



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Reconhecimento de bovinos em imagens utilizando Redes Neurais Convolucionais

Kallebe de Sousa Silva

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador

Prof. Dr. Eduardo Peixoto Fernandes da Silva

Brasília
2023

Dedicatória

Dedico este trabalho aos meus pais, fontes inesgotáveis de apoio e inspiração. Suas incansáveis jornadas, sacrifícios e dedicação moldaram não apenas quem sou como pessoa, mas também contribuíram fundamentalmente para a realização deste projeto. A cada desafio superado e a cada conquista alcançada, sinto a presença dos ensinamentos que vocês compartilharam ao longo da minha jornada. Este trabalho é, em grande parte, uma expressão da minha eterna gratidão e admiração por vocês. Muito obrigado por serem os melhores pais que alguém poderia desejar. Este é para vocês.

Agradecimentos

Agradeço a Deus, cuja graça me concedeu saúde e determinação, fundamentais para a conclusão deste trabalho.

Expresso minha profunda gratidão aos meus pais, cujos inúmeros sacrifícios proporcionaram a oportunidade de realizar este trabalho.

À minha irmã, meu sincero agradecimento por sempre estar ao meu lado, oferecendo o suporte necessário em todos os momentos.

Ao meu orientador, Eduardo Peixoto, manifesto minha sincera gratidão pela orientação dedicada e paciência, elementos essenciais para a conclusão deste trabalho.

A todos que compartilharam minha jornada acadêmica e acreditaram no meu sucesso, meu muito obrigado. Cada um de vocês desempenhou um papel valioso nessa conquista.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

A pecuária desempenha um papel significativo na economia brasileira, destacando-se como o maior rebanho bovino do mundo e líder mundial nas exportações de carne bovina. Diante da importância do setor, a implementação de novas tecnologias que suportem técnicas utilizadas, como a rastreabilidade do gado, é crucial para uma gestão mais eficiente. O reconhecimento de bovinos em imagens por meio de avanços em Inteligência Artificial, particularmente Redes Neurais Convolucionais (CNNs), surge como uma solução inovadora e não invasiva. Este estudo propõe avaliar o emprego de CNNs na identificação de bovinos utilizando um conjunto de imagens da raça Pantaneira. A pesquisa adota uma arquitetura baseada na DenseNet, incorporando técnicas de transferência de aprendizado para a classificação dos animais. Experimentos iterativos foram conduzidos, incluindo a seleção e recorte criteriosos do conjunto de dados, visando otimizar o desempenho da rede. A aplicação dessas abordagens permitiu alcançar resultados similares ao estado da arte. Espera-se que este trabalho não apenas contribua para a compreensão da eficácia das CNNs na raça Pantaneira, mas também sugira sua extensibilidade para análises em diversas raças, ampliando assim sua aplicabilidade e relevância.

Palavras-chave: Reconhecimento de bovinos, redes neurais convolucionais, rastreabilidade de gado

Abstract

Livestock plays a significant role in the Brazilian economy, standing out as the world's largest cattle herd and a global leader in beef exports. Given the importance of the sector, the implementation of new technologies that support established techniques, such as cattle traceability, is crucial for more efficient management. The recognition of cattle in images through advancements in Artificial Intelligence, particularly Convolutional Neural Networks (CNNs), emerges as an innovative and non-invasive solution. This study proposes to assess the use of CNNs in the identification of cattle using a set of images from the Pantaneira breed. The research adopts an architecture based on DenseNet, incorporating transfer learning techniques for animal classification. Iterative experiments were conducted, including careful selection and cropping of the dataset, aiming to optimize the network's performance. The application of these approaches allowed achieving results similar to the state of the art. This work is expected not only to contribute to understanding the effectiveness of CNNs in the Pantaneira breed but also to suggest their extensibility for analyses in various breeds, thereby expanding their applicability and relevance.

Keywords: Cattle recognition, convolutional neural networks, cattle traceability

Sumário

1	Introdução	1
2	Referencial Teórico	3
2.1	Aprendizado de máquina	3
2.1.1	Tipos de Aprendizado	4
2.2	Redes Neurais Artificiais	4
2.3	Redes Neurais Convolucionais	5
2.3.1	Convolução	6
2.3.2	Função de Ativação	8
2.3.3	<i>Pooling</i>	9
2.3.4	<i>Flattening</i>	10
2.3.5	Camada Totalmente Conectada	11
2.3.6	Arquiteturas de Rede	11
2.3.7	Conjuntos de dados	13
2.3.8	Treinamento	16
2.3.9	Métricas de desempenho	18
2.4	Conclusão	20
3	Revisão de Literatura	21
3.1	Contextualização	21
3.2	Evolução do campo de Estudo	22
3.3	Tipos de Rastreio	23
3.4	Trabalhos correlatos	25
4	Metodologia	28
4.1	Implementação	28
4.1.1	Arquitetura	29
4.1.2	Validação cruzada	30
4.2	Conjunto de dados	30
4.3	Pré-processamento das imagens	32

4.3.1	Aumento de dados	33
4.4	Experimentos	34
5	Resultados	37
5.1	Resultados	37
5.1.1	Experimento E1	37
5.1.2	Experimento E2	41
5.1.3	Experimento E3	43
6	Conclusão	47
	Referências	49

Lista de Figuras

2.1	Camadas presentes nas Redes Neurais Artificiais.	5
2.2	Exemplo de uma operação de convolução	6
2.3	Gráfico da função de ativação ReLU.	9
2.4	Exemplo de operação de Max Pooling	10
2.5	Arquitetura DenseNet	12
2.6	Validação cruzada <i>k-fold</i> com $k = 5$	16
2.7	Rótulos usados em matrizes de confusão para classes Positiva e Negativa.	19
3.1	Resultados obtidos por Weber <i>et al.</i>	27
4.1	Detalhamento da arquitetura DenseNet201, aplicada ao modelo de reconhecimento de bovinos proposto.	30
4.2	Distribuição de imagens no conjunto de dados original.	31
4.3	Problemas identificados no conjunto de dados utilizado	32
4.4	Bovino presente nas classes 49 e 51 do <i>dataset</i>	33
4.5	Distribuição de imagens no conjunto de dados selecionado.	34
4.6	Exemplos de imagens recortadas do conjunto de dados.	35
5.1	Curvas de acurácia e função de perda para a primeira implementação do Experimento E1a.	39
5.2	Métricas de treinamento e validação em cada época do Experimento E1b.	40
5.3	Matriz de confusão gerada para Experimento E1b.	41
5.4	Métricas de treinamento e validação em cada época do Experimento E2.	42
5.5	Métricas de treinamento e validação em cada época do Experimento E3.	43
5.6	Matriz de confusão gerada para Experimento E3.	44
5.7	Métricas de precisão alcançadas por cada classe no Experimento E3.	45
5.8	Métricas de <i>recall</i> alcançadas por cada classe no Experimento E3.	45
5.9	Medidas F1 alcançadas por cada classe no Experimento E3.	46

Lista de Tabelas

3.1	Parâmetros utilizados por Weber <i>et al.</i>	25
3.2	Resultados obtidos por Weber <i>et al.</i>	26
4.1	Detalhes da Arquitetura utilizada	29
4.2	Descrição dos experimentos propostos	36
5.1	Descrição dos experimentos propostos	38
5.2	Principais hiperparâmetros da 1 ^a implementação do Experimento E1a . . .	38
5.3	Arquitetura utilizada para a primeira implementação do Experimento E1a	40
5.4	Acurácia e função de perda do Experimento E1b no conjunto de teste . . .	40
5.5	Acurácia e função de perda do Experimento E2 no conjunto de teste	42
5.6	Acurácia e função de perda do Experimento E3 no conjunto de teste	43

Lista de Abreviaturas e Siglas

CNN *Convolutional Neural Network* (Rede Neural Convolucional).

FC *Fully Connected* (Totalmente Conectada).

FN Falsos Negativos.

FP Falsos Positivos.

IBGE Instituto Brasileiro de Geografia e Estatística.

ML *Machine Learning* (Aprendizado de Máquina).

RFID *Radio Frequency Identification* (Identificação por radiofrequência).

RNA Rede Neural Artificial.

SGD *Stochastic Gradient Descendent* (Gradiente Descendente Estocástico).

