 Arquitetura das redes

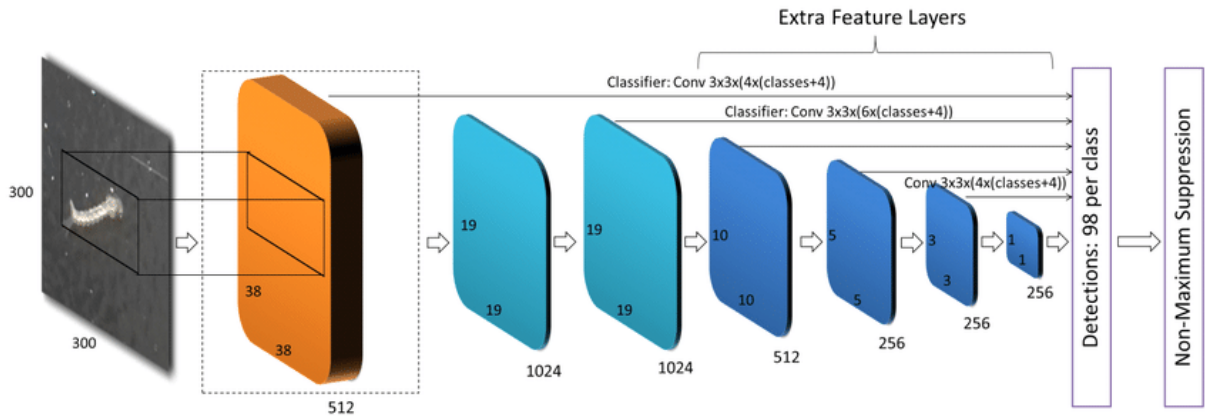
Neste trabalho, todas as redes pré treinadas utilizadas são redes neurais convolucionais. A estrutura das redes é composta por camadas de convolução, *pooling*, *dense block* e *transition*. Nas Figuras 6, 7 e 8 são apresentadas as estruturas das redes Resnet-101, SSD-Inception-V2 e YOLOv4 para visualização:

Figura 6 – Arquitetura da rede Resnet-101.



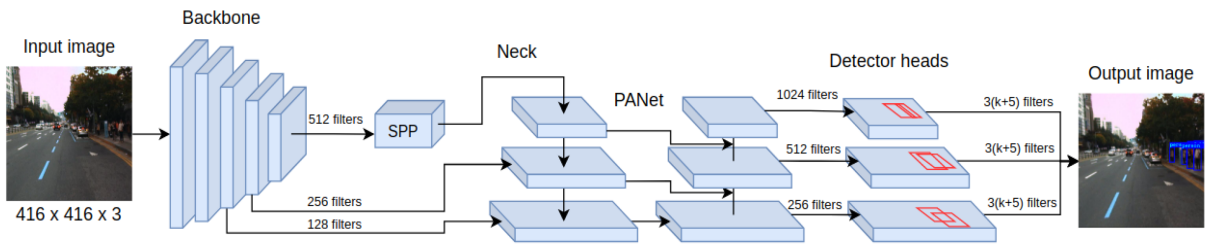
Fonte: (WANG et al., 2017).

Figura 7 – Arquitetura da rede SSD-Inception-V2



Fonte: (FUAD et al., 2019).

Figura 8 – Arquitetura da rede YOLOv4



Fonte: (ROSZYK; NOWICKI; SKRZYPCZYŃSKI, 2022).

2.7 Classificação, Detecção e Rastreamento de Objetos

As principais tarefas da Visão Computacional que estão sendo amplamente utilizadas atualmente são: classificação, detecção e localização de imagens (ELGENDY, 2020).

2.7.1 Classificação

A classificação de imagens é a tarefa que consiste em rotular uma imagem conforme o objeto presente na mesma. Geralmente, são retornados valores, expressos em forma de porcentagem, como resultado de acordo com o objeto presente na imagem. Exemplos de aplicação são: identificação de pessoas, animais, vegetais, dentre outros (SHANMUGAMANI, 2018).

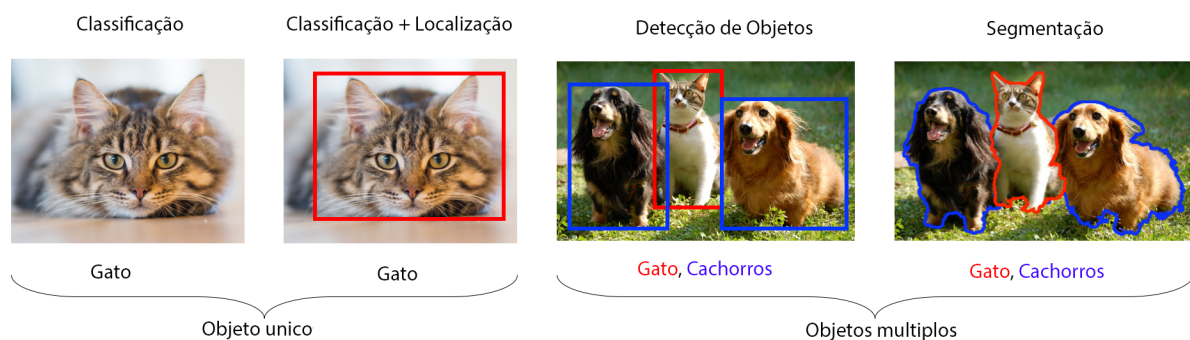
Problemas de classificação de imagens são os mais básicos para CNNs. Nesses problemas, cada imagem contém apenas um objeto, e nossa tarefa é identificá-lo. Para alcançar níveis humanos de compreensão, tem-se que adicionar complexidade a essas redes para que possa reconhecer vários objetos e suas localizações em uma imagem (ELGENDY, 2020).