SISBOV Sistema Brasileiro de Identificação Individual de Bovinos e Búfalos.

VN Verdadeiros Negativos.

VP Verdadeiros Positivos.

Capítulo 1

Introdução

A pecuária é um dos setores de grande destaque na economia brasileira. No Brasil está presente o maior rebanho bovino do mundo, com 234,4 milhões de bovinos em 2022, segundo dados do Instituto Brasileiro de Geografia e Estatística (IBGE) [1]. Ainda, no ano de 2022 o Brasil foi responsável por 27,7% das exportações de carne bovina mundiais, liderando como maior exportador a nível internacional [2].

Tendo em vista o tamanho do setor pecuário e sua importância para a economia brasileira, torna-se imprescindível o suporte aos pecuaristas a partir da aplicação de novas tecnologias que contribuam para a gestão do gado. Uma das necessidades observadas no setor pecuário é a de rastreabilidade do gado, que possibilita a manutenção de registros relativos ao boi desde seu nascimento até seu abate. Tais registros garantem melhor segurança ao consumidor e ao próprio produtor, permitindo maior facilidade de acesso a informações relevantes de cada cabeça de gado [3].

A prática de rastreamento de animais bovinos se faz necessária no cenário pecuário brasileiro a fim de assegurar a segurança alimentar, a qualidade dos produtos de origem animal além de uma gestão eficiente da cadeia produtiva. O Brasil adota diversas abordagens de rastreio, buscando a identificação individual e o monitoramento ao longo do ciclo de vida dos bovinos. As opções incluem o uso de brincos auriculares e *bottons*, com números de manejo SISBOV, dispositivos eletrônicos para identificação, tatuagens, e marcação a ferro quente na perna traseira. Além dessas opções, há a utilização de dispositivos eletrônicos, como chips RFID, que proporcionam um rastreamento mais automatizado e preciso, embora possam ser menos acessíveis. Essas variações destacam a flexibilidade e robustez do sistema de identificação pecuário, incorporando a métodos tradicionais e tecnologias avançadas [4].

Nesse sentido, métodos baratos, rápidos e seguros de identificação animal são cruciais para a viabilidade de sistemas de gestão do setor pecuário. Com o avanço tecnológico, as aplicações envolvendo Inteligência Artificial tem crescido de forma acelerada nos úl-

timos anos. Aplicações como identificação de tumores em imagens médicas [5], carros auto-dirigíveis [6], assistentes virtuais [7], dentre várias outras, já estão em uso e estão cada vez mais presentes no cotidiano das pessoas [8]. Nesse contexto, a aplicação dessas novas tecnologias surgem como métodos interessantes para uma revolução tecnológica no setor pecuário. A utilização de técnicas avançadas, como as Redes Neurais Convolucionais, no reconhecimento de bovinos, revela um potencial inovativo na gestão do gado, apresentando uma solução não invasiva para a identificação do gado, além de facilitar processos zootécnicos, atividades de controle e prestação de contas para a produção. A implementação de técnicas modernas da rastreabilidade na produção de gado pode levar a uma maior qualidade e produtividade, possibilitando o acesso a mercados nacionais e internacionais competitivos [9].

O presente estudo visa a avaliação do emprego de Redes Neurais Convolucionais (CNNs) no reconhecimento de bovinos em imagens, apresentando experimentos destinados a essa finalidade. Para tal, foram empregadas imagens da raça Pantaneira, motivadas pela limitada disponibilidade de conjuntos de dados específicos para bovinos. No Capítulo 2 serão apresentados conhecimentos necessários para a compreensão da metodologia abordada. Dentre os conceitos abordados, destacam-se o funcionamento das Redes Neurais Convolucionais, técnicas de transferência de aprendizado e de particionamento do conjunto de dados e principais métricas utilizadas na avaliação de modelos de CNN.

O Capítulo 3 delineará a revisão de literatura, abrangendo estudos correlatos ao tema central desta monografia. Em seguida, o Capítulo 4 detalhará a metodologia de implementação escolhida para a solução de reconhecimento de bovinos, incluindo uma análise do conjunto de dados utilizado na pesquisa. O Capítulo 5 apresentará os resultados obtidos na implementação da solução desenvolvida, seguidos por uma análise aprofundada destes. Por fim, no Capítulo 6 será apresentada a conclusão desta pesquisa, consolidando os principais *insights* derivados dos resultados obtidos nos experimentos propostos.

Espera-se que os resultados obtidos nesse estudo desempenhem o papel de um indicativo, oferecendo uma perspectiva inicial sobre a aplicabilidade desse modelo em reconhecimento de bovinos em imagens. Este trabalho não apenas contribui para o entendimento da eficácia das CNNs na raça Pantaneira, mas também sugere uma potencial extensibilidade para a análise de diversas raças, ampliando assim a aplicabilidade e relevância dos resultados alcançados.

Capítulo 2

Referencial Teórico

Este capítulo tem como objetivo fornecer uma base sólida para a compreensão de conceitos relevantes aplicados durante o experimento. Será realizada uma breve introdução aos principais conceitos de aprendizado de máquina e, em seguida, será abordada uma base para o entendimento do funcionamento das CNNs, com ênfase na arquitetura DenseNet201 [10].

2.1 Aprendizado de máquina

Aprendizado de máquina, também conhecido pelo termo em inglês *Machine Learning* (ML) é um assunto que tem sido amplamente discutido nos últimos anos. Com diversos avanços e aplicações, é atualmente uma área que está em constante crescimento, atraindo cada vez mais a atenção de pesquisadores e desenvolvedores, que estudam e desenvolvem aplicações nos mais diversos setores.

Muito tem se discutido sobre a preferência de, para certas aplicações, treinar sistemas para a solução de problemas a partir da apresentação de exemplos de entrada e saída esperadas, em contraposição à programação de um sistema que antecipa respostas para todas as entradas possíveis [11]. Nesse contexto, notáveis avanços têm sido observados em aplicações como reconhecimento facial [12], detecção de objetos [13] e reconhecimento de fala [14] onde o uso de aprendizado de máquina tem gerado resultados promissores.

Para tais aplicações, portanto, são interessantes algoritmos que sejam adaptáveis a mudanças. Em outras palavras, é necessário que o algoritmo "aprenda" a fornecer soluções satisfatórias ao problema em questão. Em cenários como este, destaca-se a relevância do uso de *Machine Learning*.

2.1.1 Tipos de Aprendizado

Os principais tipos de aprendizado de algoritmos de ML são o Aprendizado Supervisionado e o Aprendizado Não Supervisionado [15].

No Aprendizado Supervisionado, são utilizados dados rotulados no treinamento do modelo. Assim, cada dado de exemplo possui um rótulo relacionado indicando a saída esperada. Retomando o exemplo de reconhecimento facial, os dados para treinamento do modelo conterão uma certa quantidade de exemplos para cada indivíduo. Cada imagem possui um rótulo indicando a pessoa presente na imagem. O modelo, então, buscará extrair as principais características que ajudam a identificar o indivíduo da imagem. Posteriormente, quando apresentado uma imagem não rotulada, o modelo realizará a predição com base no que foi aprendido durante o treinamento.

Diferentemente do Aprendizado Supervisionado, o Aprendizado Não Supervisionado utiliza dados não rotulados durante o treinamento. Neste paradigma, o modelo busca reconhecer padrões e agrupar dados com características semelhantes. No exemplo de reconhecimento facial, seriam enviadas ao modelo várias imagens dos indivíduos, sem identificação da pessoa que está presente em cada imagem. O modelo procuraria por padrões e similaridades nas imagens para agrupá-los com base nessas características.

2.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA), ou simplesmente Redes Neurais, são um subconjunto do amplo campo de *Machine Learning* que, em um nível fundamental, são projetadas para simular o funcionamento do cérebro humano, buscando imitar a forma que os neurônios interagem e processam informações [16].

O modelo das RNAs é estruturado a partir de unidades básicas de processamento denominadas de neurônios artificiais, ou apenas "nós", organizadas em camadas, como apresentado na Figura 2.1. Cada nó recebe uma entrada proveniente dos nós da camada anterior, realiza um processamento específico e transmite uma saída para nós da camada seguinte. Dessa maneira, cada camada é composta por uma variedade de nós que operam em paralelo, permitindo um processamento eficiente e complexo de informações.