2.7.2 Detecção de objetos

A detecção de objetos é semelhante a classificação, porém ela gera uma caixa delimitadora ao redor de um ou mais objetos presentes na imagem que indicam seu posicionamento. As principais aplicações são: veículos autônomos, contadores de objetos, diferenciação de classes e afins (SHANMUGAMANI, 2018) (CHOLLET, 2017).

Pode-se construir sistemas de detecção de objetos como YOLO (você só olha uma vez), SSD (detector de disparo único), e o Faster R-CNN, que não só classifica as imagens, mas também pode localizar e detectar cada objeto em imagens que contêm vários objetos. Esses sistemas de *Deep Learning* podem olhar para uma imagem, dividir em regiões menores e rotular cada região com uma classe, sendo assim possível localizar e rotular um objeto (ELGENDY, 2020). Exemplos de detecção e classificação estão na Figura 9 a seguir.

Figura 9 – Classificação e detecção de objetos.



Fonte: Adaptado de (SHANMUGAMANI, 2018).

2.7.3 Rastreamento de objetos

O rastreamento de objetos em imagens possibilita seguir objetos em vídeos no decorrer do tempo, sendo similar ao conceito de detecção de objetos. Para o rastreamento é necessário fornecer ao algoritmo a posição inicial do objeto a ser rastreado, de forma

manual ou automatizada, desta forma o objeto será rastreado no decorrer do tempo (SONKA; HLAVAC; BOYLE, 2014).

Na forma automatizada, utiliza-se a biblioteca dlib, (KING, 2009). (DANELLIAN et al., 2014) criaram o rastreador que funciona por meio de filtros correlacionais e amplamente utilizados nos trabalhos relacionados.

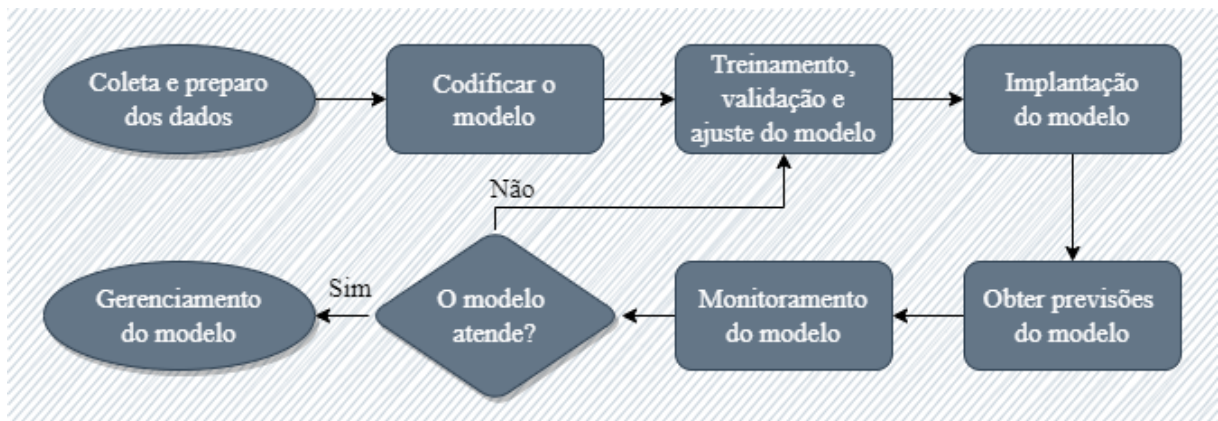
As principais aplicações de rastreamento de objetos são em rastreamento de veículos, pedestres e animais.

3 Metodologia

Este capítulo apresenta as ferramentas e técnicas utilizadas na pesquisa. São abordados: os equipamentos e as ferramentas, o *dataset* utilizado, o método de treinamento das redes, o *framework* desenvolvido e as métricas de validação utilizadas.

Este trabalho foi realizado seguindo o fluxo de trabalho de aprendizagem de máquina de Google (2019), apresentado na Figura 10.

Figura 10 – Fluxo de trabalho de *Machine Learning*.



Fonte: Adaptado de (GOOGLE, 2019)

Cada etapa do fluxo de trabalho é descrita a seguir:

- Coleta de dados: pesquisa e adequação dos dados de um *dataset* para a aplicação ao modelo;
- Codificar o modelo: estruturação do modelo com base nos requisitos necessários e utilizando técnicas estabelecidas de *Machine Learning*;
- Treinamento, validação e ajuste do modelo: o modelo é inicialmente ajustado com parâmetros iniciais de acordo com os dados e os requisitos prévios. O treinamento consiste em um processo de aprendizagem do modelo por meio da inserção dos dados coletados. A validação do modelo treinado é realizada em uma pequena porcentagem dos dados coletados com o objetivo de atestar se os resultados (acurácia) são coerentes;
- Implantação do modelo treinado: aplicação do modelo ao sistema desenvolvido;

- Obter predição do modelo: obtêm-se os resultados (predições) do modelo treinado após a aplicação no sistema;
- Monitoramento do modelo: monitoramento do modelo analisando os resultados obtidos e possíveis erros que possam ocorrer;
- Gerenciamento do modelo: versionamento do modelo.

3.1 Equipamentos e ferramentas

O desenvolvimento e execução do *framework*, assim como treinamento das redes, foram realizados em um notebook que possui uma CPU (Central Processing Unit) Intel core i5 8th Gen, 8 GB de memória RAM e SSD de 250 GB, com o sistema operacional Linux Ubuntu 18.04 LTS x64 instalado.