A primeira camada de uma RNA é designada camada de entrada. Nessa camada são inseridos os dados iniciais que serão processados ao longo da rede. A última camada da rede, por sua vez, é denominada camada de saída, onde os resultados e predições da rede neural são gerados. Entre essas duas extremidades, encontram-se as chamadas camadas ocultas, que são responsáveis por todo o processamento interno dos dados. Uma RNA pode possuir várias camadas ocultas. A quantidade de camadas presente na rede irá

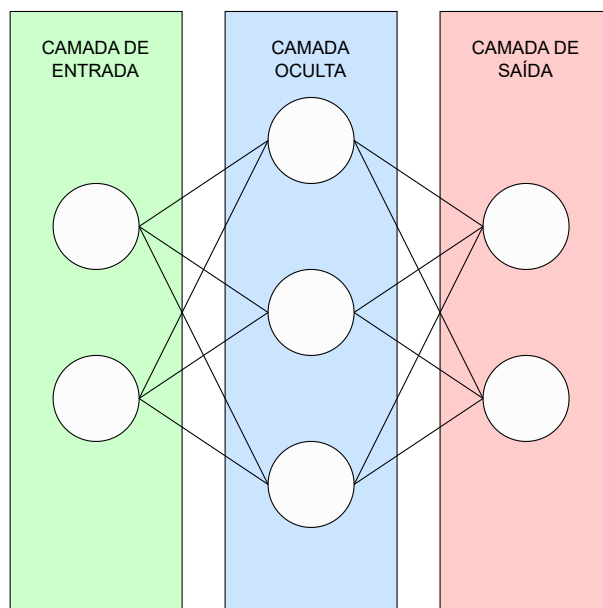


Figura 2.1: Camadas presentes nas Redes Neurais Artificiais.

definir a sua profundidade. É a partir desse conceito que são conhecidas as Redes de Aprendizado Profundo, ou, do inglês, *Deep Learning* [15].

2.3 Redes Neurais Convolucionais

Redes Neurais Convolucionais (CNNs) são uma especialização das RNAs que utilizam dados em formato de matrizes [15], sendo elas 1D como podem ser representados dados de áudio, por exemplo, ou 2D, como imagens.

As CNNs possuem esse nome por realizar uma operação de convolução em suas camadas. Nessas camadas de convolução, o objetivo é que a rede consiga aprender os pesos usados nos filtros de convolução a fim de extrair características relevantes das imagens.

Uma característica das imagens explorada pelas CNNs é o fato de os pixels vizinhos possuírem um maior nível de correlação e apresentarem características básicas, como bordas e curvas [16]. Dessa forma, as primeiras camadas de uma CNN podem aprender a detectar essas bordas, por exemplo. As camadas seguintes irão aprender características cada vez mais complexas, iniciando por combinações de bordas, tais como arcos, cantos e curvas, chegando até detecção de partes do objeto, como os olhos, se deseja-se reconhecer faces, cores, e outras características mais complexas.

Em uma camada convolucional típica, estão presentes três estágios de processamento [15]. No primeiro estágio, são realizadas operações de convolução nos dados de entrada da camada. No segundo estágio, os resultados obtidos da convolução passam por uma

função de ativação não-linear. Por fim, no último estágio, os resultados da camada anterior passam por uma função de *pooling*. Esses estágios serão abordados com mais detalhes nas próximas seções.

2.3.1 Convolução

A convolução desempenha um papel crucial no funcionamento das CNNs, permitindo a detecção de padrões locais e o aprendizado automático de características relevantes para a tarefa em questão. Embora a operação empregada em redes profundas seja, na verdade, uma correlação cruzada (*cross-correlation*), o termo "convolução", amplamente utilizado no contexto de redes neurais [15], será adotado neste trabalho.

A convolução é uma operação linear que envolve a aplicação repetida de um filtro, também conhecido como *kernel*, sobre os dados de entrada, resultando na geração de um mapa de características. Matematicamente, a operação de convolução consiste em multiplicar o *kernel* por uma fatia dos dados de entrada e somar os resultados. Este é um procedimento no qual é aplicado um filtro, repetidamente, aos dados de entrada, resultando na geração de um mapa de características (*features*). O *kernel*, tipicamente de dimensões menores, representa padrões locais a serem identificados. Do ponto de vista matemático, a operação envolve a multiplicação do *kernel* por uma fatia dos dados de entrada. (Figura 2.2).

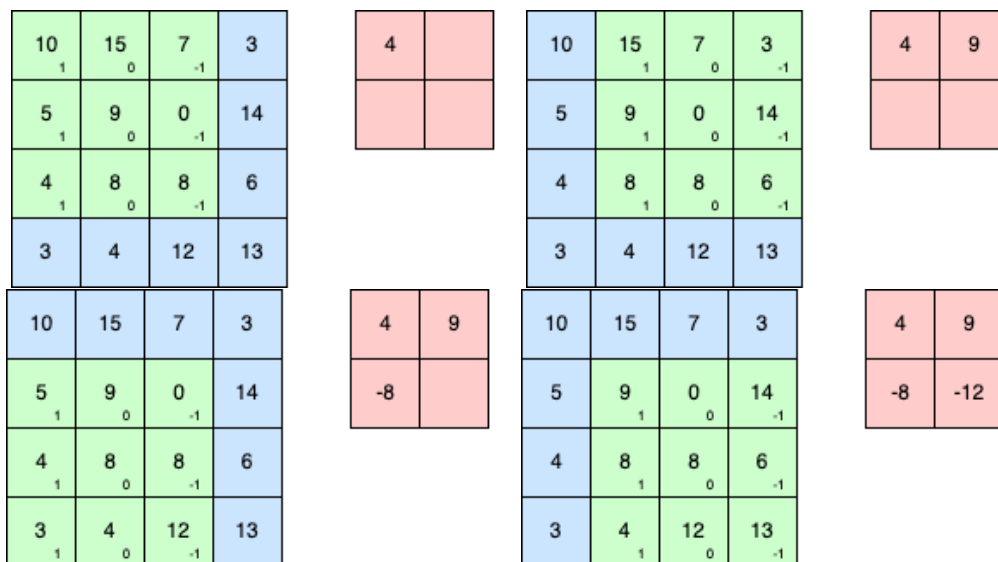


Figura 2.2: Operação de convolução sobre uma matriz com os dados de entrada representados em azul. O *kernel*, representado em verde, com os valores nos cantos inferiores de cada célula, percorre como uma janela deslizante pelos dados de entrada. Para cada iteração, os valores do *kernel* são multiplicados, elemento a elemento, aos valores da entrada, e depois somados. O resultado é armazenado no mapa de características, representado em vermelho.

Diferentemente de filtros em outros contextos, como processamento de imagens, os valores do *kernel* em redes profundas são aprendidos durante o treinamento. O *kernel* representa um padrão local ou um atributo aprendido, assemelhando-se aos filtros de borda em imagens, linhas ou padrões mais intrincados. Quando a característica representada pelo *kernel* está presente em uma região dos dados de entrada, o resultado da operação indica um valor maior do que em regiões onde nada foi detectado.

A utilização da operação resulta em um mapa de resposta ou mapa de características (*feature map*), geralmente de dimensões inferiores às dos dados originais, e serve como entrada para as camadas subsequentes da rede. Duas características fundamentais das redes baseadas em convolução são destacadas. Primeiramente, os padrões aprendidos são independentes de localização, permitindo a detecção do padrão em qualquer posição da imagem. Isso implica na invariância de translação dos padrões aprendidos. Em segundo lugar, os padrões aprendidos nas camadas iniciais são utilizados como elementos para a composição de padrões mais complexos nas camadas subsequentes, formando assim uma hierarquia de padrões [17].

Ao longo do treinamento, a rede automaticamente ajusta os valores do *kernel* para otimizar a detecção dos elementos desejados, aprimorando assim a capacidade da rede em reconhecer padrões específicos. Essa capacidade de aprendizado é fundamental para o sucesso das redes neurais em diversas aplicações.