Para o treinamento das redes em nuvem utilizou-se a plataforma Google Colab da Google LLC, onde são disponibilizadas gratuitamente algumas horas para a aplicação de algoritmos de aprendizagem de máquina.

3.2 *Dataset*

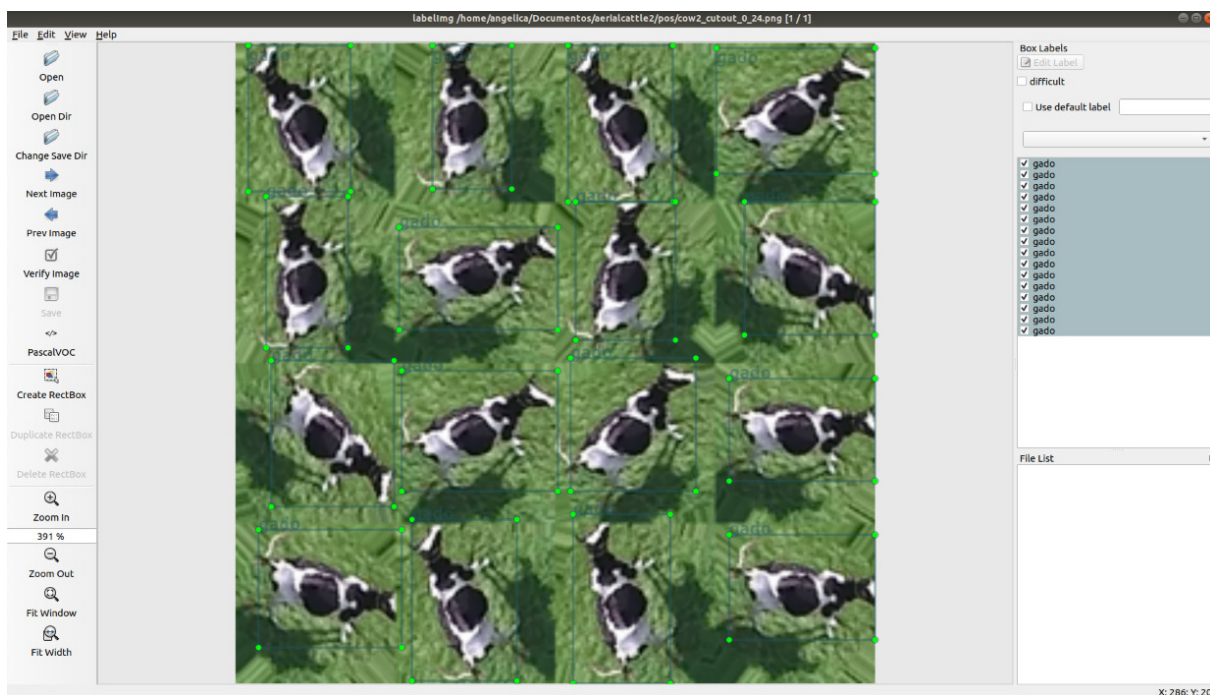
Para o desenvolvimento deste trabalho, o primeiro passo foi encontrar um *dataset* de imagens aéreas de gado disponível. Foi optado por utilizar o *dataset open-source* Aerial UAV Dataset Okafor et al. (2019), sendo este constituído apenas de fotos e vídeos, com *data augmentation* de rotação aplicado. Algumas das imagens do *dataset* utilizado estão na Figura 11:

Figura 11 – Exemplos de imagens presentes no *dataset*.

Fonte: Adaptado de (OKAFOR et al., 2019).

Desta forma foi necessário mapear as imagens e rotulá-las utilizando o programa LabelImg, como apresentado na Figura 12.

Figura 12 – Exemplo de rotulagem no programa labelImg.



Fonte: do autor

Os arquivos de saída gerados no programa LabelImg são em formato xml, tiveram que ser convertidos por meio do *script* presente no TensorFlow Object API, para o formato TFRecord, aceito no kit de ferramentas de visualização do TensorFlow, o TensorBoard.

3.3 Treinamento

As 524 imagens positivas, rotacionadas e de teste, presentes no *dataset* foram selecionadas e separadas aproximadamente na proporção de 70%, 20% e 10%, como usualmente é realizado nos projetos relacionados, resultando em 367 imagens para treinamento, 105 para validação e 52 para teste.

O treinamento das redes foi realizado na plataforma Google Colab utilizando o *framework* do TensorFlow e a Object Detection API, com as redes pré-treinadas Resnet-50 (*Residual Network*), Resnet-101 (*Residual Network*) e SSD-Inception-V2 (*Single Shot Detection*), disponibilizadas em Tensorflower Gardener (2021).

E também a rede pré-treinada YOLOv4, disponível em AlexeyAB (2021), que utiliza a biblioteca Darknet, semelhante ao TensorFlow utilizado nas outras redes. A escolha das redes pré-treinadas se baseou nas informações de precisão e velocidade disponibilizadas, e também por serem redes utilizadas nos trabalhos relacionados: Andrew, Greatwood e

Burghardt (2017), Barbedo et al. (2019), Shao et al. (2020), Múcher et al. (2022) e Porto et al. (2021).

Existem duas situações que queremos evitar em um aprendizado de máquina, que são:

- *Underfitting*: quando o modelo tem seus dados de treino subajustados e isso causa uma baixa taxa de acerto nos dados de treinamento e teste;
- *Overfitting*: quando o modelo foi tão ajustado aos dados de treinamento, que não consegue trabalhar com dados de fora desse conjunto, ou seja, possui uma baixa capacidade de generalização.

Para se evitar essas situações é importante um número adequado de dados e se interromper o treinamento no pico da acurácia. Os hiperparâmetros utilizados nas redes: Resnet-50, Resnet-101 e SSD-Inception-V2, foram os apresentados na tabela 1.