Como observado na Figura 2.2, a operação de convolução 2D reduz a dimensionalidade do mapa de características. Muitas vezes, esta é uma consequência esperada pelo projetista da rede. Porém, é possível manter a dimensão dos dados, ou ainda ajustá-la conforme a necessidade. Para isso, são usados dois parâmetros de ajuste: o *padding* e o *stride*.

Padding

Na técnica de *padding*, são adicionados pixels ao redor da borda da imagem antes da aplicação da convolução. Os pixels adicionados podem ser chamados de pixels de preenchimento. Geralmente, atribui-se o valor zero a esses pixels de preenchimento, mas isso pode variar dependendo da aplicação.

Stride

Na operação convolucional convencional, a janela de convolução inicia no canto superior esquerdo e é deslocada, elemento por elemento, para a direita e para baixo, conforme ilustrado na Figura 2.2. No entanto, caso seja necessário uma redução na dimensionalidade do mapa de características, ou ainda uma redução na carga computacional da rede, pode-se recorrer à aplicação de uma técnica conhecida como "*stride*" [18].

Nessa abordagem, um valor é definido para indicar a quantidade de passos que a janela de convolução dará a cada iteração. Ao estabelecer o *stride* $s = 1$, a operação de convolução ocorrerá de maneira idêntica à convolução usual. Ao utilizar valores de *stride* mais altos, a sobreposição entre as janelas de convolução diminui, resultando em uma redução mais pronunciada na dimensão do mapa de características.

2.3.2 Função de Ativação

Tendo passado pelo estágio de convolução, o mapa de características gerado passa por uma função não-linear de ativação. A introdução de não-linearidade é fundamental para que a rede seja capaz de aprender padrões mais complexos e não-lineares.

As funções de ativação mais amplamente reconhecidas na construção de redes neurais são a sigmoide e a tangente hiperbólica. No entanto, é importante notar que muitas funções de ativação podem ocasionar o problema do gradiente próximo de zero em redes profundas. Esse fenômeno ocorre devido à multiplicação do gradiente ao longo de várias camadas durante o processo de retropropagação (*backpropagation*) [17], que será abordado na Seção 2.3.8. Consequentemente, em redes neurais profundas, várias outras funções de ativação são empregadas para contornar esse desafio, sendo a *Rectified Linear Unit* (ReLU) uma das mais proeminentes.

ReLU

Existem diversas funções de ativação que atualmente são usadas nas CNNs. Uma que se destaca por sua simplicidade e eficiência computacional é a *Rectified Linear Unit* (ReLU), sendo amplamente utilizada em CNNs e em RNAs em geral. A função ReLU é comumente empregada após a operação de convolução. Isso resulta na preservação apenas dos valores positivos. Essa abordagem tem se mostrado eficiente em mitigar o problema do gradiente desvanecente, permitindo um treinamento mais eficaz de redes neurais profundas. Além disso, a ReLU não impõe restrições aos valores positivos, o que significa que não há saturação quando x é positivo. A ReLU é frequentemente aplicada em camadas escondidas após uma camada de convolução.

Existem diversas alternativas à função ReLU, como Leaky ReLU, GeLU, entre outras, todas projetadas para abordar alguma limitação da ReLU. No entanto, mesmo com essas opções, a ReLU continua sendo amplamente adotada na maioria das arquiteturas de redes profundas.

Matematicamente, a ReLU é representada pela Equação 2.1, possuindo um gráfico como o representado na Figura 2.3.

$$f(x) = \max(0, x) \quad (2.1)$$

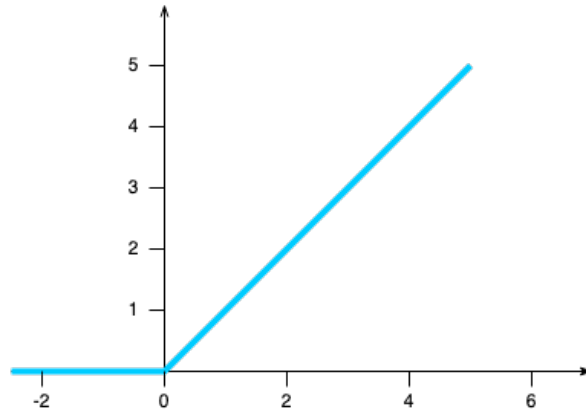


Figura 2.3: Gráfico da função de ativação ReLU.

Softmax

Outra função de ativação amplamente empregada em redes neurais é a função Softmax. Sendo mais comumente usada na camada de saída da RNA, especialmente prevalente em sistemas de classificação, a softmax recebe um vetor de entrada x e realiza uma transformação que resulta em uma distribuição de probabilidades. Essa distribuição atribui probabilidades a cada classe possível, indicando a confiança ou probabilidade associada a cada classe específica. Em contexto de classificação, a softmax é fundamental para converter as saídas da rede em uma interpretação probabilística, facilitando a identificação da classe mais provável para uma dada instância de entrada. Matematicamente, a função softmax é representada pela Equação 2.2.

$$f(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}} \quad (2.2)$$

2.3.3 Pooling

Após passar pela camada convolucional, o mapa de características pode passar por uma camada de *pooling*, que basicamente substitui a saída da rede em determinado local com uma estatística das saídas vizinhas [15]. Esta função traz uma certa invariabilidade para pequenas translações e rotações na imagem, ao mesmo tempo em que pode reduzir a dimensionalidade do mapa de características, preservando suas principais informações e aliviando a carga computacional.

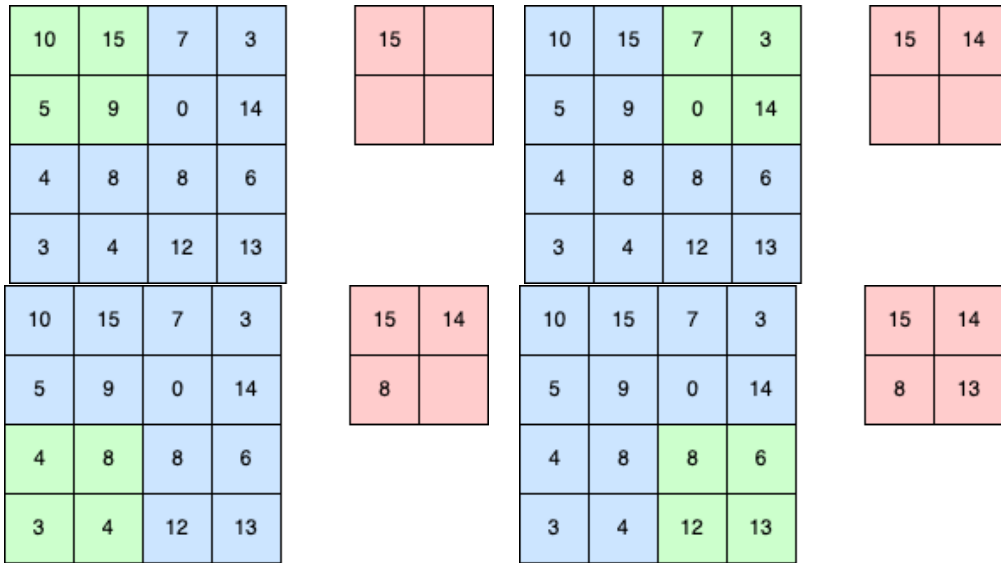


Figura 2.4: Operação de *Max Pooling* com *stride* $s = 2$ sobre um mapa de características representados em azul. A região 2×2 de *pooling*, representada em verde, desloca de dois em dois elementos a cada iteração, devido ao valor do *stride*. Para cada iteração, calcula-se o valor máximo dentro da região de *pooling*, armazenando o resultado subamostrado em vermelho.

Uma das técnicas amplamente adotadas é o *Max Pooling* (Figura 2.4). Nessa abordagem, a ideia central é reter o valor máximo de uma região do mapa de características, descartando os demais valores dessa região. Essa estratégia assegura que os valores mais proeminentes sejam preservados.

Um segundo método de *pooling* é o *Average Pooling*. Em contraste com o *Max Pooling*, essa técnica realiza uma média dos valores em determinada região do mapa de características, proporcionando uma abordagem mais suavizada, levando em consideração a contribuição média das características presentes na região.