Tabela 1 – Hiperparâmetros utilizados nas redes Resnet-50, Resnet-101 e SSD-Inception-V2.

Hiperparâmetro	Valor
Numero de classes	1
Width	300
Height	300
Batches	64
Num. steps	5000
Learning rate	0,004

Fonte: Do autor.

A seguir na tabela 2 os dados dos hiperparâmetros utilizados para a rede YOLOv4.

Tabela 2 – Hiperparâmetros usados na rede YOLOv4.

Hiperparâmetro	Valor
Numero de classes	1
Width	416
Height	416
Batches	64
Burn-in	2000
Subdivisions	32
Num. steps	2000
Learning rate	0,004

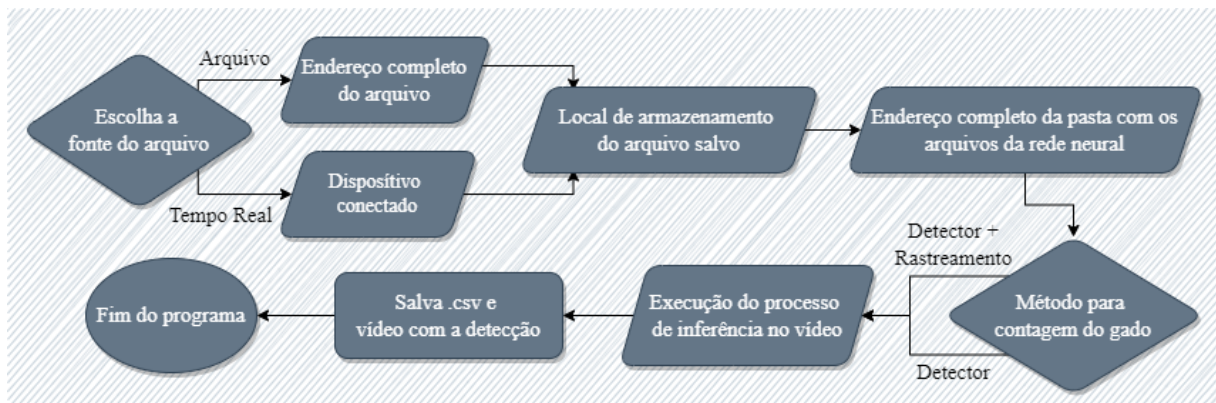
O uso destes hiperparâmetros foi baseado nas recomendações presentes nas documentações e projetos do TensorFlow e da Darknet em: Lyudmil Vladimirov (2020), Tensorflower Gardener (2021), AlexeyAB (2021) e Cloud Google (2022).

3.4 Framework

O *framework* foi desenvolvido utilizando o código *open-source* de Geraldini (2021) como base para o desenvolvimento do sistema de detecção de gado. Foi utilizada a linguagem Python e ele realiza a contagem atual e total de gado por meio de vídeos em tempo real ou gravados, visando uma futura implementação de conexão entre o sistema e um drone em tempo real. O *framework* gera como saída um arquivo .csv e um vídeo com a contagem total de bovinos. Foram utilizadas redes neurais convolucionais para detecção e um algoritmo de rastreamento auxiliar presente na biblioteca dlib, acompanhando o movimento dos bovinos no rebanho.

Na Figura 13 é apresentado um fluxograma simplificado do funcionamento do *framework*:

Figura 13 – Fluxograma simplificado do *framework*.



Fonte: do autor

O código completo do *framework* está anexado no Apêndice ??.

3.5 Métricas de validação

Foram aplicadas métricas de PASCAL VOC sobre os processos de detecção. Sendo essas:

- IoU - *Intersect Over Union*: descreve a extensão da sobreposição da área de duas caixas de rótulo.
- *Precision* (Precisão): é a capacidade de um modelo para evitar rotular amostras negativas como positivas.

- *Recall*: é a capacidade de um modelo para detectar todas as amostras positivas.
- AP - *Average Precision* (Precisão média): a média ponderada de precisões atingidas em cada limite, com o aumento no recolhimento do limite anterior usado como o peso.

A Darknet utilizada para o treinamento da rede YOLOv4 nos fornece os valores das métricas. Para as demais redes treinadas, o valor da métrica de *Average Precision* foi realizado pelo TensorBoard, kit de ferramentas de visualização do TensorFlow. Para calcular as demais métricas utiliza-se os valores da matriz de confusão, presente na tabela 3, calculada pelo *script* presente no Tensorflow Object API.

Tabela 3 – Matriz de confusão.

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

E as equações a seguir:

$$IoU = \frac{TP}{TP + FP + FN}, \quad (3.1)$$

$$Precision = \frac{TP}{TP + FP}, \quad (3.2)$$

$$Recall = \frac{TP}{TP + FN}. \quad (3.3)$$

Com o treinamento das redes e a comparação dos dados reais e previstos, calcula-se a perda (*loss*) total que consiste em duas partes, a perda de localização para a previsão de deslocamento da caixa delimitadora e a perda de classificação para probabilidades de classe condicional. A perda nos dá um parâmetro sobre o desempenho da rede, onde busca-se minimizá-la ao máximo.

Além disso, comparações foram realizadas entre a contagem automatizada do *framework* e a contagem visual, em vídeos previamente gravados.

4 Resultados e Discussão

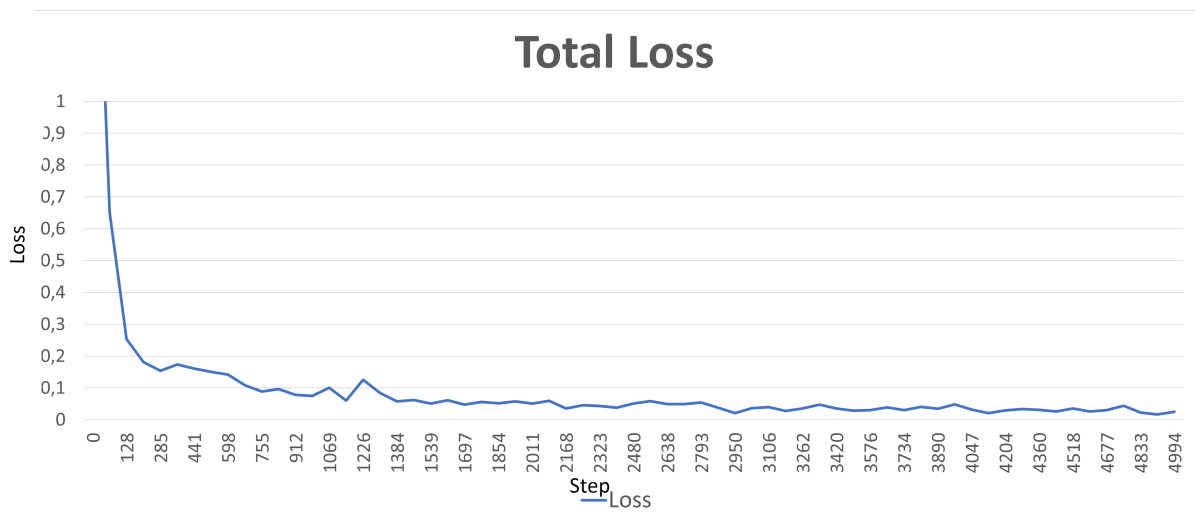
O treinamento das redes Resnet-50, Resnet-101 e SSD-Inception-V2 gerou arquivos de pesos que, utilizando a ferramenta TensorBoard do *framework* TensorFlow, foi possível obter os resultados das métricas PASCAL VOC: IoU (*Intersect Over Union*), *Precision*, *Recall* e AP (*Average Precision*). Para a rede YOLOv4 o *framework* da Darknet nos fornece esses dados. A Tabela 4 apresenta os resultados.

Tabela 4 – Métricas e seus valores resultantes do processo de treinamento.

Rede	Métrica				
	IoU	<i>Precision</i>	<i>Recall</i>	AP	
Resnet-50	90,36%	0,97	0,92	96,41%	
Resnet-101	89,93%	0,97	0,92	96,34%	
SSD-Inception-V2	76,73%	1	0,77	96,71%	
YOLOv4	77,94%	0,96	0,99	90,91%	

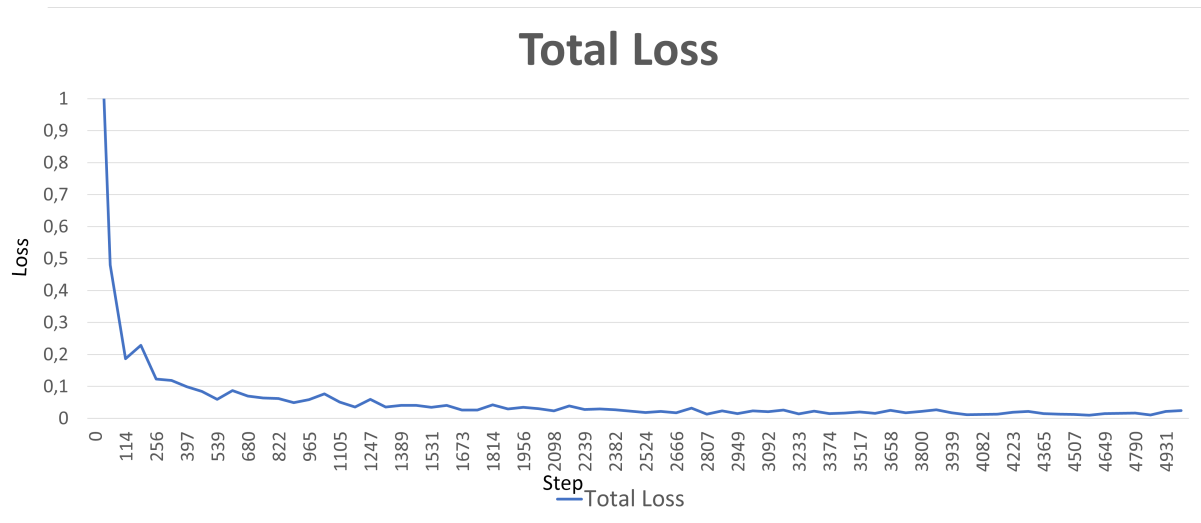
Nas Figuras de 14 a 22 estão apresentados os gráficos das perdas de classificação, localização e total dos treinamentos realizados nas redes Resnet-50, Resnet101 e SSD-Inception-V2:

Figura 14 – Resnet-50 *Total loss*.