2.3.4 *Flattening*

Após passar por camadas convolucionais e de subamostragem, alcança-se como resultado uma matriz multidimensional, composta por diversas camadas de mapas de características aprendidas ao longo da rede. Contudo, para a aplicação de operações densamente conectadas, tais como as presentes em redes neurais tradicionais, é imperativo que a entrada seja um vetor unidimensional [19].

Para realizar essa transição, emprega-se uma camada de *Flattening*, responsável por efetuar um processo de "achatamento" na matriz multidimensional de características, transformando-a em um vetor unidimensional. Esse vetor resultante pode, então, ser

direcionado para uma ou mais camadas densamente conectadas, possibilitando que a rede prossiga na aprendizagem de padrões complexos e abstratos.

O processo de achatamento, na prática, é bastante direto. Em cada posição na matriz de características, a camada de *Flattening* simplesmente empilha os valores encontrados, criando um único vetor. Esse vetor ainda possui todas as informações obtidas pelas camadas anteriores, mas, agora, em um formato adequado para ser processado por camadas totalmente conectadas.

2.3.5 Camada Totalmente Conectada

As camadas totalmente conectadas, também conhecidas como camadas densamente conectadas ou pelo termo em inglês, *Fully Connected* (FC), geralmente compõem as últimas camadas das CNNs, incluindo a camada de saída. Nas camadas totalmente conectadas, cada neurônio está conectado a todos os neurônios da camada anterior e aos da próxima camada, formando uma densa rede de conexões.

As FCs permitem que a rede aprenda padrões mais abstratos e complexos. Elas são acompanhadas por funções de ativação, que permitem que essa arquitetura capture relações não lineares entre as características extraídas. Na camada de saída, é comum utilizar a função softmax (Seção 2.3.2), retornando a probabilidade de cada classe na saída.

2.3.6 Arquiteturas de Rede

A combinação de camadas na formação de arquiteturas de redes neurais é um aspecto fundamental no desenvolvimento de modelos de aprendizado profundo. A arquitetura de uma rede neural refere-se à organização e interconexão das camadas que compõem a rede. Cada camada desempenha um papel específico no processamento e na extração de informações dos dados de entrada [20].

Uma arquitetura de rede neural começa com uma camada de entrada, que recebe os dados brutos. Seu tamanho é determinado pela dimensionalidade dos dados de entrada. Essa camada é seguida por uma série de camadas ocultas, que são responsáveis por aprender representações hierárquicas e abstratas dos dados. A combinação de várias camadas ocultas permite que a rede capture características complexas e não lineares nos dados, tornando-a mais capaz de lidar com problemas desafiadores. Por fim a camada de saída, que produz os resultados finais da rede neural. O número de neurônios nesta camada depende do tipo de tarefa, como classificação (um neurônio por classe) ou regressão (um neurônio para a saída contínua) [20].

A combinação de camadas convolucionais é especialmente eficaz em tarefas de visão computacional. Camadas convolucionais empregam filtros locais nos dados de entrada, o que possibilita a identificação de padrões espaciais e hierarquias de características, como é explicado na Seção 2.3.1 deste texto. Normalmente, camadas de *pooling* são inseridas para diminuir a dimensionalidade dos dados e manter características essenciais [21].

Também é de suma importância a escolha das funções de ativação em cada camada, a depender principalmente da utilização, a mais utilizada, normalmente, é a função ReLU (*Rectified Linear Unit*). A técnica de regularização é frequentemente aplicada em camadas específicas para evitar o *overfitting*, conceito que será abordado na Seção 2.3.7. *Dropout*, por exemplo, é uma abordagem comum em que aleatoriamente desativamos neurônios durante o treinamento, forçando a rede a aprender representações mais robustas e generalizáveis [21].

A última camada da rede é a camada de saída, cuja configuração depende da natureza da tarefa. Para problemas de classificação, por exemplo, é comum usar uma camada de saída com função de ativação softmax. O treinamento da rede neural é alcançado por meio de algoritmos de otimização, como o gradiente descendente. O ajuste dos pesos das conexões entre as camadas é realizado durante o processo de treinamento, onde a rede é exposta a dados rotulados e ajusta seus parâmetros para minimizar uma função custo. É válido mencionar sobre a LeNet [13], que é uma arquitetura de rede neural convolucional desenvolvida por Yann LeCun, Léon Bottou, Yoshua Bengio e Patrick Haffner. Esta arquitetura foi proposta em 1998 e é uma das primeiras CNNs projetadas para reconhecimento de padrões em imagens, especialmente para tarefas de reconhecimento de dígitos manuscritos. Ela é composta por camadas convolucionais e de *pooling*, seguidas por camadas totalmente conectadas. Nesta monografia, entretanto, é utilizada a DenseNet201 [10], como modelo base de arquitetura de rede.

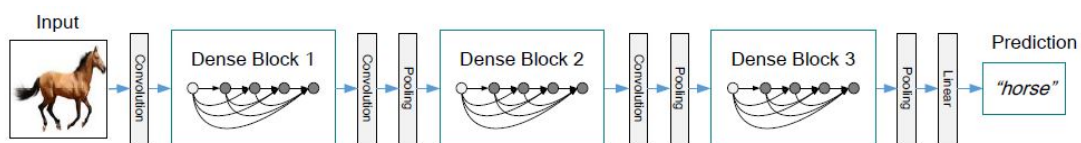


Figura 2.5: Arquitetura DenseNet com três blocos densos. Dentro de cada bloco, cada camada convolucional está conectada a todas as outras camadas, dentro do bloco. Camadas entre blocos são denominadas camadas de transição. Figura retirada de [10]

A DenseNet representa uma abordagem diferenciada na arquitetura de redes neurais convolucionais (Figura 2.5), introduzindo conectividade densa para promover a propagação eficiente de informações e reutilização de características como classificação de imagens, detecção de objetos e segmentação semântica, demonstrando um desempenho competitivo em comparação com outras arquiteturas populares, como ResNet e Inception. Por

ser eficaz em termos de desempenho e eficiência de parâmetros, é utilizada em diversas aplicações de aprendizado profundo [10].

Os Dense Blocks, ou blocos densos, são componentes específicos da arquitetura da DenseNet, usados em redes neurais convolucionais (CNNs). A ideia central por trás dos Dense Blocks é promover uma maior conectividade entre as camadas da rede, permitindo um fluxo mais eficiente de informações durante o treinamento. Em uma arquitetura de DenseNet, cada camada recebe como entrada as saídas de todas as camadas anteriores no bloco. Isso contrasta com abordagens tradicionais, onde as camadas recebem apenas as saídas das camadas imediatamente anteriores [10].

Das vantagens associadas aos blocos densos, destacam-se a promoção da Reutilização de Recursos, onde, ao receber as saídas de todas as camadas anteriores, cada camada em um Dense Block tem acesso a uma rica variedade de características aprendidas. Soma-se a isso a redução de parâmetros da rede, o que reduz a carga computacional durante seu treinamento. Outra vantagem é a mitigação do desvanecimento de gradiente através da conectividade densa, pois o sinal de retropropagação adquire acesso a diversas rotas para alcançar camadas anteriores [10].

Uma variante específica da arquitetura DenseNet é a DenseNet201, conhecida por sua estrutura densamente conectada. O número "201" em seu nome é devido ao total de camadas na rede. Ela é composta por várias camadas, incluindo blocos densos, blocos de transição e camadas totalmente conectadas. Os blocos densos são a característica distintiva da DenseNet e são compostos por várias camadas de convolução em que cada camada recebe entradas de todas as camadas anteriores no bloco. Isso cria uma arquitetura altamente conectada [10].

A DenseNet201 começa com uma camada de convolução inicial seguida por quatro blocos densos. Cada bloco denso contém várias camadas de convolução em sequência, com conexões densas entre elas. Entre os blocos densos, existem camadas de transição que reduzem a dimensionalidade espacial. A última parte da rede consiste em camadas totalmente conectadas que levam à camada de saída, onde as predições são feitas para tarefas específicas, como classificação de imagem. A função de ativação usada geralmente é ReLU, exceto na camada de saída, onde pode ser uma função como softmax para problemas de classificação [10].

2.3.7 Conjuntos de dados

Antes de se falar do treinamento das redes neurais, vale ressaltar sobre o conjunto de dados que será utilizado, também chamado de *dataset*. Na Seção 2.1.1 foram apresentados dois principais tipos de aprendizado em redes neurais: o aprendizado supervisionado e o aprendizado não-supervisionado. Para cada tipo de aprendizado, tem-se um *dataset*

configurado diferentemente. Para redes de aprendizado não-supervisionado, os dados presentes no *dataset* não estão rotulados. Em redes de aprendizado supervisionado, contudo, cada elemento do *dataset* está relacionado a um rótulo, representando a classe que esse elemento pertence.

No entanto, deve-se ter cuidado ao selecionar os dados pertencentes ao *dataset*. A qualidade dos dados afeta o desempenho do modelo tanto quanto a quantidade de dados. Dessa forma, quanto maior for a quantidade de dados presentes no *dataset*, e quanto melhor esses dados representarem todo o universo de possibilidades, melhor será o desempenho do modelo na predição de dados desconhecidos [8].

Os dados presentes no *dataset* geralmente são separados em três conjuntos: conjunto de treinamento, de validação, e de teste [18]. Não há uma norma quanto a proporção de cada subconjunto de dados, contudo. Uma divisão comum é separar 70% dos dados para treinamento, 20% para validação, e os 10% restantes para teste.

Os dados de treinamento são utilizados para o treinamento, de fato, da rede neural. Durante esse treinamento, a função custo será calculada e os pesos da rede serão atualizados. Esse processo de treinamento será melhor abordado na Seção 2.3.8. Após a atualização dos pesos da rede, o desempenho do modelo é avaliado utilizando o conjunto de validação. Durante essa etapa, é possível calcular métricas como a acurácia do modelo com os dados de validação, avaliando o desempenho do modelo em dados não apresentados à rede durante o treinamento. Após o treinamento completo da rede, pode-se avaliar o desempenho final do modelo a partir do conjunto de teste.

Overfitting e Underfitting

Overfitting e *Underfitting* estão relacionados à capacidade de um modelo de generalizar a partir de dados de treinamento para novos dados, mas representam extremos opostos desse espectro. *Overfitting* ocorre quando um modelo se ajusta excessivamente aos dados pertencentes ao conjunto de treinamento. Em outras palavras, o modelo "aprende" os detalhes específicos dos dados de treinamento, incluindo o ruído, em vez de capturar os padrões subjacentes que se aplicariam a novos dados. Isso geralmente resulta em um modelo que funciona bem nos dados de treinamento, mas tem um desempenho deficiente em dados não vistos. Os sinais típicos de *overfitting* incluem um desempenho excepcional durante o treinamento, mas uma queda significativa na precisão quando confrontado com novos dados [22].

Modelos altamente flexíveis, como redes neurais profundas, podem se tornar suscetíveis ao *overfitting* se não houver mecanismos de regularização adequados. Estratégias para lidar com o *overfitting* incluem a redução da complexidade do modelo, a aplicação de

técnicas de regularização (como o *dropout* em redes neurais) e o aumento do conjunto de dados de treinamento [23].

Underfitting, por outro lado, ocorre quando um modelo é muito simples para capturar os padrões nos dados de treinamento. Em vez de se ajustar adequadamente aos dados, o modelo subajusta e não consegue capturar as nuances ou relações importantes. Isso resulta em um desempenho insatisfatório tanto nos dados de treinamento quanto nos dados de teste. Os sinais de *underfitting* incluem um desempenho pobre durante o treinamento e uma melhoria insignificante, mesmo quando mais dados são adicionados. A causa subjacente pode ser uma escolha inadequada do modelo ou uma falta de capacidade para lidar com a complexidade inerente aos dados.

Para mitigar o *underfitting*, é necessário aumentar a complexidade do modelo. Isso pode envolver a seleção de modelos mais sofisticados, o aumento do número de recursos ou a aplicação de técnicas de pré-processamento mais avançadas nos dados [22]. O desafio na construção de modelos eficazes reside em encontrar o equilíbrio adequado entre *overfitting* e *underfitting*. Isso muitas vezes é alcançado através da validação cruzada e ajuste de hiperparâmetros. A validação cruzada permite avaliar o desempenho do modelo em conjuntos de dados não vistos, enquanto o ajuste de hiperparâmetros envolve a seleção adequada destes para encontrar a configuração ideal do modelo que equilibra a capacidade de generalização e a complexidade.

Validação Cruzada

O método de validação cruzada é amplamente utilizado em redes neurais e consiste, basicamente, na separação do *dataset* em um conjunto de dados utilizado para treinamento e um conjunto separado para avaliar o desempenho do modelo, chamado de conjunto de validação [18].

Uma abordagem bastante utilizada de validação cruzada, devido à sua simplicidade é o *hold-out*. Nesse método, o conjunto de dados é separado, aleatoriamente, em 3 partições disjuntas de treinamento, validação e teste. Durante o treinamento, portanto, o modelo é treinado utilizando este conjunto e avaliado pelo conjunto de validação [24]. Ao final do treinamento, o modelo é avaliado, finalmente, pelo conjunto de teste, ainda não apresentado durante o processo de treinamento. Embora essa abordagem seja barata computacionalmente e simples de se implementar, possui a dificuldade de determinar acuradamente o desempenho do modelo, já que sua avaliação irá depender da escolha inicial dos conjuntos de treinamento, validação e teste, podendo conter vieses que aumentam a variância dos dados [24].

Para corrigir a deficiência do método *hold-out*, há outra abordagem para particionamento do conjunto de dados denominada validação cruzada *k-fold* [25]. Nessa abordagem,

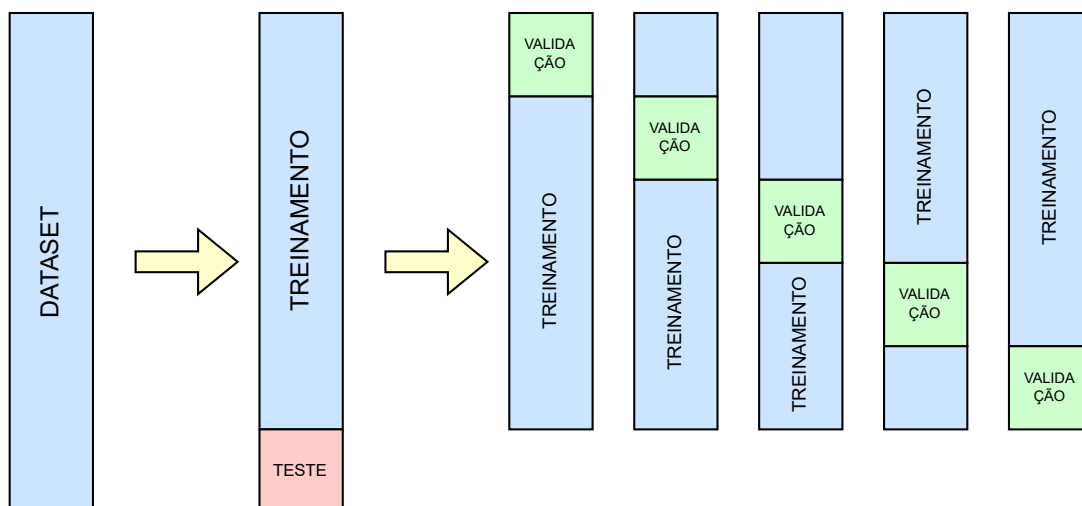


Figura 2.6: Validação cruzada k -fold com $k = 5$.

o *dataset* é dividido em k partições de mesmo tamanho. $k - 1$ partições irão compor o conjunto de treinamento da rede, enquanto a partição restante será utilizada como conjunto de validação (Figura 2.6). Na validação cruzada k -fold, esse processo de separação de conjuntos e treinamento do modelo é repetido por k vezes, onde a cada iteração é utilizada uma diferente partição para o conjunto de validação, e as demais para o conjunto de treinamento [24]. Ao final das k iterações, é calculada uma média das métricas de desempenho a fim de fornecer uma estimativa mais robusta do desempenho do modelo.

2.3.8 Treinamento

Tendo sido definida a arquitetura da CNN, já é possível iniciar o processo de treinamento da rede. Como visto na Seção 2.1.1, o treinamento da rede difere, dependendo se os dados são rotulados ou não. No presente trabalho, será considerado apenas o treinamento supervisionado, que é de maior relevância para o que foi desenvolvido.