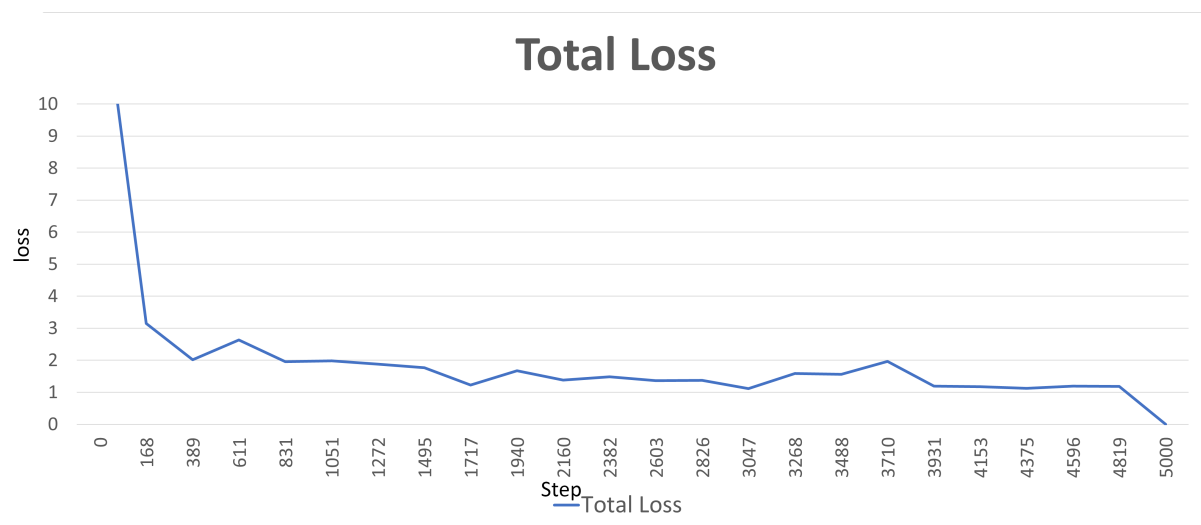
Fonte: do autor

Figura 15 – Resnet-101 *Total loss*.



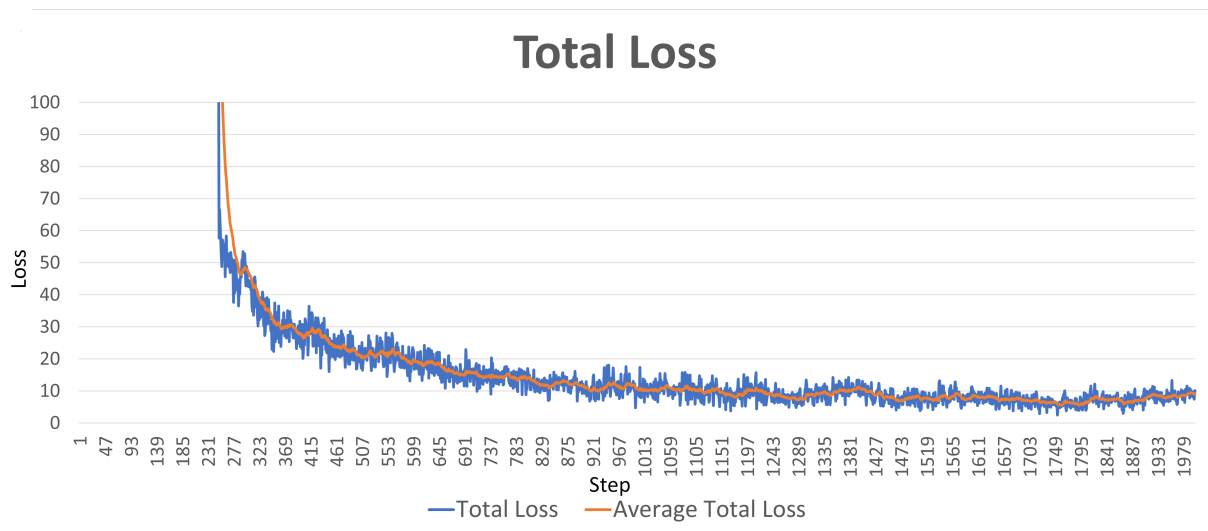
Fonte: do autor

Figura 16 – SSD-Inception-V2 *Total loss*.



Fonte: do autor

O treinamento da rede YOLOv4 gerou os gráficos de *Average total loss* e *Total loss* presentes na Figura 17.

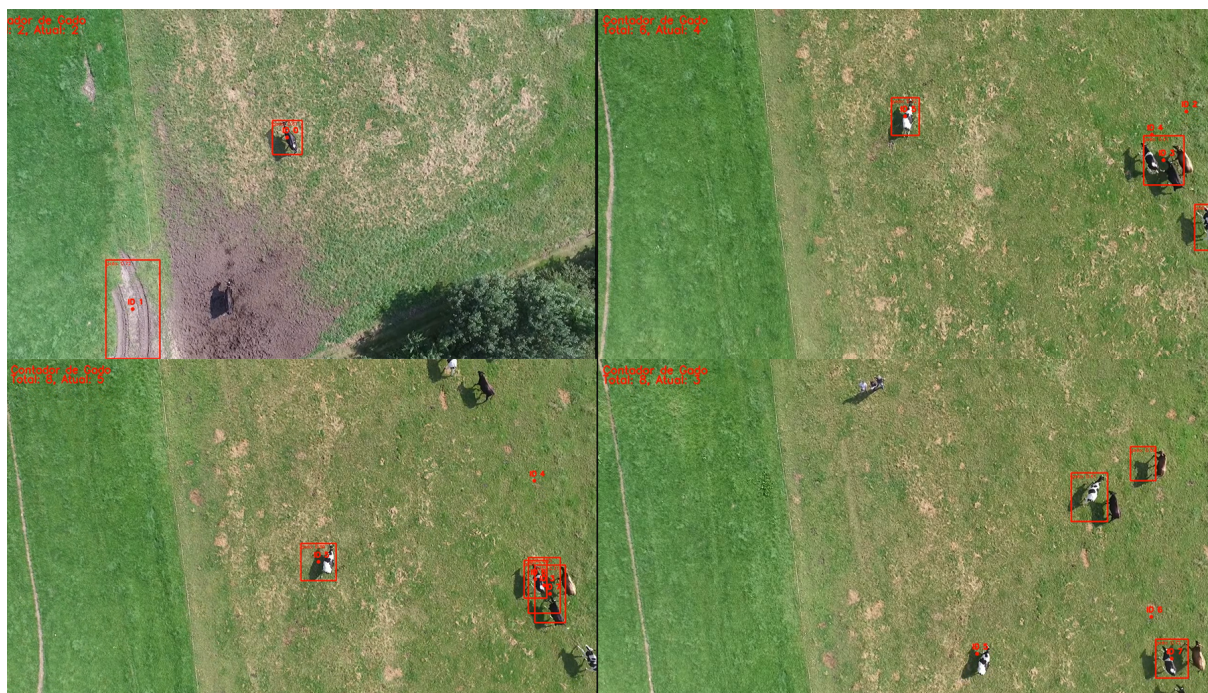
Figura 17 – YOLOv4 *Average total loss* e *Total loss*.