No treinamento da CNN, busca-se aprender os parâmetros de rede. Esses parâmetros são os pesos e vieses dos filtros utilizados nas camadas convolucionais e das camadas totalmente conectadas. O termo "parâmetro" não deve ser confundido com outro termo utilizado no contexto de redes neurais, o "hiperparâmetro". Os hiperparâmetros são variáveis que devem ser definidas antes do início do treinamento e, diferentemente dos parâmetros de rede, não sofrem alterações durante o treinamento [18]. Exemplos de hiperparâmetros são o tamanho e número de *kernels* usados nas camadas convolucionais, *stride*, *padding* e funções de ativação.

No processo de treinamento, busca-se minimizar a diferença entre a predição do modelo e a saída esperada, obtida através dos rótulos dos dados do *dataset* utilizado. Para isso, é

utilizado um algoritmo de retropropagação, com o cálculo de gradientes sendo propagados da saída à entrada da rede [13]. Esse algoritmo opera associado ao conceito de funções de custo e gradiente descendente.

Função de Custo

A função de custo, também conhecida como função de perda, é responsável por calcular a diferença entre as saídas preditivas da rede neural e a saída esperada. Para problemas de classificação multi-classe, é comum usar a função de custo de entropia cruzada [18].

Na entropia cruzada, é retornada a diferença entre a distribuição de probabilidade da saída da rede, representando a confiança que a rede neural possui em cada classe, e a distribuição de probabilidade que se esperava [26]. Dessa forma, é calculada essa distância entre a distribuição de probabilidade obtida pela função de ativação Softmax, presente na camada de saída, e a distribuição de probabilidade que representa a saída esperada.

A função de perda, representada matematicamente por $J(\theta)$, deve, portanto, possuir o menor valor possível, para que a distância das distribuições de probabilidade esperada e obtida seja minimizada. Para isso é utilizado o algoritmo de Gradiente Descendente.

Gradiente Descendente

O Gradiente descendente é um algoritmo de otimização amplamente utilizado em redes neurais que é responsável pela atualização dos pesos da rede, com o objetivo de minimizar a função de custo. No algoritmo, é utilizado o gradiente da função de perda que indica a direção de crescimento da função. Nesta etapa, é utilizado um dos principais hiperparâmetros da rede chamado de taxa de aprendizado [18]. Tendo a direção do crescimento da função de perda, os pesos da rede serão atualizados na direção a um mínimo local da função [27]. A taxa de aprendizado, portanto, é utilizada para limitar o "passo" que será dado a esse mínimo local. Uma boa escolha é essencial, pois uma taxa de aprendizado muito alta, pode fazer com que os pesos sejam atualizados além do mínimo local, dificultando sua convergência. Por outro lado, uma taxa de aprendizado com valor muito baixo fará com que a rede demore muito até chegar a um mínimo.

O custo computacional do algoritmo, contudo, é um importante fator a ser considerado. Os *datasets* para treinamento das CNNs geralmente possuem milhares de dados, fazendo com que o cálculo do gradiente para esses milhares de dados seja altamente custoso computacionalmente. Para contornar esse problema, os gradientes da função de perda são calculados utilizando amostras do *dataset* chamados de *mini-batches* ou lotes. O tamanho do lote, portanto, é outro hiperparâmetro da rede, sendo que o algoritmo com um lote de tamanho unitário recebe o nome especial Gradiente Descendente Estocástico (SGD) [18].

Um outro hiperparâmetro fundamental para o contexto de treinamento do modelo é o número de épocas. As épocas irão definir quantas vezes o algoritmo irá passar por todo o conjunto de dados durante o treinamento [28]. Uma prática comum é a de gerar um gráfico apresentando o valor da função de custo para cada época, durante o treinamento e validação do modelo.

Transferência de Aprendizado

Uma importante etapa antes do treinamento do modelo de rede neural é a inicialização dos pesos que serão aprendidos. Possíveis abordagens podem incluir a inicialização dos pesos com zero, ou ainda com valores gerados aleatoriamente. Inicializações arbitrárias, contudo, podem atrasar, ou ainda atrapalhar o processo de convergência [29].

Para evitar que o modelo precise aprender características do zero, é comum a prática de utilizar a transferência de aprendizado. Nessa abordagem, é treinado um modelo em um domínio de origem e, então, esse modelo é modificado para realizar previsões em um diferente domínio [30].

Na transferência de aprendizado, após o treinamento do modelo no domínio de origem, é necessária a modificação do número de neurônios presentes na camada de saída, para que a rede se ajuste à quantidade de classes para o domínio que se deseja treinar o modelo. Esses pesos da camada de saída são, então, reinicializados aleatoriamente, mantendo-se os demais [31].

Tem se tornado uma prática comum na comunidade de visão computacional o uso de modelos pré-treinados com o *dataset* da ImageNet [32], com o fim de realizar transferência de aprendizado para o treinamento de modelos mais específicos [33].

2.3.9 Métricas de desempenho

No desenvolvimento de uma rede neural, a avaliação de desempenho dos modelos se tornam fundamentais para a verificação do bom funcionamento da rede e sua adaptação aos dados pertinentes ao problema. Métricas de desempenho em redes neurais, portanto, são indicadores quantitativos que avaliam a eficácia e precisão do modelo.

Matriz de confusão

Para modelos de classificação, é possível gerar uma tabela que indica os erros e acertos realizados por um modelo, chamada de Matriz de Confusão, a qual fornece uma visão mais detalhada do desempenho da rede. Nela, estão presente os seguintes rótulos: os verdadeiros positivos (VP), representando as previsões que o modelo classificou corretamente o elemento como pertencendo à classe desejada; os verdadeiros negativos (VN),

representando os exemplos em que o modelo classificou corretamente como o elemento não pertencendo à classe; os falsos positivos (FP), sendo os dados que o modelo classificou como pertencendo à classe, de maneira errônea; e os falsos negativos (FN), quando o modelo classificou, erroneamente, o dado como não pertencendo à classe desejada [34]. A Figura 2.7 ilustra esse conceito de matriz de confusão.

		Valor Predito	
		Positivo	Negativo
Valor Real	Positivo	VP	FN
	Negativo	FP	VN

Figura 2.7: Rótulos usados em matrizes de confusão para classes Positiva e Negativa.

Acurácia

Uma métrica comumente utilizada para a avaliação de modelos é a acurácia. Ela representa a proporção de instâncias corretamente classificadas em relação ao total de instâncias. É uma métrica geralmente aplicada a problemas de classificação. A acurácia é dada pela Equação 2.3.

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.3)$$

Precisão

Utilizando os conceitos abordados em Matriz de Confusão, a precisão é outra métrica interessante que estabelece uma razão entre todos os VP por todos os Positivos do *dataset* [26]. A Precisão é dada pela Equação 2.4.

$$Precisão = \frac{VP}{VP + FP} \quad (2.4)$$

Recall

A métrica *recall* é utilizada para quantificar os dados positivos que foram classificados corretamente, podendo ser vista, também, como a frequência que um modelo detecta exemplos de uma classe [35]. Matematicamente, é representada pela Equação 2.5.

$$Recall = \frac{VP}{VP + FN} \quad (2.5)$$

F1

A métrica F1 é interessante por combinar as métricas de precisão e *recall* a partir de uma média harmônica entre os dois valores (Equação 2.6). Altos valores para F1 indicam baixos FN e FP [35].

$$F1 = \frac{2 \times precisão \times recall}{precisão + recall} \quad (2.6)$$

2.4 Conclusão

O presente capítulo teve como objetivo fornecer uma base sólida para a compreensão de conceitos utilizados na abordagem metodológica que foi escolhida na implementação do trabalho de reconhecimento de bovinos. Nele, foram abordados desde elementos fundamentais de *Machine Learning*, até explicações mais detalhadas do funcionamento das Redes Neurais Convolucionais, em especial da arquitetura da DenseNet201 [10], que foi utilizada neste trabalho.

Este capítulo, portanto, serve como um ponto de partida para as próximas etapas da pesquisa, onde as nuances das CNNs aqui abordadas alimentarão diretamente a abordagem prática adotada no reconhecimento de bovinos por meio de técnicas de aprendizado de máquina.