Fonte: do autor

Analisando o comportamento da perda nos gráficos notamos que todas as redes tiveram um treinamento satisfatório, porém os resultados da rede YOLOv4 ficaram aquém do esperado, com valores maiores de perdas e uma precisão menor, em comparação com as demais redes. Nota-se pelo comportamento da curva ao final do gráfico a presença de *overfitting*. O provável motivo é que a YOLO não é uma rede recomendada para detecção de objetos pequenos e/ou irregulares, necessitando de alterações em sua arquitetura e nos hiperparâmetros como recomendado em AlexeyAB (2021).

Apesar dos valores da rede SSD-Inception-V2 de *Recall* e *IoU* serem inferiores das redes Resnet-50 e Resnet-101, por conta da *Precision* e *Average Precision*, ela apresentou um desempenho melhor para o *framework*, sendo então a mesma aplicada.

A Figura 18 apresenta o teste do *framework* realizado em um dos vídeos presentes no Okafor et al. (2019) utilizando a rede SSD-Inception-V2. Nota-se na primeira imagem uma caixa delimitadora fora do objeto e que na última imagem os humanos que aparecem não são detectados.

Figura 18 – *Frames* do vídeo de saída do *framework*.

Fonte: do autor.

O *framework* funcionou como esperado, realizando a detecção do gado e com o rastreamento seguindo-o por alguns instantes, possibilitando a contagem mesmo em movimento. Nota-se, a necessidade de melhoria das redes e do *dataset* utilizados, visto o problema da caixa delimitadora em local errado.

5 Conclusão

O *framework* desenvolvido conseguiu realizar a contagem do gado dos vídeos de teste, contudo aperfeiçoamentos são necessários visto que o mesmo apresenta algumas falhas de detecção, como a caixa fora do gado a qual ela pertence. O treinamento da rede SSD-Inception-V2 apresentou como resultado uma *Average Precision* de 96,71%, sendo este o melhor resultado do trabalho e a rede aplicada ao *framework*.

A principal limitação encontrada durante a realização do trabalho foi não possuir um *hardware* mais robusto para realizar o treinamento de forma local. A utilização da nuvem para o processo de treinamento de forma gratuita é limitada em algumas poucas horas, gerando um inconveniente durante o desenvolvimento do trabalho.

O desenvolvimento de um *dataset* próprio e o treinamento de outras redes neurais convolucionais, ajuste diferente de hiperparâmetros podem melhorar o desempenho do processo de detecção. A principal contribuição do trabalho é o estudo e aplicação de redes neurais e o *framework* que está disponível no GitHub em: Manfrim A. C. (2022), para utilização em outros estudos e podendo receber atualizações futuramente.

5.1 Trabalhos futuros

Com base nas contribuições do trabalho e buscando a continuação da pesquisa, a seguir são listadas algumas possíveis contribuições futuras:

- Criação e aplicação de um *dataset* próprio ao treinamento e *framework* utilizados neste trabalho;
- Treinamento e implementação de outras redes neurais;
- Criação de uma rede neural própria e específica para o *framework*;
- Desenvolvimento da comunicação em tempo real das imagens de um drone com o dispositivo
- Desenvolvimento da aplicação do trabalho em dispositivos móveis;
- Implementação de interface gráfica no *framework*, tornando-o mais intuitivo;

Referências

- ADIT DESHPANDE. *A Beginner's Guide To Understanding Convolutional Neural Networks*. 2016. Disponível em: <<https://adeshpande3.github.io/A-Beginner27s-Guide-To-Understanding-Convolutional-Neural-Networks/>>.
- ALEXEYAB. *Yolo v4, v3 and v2 for Windows and Linux*. Yolo v4, v3 and v2 for Windows and Linux, 2021. Disponível em: <<https://github.com/AlexeyAB/darknet>>.
- ANDREW, W.; GREATWOOD, C.; BURGHARDT, T. Visual localisation and individual identification of holstein friesian cattle via deep learning. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*. [S.l.: s.n.], 2017.
- BARBEDO, J. G. A. et al. A study on the detection of cattle in uav images using deep learning. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 19, n. 24, p. 5436, 2019.
- BUDUMA, N.; LOCASCIO, N. *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*. [S.l.]: "O'Reilly Media, Inc.", 2017.
- CHOLLET, F. *Deep Learning with Python*. [S.l.]: Manning, 2017. ISBN: 978-1617294433.
- CLOUD GOOGLE. *Reference for built-in image object detection algorithm*. Reference for built-in image object detection algorithm, 2022. Disponível em: <<https://cloud.google.com/ai-platform/training/docs/algorithms/reference/object-detection>>.
- DANELLIAN, M. et al. Accurate scale estimation for robust visual tracking. In: *Proceedings of the British Machine Vision Conference*. [S.l.]: BMVA Press, 2014.
- DATA SCIENCE ACADEMY. *Deep Learning Book*. Deep Learning Book, 2021. Disponível em: <<https://www.deeplearningbook.com.br/>>.
- ELGENDY, M. *Deep Learning for Vision Systems*. [S.l.]: Simon and Schuster, 2020.
- FUAD, M. et al. Detection of aedes aegypti larvae using single shot multibox detector with transfer learning. *Bulletin of Electrical Engineering and Informatics*, v. 8, 06 2019.
- GERALDI, M. T. *Fish-CNN-Tracking*. [S.l.]: GitHub, 2021. <https://github.com/mtg1603/Fish-CNN-Tracking>.
- GOOGLE, L. *Google Cloud Documentação - Fluxo de trabalho de machine learning*. 2019. Disponível em: <<https://cloud.google.com/ai-platform/docs/ml-solutions-overview>>. Acesso em: 5 de setembro de 2019.
- HAYKIN, S. S. et al. *Neural networks and learning machines/Simon Haykin*. [S.l.]: New York: Prentice Hall, 2009.
- HERRERO, M. et al. The roles of livestock in developing countries. *Animal*, v. 7, p. 3–18, 2013. ISSN 1751-7311. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1751731112001954>>.

- IBGE. *Rebanho bovino reduz em 2018, em ano de crescimento do abate e exportação*. Agência IBGE (Instituto Brasileiro de Geografia e Estatística) notícias, 2020. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/25483-rebanho-bovino-reduz-em-2018-em-ano-de-crescimento-do-abate-e-exportacao>>.
- KAUR, J. et al. Machine learning techniques for 5g and beyond. *IEEE Access*, IEEE, v. 9, p. 23472–23488, 2021.
- KING, D. E. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, v. 10, p. 1755–1758, 2009. <<https://dl.acm.org/doi/10.5555/1577069.1755843>>.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998.
- LYUDMIL VLADIMIROV. *Training Custom Object Detector*. Training Custom Object Detector, 2020. Disponível em: <<https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/training.html>>.
- MANFRIM A. C. *Mestrado Detecta Gado*. mestradoDetectaGado, 2022. Disponível em: <<https://github.com/ManfrimAngelica/mestradoDetectaGado>>.
- MAPA. *Agropecuária Brasileira em Números*. Ministério da Agricultura, Pecuária e Abastecimento (MAPA), 2021. Disponível em: <<https://www.gov.br/agricultura/pt-br/assuntos/politica-agricola/todas-publicacoes-de-politica-agricola/agropecuaria-brasileira-em-numeros>>.
- MüCHER, C. A. et al. Detection, identification and posture recognition of cattle with satellites, aerial photography and uavs using deep learning techniques. *International Journal of Remote Sensing*, Taylor Francis, v. 43, n. 7, p. 2377–2392, 2022. Disponível em: <<https://doi.org/10.1080/01431161.2022.2051634>>.
- OKAFOR, E. et al. *Aerial UAV Dataset*. OSF, 2019. Disponível em: <osf.io/d6vrf>.
- PATTERSON, J.; GIBSON, A. *Deep learning: A practitioner's approach*. [S.l.]: O'Reilly Media, Inc., 2017.
- PORTO, J. V. et al. Automatic counting of cattle with faster r-cnn on uav images. In: *Anais do XVII Workshop de Visão Computacional*. Porto Alegre, RS, Brasil: SBC, 2021. p. 1–6. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/wvc/article/view/18880>>.
- RIVAS, A. et al. Detection of cattle using drones and convolutional neural networks. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 7, p. 2048, 2018.
- ROSZYK, K.; NOWICKI, M. R.; SKRZYPCZYŃSKI, P. Adopting the yolov4 architecture for low-latency multispectral pedestrian detection in autonomous driving. *Sensors*, v. 22, n. 3, 2022. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/22/3/1082>>.
- SAIZ-RUBIO, V.; ROVIRA-MAS, F. From smart farming towards agriculture 5.0: A review on crop data management. *Agronomy*, v. 10, n. 2, 2020. ISSN 2073-4395. Disponível em: <<https://www.mdpi.com/2073-4395/10/2/207>>.

- SCHMIDEK, A. S. *Boas práticas de manejo: identificação*. [S.l.: s.n.], 2009.
- SHANMUGAMANI, R. *Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras*. [S.l.]: Packt Publishing Ltd, 2018.
- SHAO, W. et al. Cattle detection and counting in uav images based on convolutional neural networks. *International Journal of Remote Sensing*, Taylor and Francis, v. 41, n. 1, p. 31–52, 2020. Disponível em: <<https://doi.org/10.1080/01431161.2019.1624858>>.
- SIDDHARTH, D.; SAINI, D. K.; KUMAR, A. Precision agriculture with technologies for smart farming towards agriculture 5.0. *Unmanned Aerial Vehicles for Internet of Things (IoT) Concepts, Techniques, and Applications*, Wiley Online Library, p. 247–276, 2021.
- SOARES, V. et al. Cattle counting in the wild with geolocated aerial images in large pasture areas. *Computers and Electronics in Agriculture*, v. 189, p. 106354, 2021. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169921003719>>.
- SONKA, M.; HLAVAC, V.; BOYLE, R. *Image processing, analysis, and machine vision*. [S.l.]: Cengage Learning, 2014.
- TENSORFLOW GARDENER. *TensorFlow 1 Detection Model Zoo*. TensorFlow 1 Detection Model Zoo, 2021. Disponível em: <https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md>.
- THE ASIMOV INSTITUTE. *Neural networks*. 2016. Disponível em: <<https://www.asimovinstitute.org/>>.
- WANG, P. et al. Understanding convolution for semantic segmentation. 02 2017.