Capítulo 3

Revisão de Literatura

A utilização de Redes Neurais Convolucionais para o reconhecimento de bovinos tem emergido como uma área de pesquisa fundamental no setor pecuário. As CNNs possuem um potencial significativo na identificação precisa de características individuais e distintas de bovinos. Esta revisão literária busca explorar os avanços recentes nesse campo, destacando a importância dessas tecnologias na gestão eficiente de rebanhos [8].

A capacidade das CNNs de proporcionar uma identificação individual precisa não apenas contribui para a rastreabilidade e gestão da saúde do gado, mas também promove a automação de processos, resultando em melhorias substanciais na eficiência e na qualidade da produção pecuária. Este estudo examina as aplicações práticas e benefícios associados ao emprego de CNNs no reconhecimento de bovinos, contribuindo não apenas para a compreensão da eficácia na identificação de bovinos, mas também para o avanço da gestão pecuária por meio de tecnologias inovadoras [4].

3.1 Contextualização

As redes neurais encontram aplicações em diversos setores, no entanto, sua utilização específica para a identificação de animais ainda necessita de atenção adequada. O reconhecimento de animais por meio de redes neurais convolucionais (CNNs) representa um nicho promissor e pouco explorado dentro do amplo espectro de aplicações dessa tecnologia [8].

As CNNs são particularmente eficazes na análise de imagens devido à sua capacidade de aprender padrões hierárquicos e características relevantes em diferentes níveis de abstração. No entanto, a aplicação dessa tecnologia para reconhecimento de animais apresenta desafios únicos, como a variação significativa nas formas, tamanhos e poses dos animais, bem como a diversidade de ambientes em que podem ser encontrados [4].

A importância de estudos nesse setor vai além do aspecto tecnológico, estendendo-se aos benefícios tangíveis que podem ser alcançados, como o monitoramento mais eficaz de espécies ameaçadas, a avaliação de ecossistemas em risco e a coleta de dados valiosos para a pesquisa biológica. Além disso, a automação do processo de identificação de animais por meio de CNNs poderia proporcionar uma abordagem escalável e eficiente para lidar com grandes conjuntos de dados, permitindo avanços significativos em diversas áreas científicas e ambientais.

3.2 Evolução do campo de Estudo

A relevância da rastreabilidade dos animais e de seus produtos derivados tem crescido à medida que os consumidores perdem o controle direto sobre a produção e a comercialização de alimentos. Os sistemas de rastreabilidade de produtos necessitam de uma cadeia transparente de ações para manter sua credibilidade e assegurar sua eficácia na transferência de informações. Esses sistemas devem incluir um mecanismo confiável e verificável, fundamental para preservar a identidade dos exemplares ao longo da cadeia alimentar [36].

Esse aumento na importância da rastreabilidade está intrinsecamente ligado à complexidade crescente da cadeia de suprimentos alimentar. À medida que os alimentos passam por diversas etapas, desde a produção até a distribuição, o consumidor final muitas vezes perde a capacidade de rastrear a origem dos produtos. A rastreabilidade, portanto, torna-se crucial para garantir a segurança alimentar e manter a confiança do consumidor [37].

Os sistemas de rastreabilidade eficazes não apenas documentam a jornada dos alimentos, mas também exigem transparência em todas as etapas do processo. Isso inclui a coleta de informações sobre a origem dos animais, as condições de criação, os métodos de processamento e os locais de distribuição. A credibilidade desses sistemas depende da capacidade de fornecer informações precisas e acessíveis a todos os participantes da cadeia alimentar [37].

Por essa razão, é importante que os sistemas de rastreabilidade incorporem mecanismos confiáveis e verificáveis. A confiabilidade garante que as informações registradas sejam precisas e não manipuladas, enquanto a verificabilidade permite que os interessados confirmem a autenticidade dos dados. Esses elementos são essenciais para preservar a identidade dos produtos ao longo da cadeia alimentar e mitigar riscos relacionados à segurança alimentar e à falsificação [36].

Além disso, a rastreabilidade dos animais emerge como um importante aliado para os produtores bovinos, desempenhando um diferencial em diversos aspectos da gestão pecuária-

ria. Este sistema oferece benefícios significativos que vão além da simples identificação da origem dos produtos. No contexto da produção bovina, a rastreabilidade proporciona aos produtores uma ferramenta valiosa para monitorar e aprimorar o desempenho do rebanho. Ao rastrear individualmente cada animal, os produtores podem coletar dados específicos sobre sua saúde, alimentação e condições de vida. Essas informações não apenas auxiliam na identificação precoce de possíveis problemas de saúde, mas também contribuem para a implementação de práticas de manejo mais eficientes e sustentáveis [38].

Adicionalmente, a rastreabilidade dos animais permite uma melhor gestão dos processos de reprodução e genética. Ao ter acesso a dados detalhados sobre a linhagem e histórico reprodutivo de cada animal, os produtores podem tomar decisões mais informadas na seleção de reprodutores, visando aprimorar características desejadas no rebanho, como resistência a doenças, eficiência alimentar e qualidade da carne [36].

Outro benefício relevante é a possibilidade de atendimento a requisitos específicos de mercados mais exigentes. Com a capacidade de rastrear cada fase da produção, os produtores podem demonstrar a conformidade com normas de bem-estar animal, práticas sustentáveis e padrões de qualidade, o que se torna um diferencial competitivo em mercados que valorizam tais aspectos [36].

Dessa forma, a rastreabilidade dos animais não apenas atende às demandas crescentes por transparência na cadeia alimentar, mas também se apresenta como uma ferramenta multifuncional para os produtores bovinos. Ao proporcionar controle, eficiência e vantagens competitivas, esse sistema contribui para a sustentabilidade e o aprimoramento contínuo da produção pecuária.

3.3 Tipos de Rastreio

No contexto pecuário brasileiro, o rastreamento de animais bovinos é uma prática fundamental para garantir a segurança alimentar, a qualidade dos produtos de origem animal e a gestão eficiente da cadeia produtiva. O Brasil adota diversas abordagens de rastreio, visando a identificação individual e o monitoramento ao longo do ciclo de vida dos bovinos [38].

O Sistema Brasileiro de Identificação Individual de Bovinos e Búfalos (SISBOV) desempenha um papel crucial na implementação de procedimentos e controles para a identificação eficaz de animais, conforme detalhado no Capítulo III do Anexo da Instrução Normativa nº 21, datada de 2 de abril de 2004 [39]. O Artigo 4º desta instrução estabelece que cada animal deve possuir uma identificação dupla, e oferece quatro opções distintas para a adoção desse sistema.

A primeira opção consiste na utilização de um brinco auricular na orelha direita, acompanhado de um *botton*, cujas dimensões variam entre 2,6 e 3,0 cm, na orelha esquerda. Este *botton* deve conter o número de manejo SISBOV, composto por seis algarismos, situados entre o 9º e o 14º número do sistema [39].

Alternativamente, a segunda opção propõe um brinco auricular na orelha direita, combinado com um dispositivo eletrônico para identificação do animal. Essa abordagem destaca a flexibilidade do SISBOV ao incorporar tecnologias avançadas para garantir a rastreabilidade. A terceira opção apresenta um brinco auricular na orelha direita e uma tatuagem na outra orelha, contendo o número de manejo SISBOV. Essa variação na identificação contribui para a diversificação dos métodos adotados [39].

Por fim, a quarta opção requer um brinco auricular na orelha direita e a marcação a ferro quente do número de manejo SISBOV na perna direita traseira. Esse processo específico ocorre na região situada abaixo de uma linha imaginária que conecta as articulações das patas dianteira e traseira. Essa abordagem destaca a robustez do sistema, empregando métodos tradicionais para a identificação dos animais [38].

Além destes, há também o uso de dispositivos eletrônicos, como chips RFID (Identificação por Radiofrequência). Estes chips, operam por meio de um sistema que utiliza ondas de rádio para transmitir dados entre um leitor e uma etiqueta ou chip RFID. Essa tecnologia permite a identificação automática e a coleta de informações remotamente, sem a necessidade de contato direto entre o leitor e a etiqueta [40].

O processo inicia quando o leitor RFID emite um sinal de radiofrequência. A etiqueta RFID, que contém um microchip e uma antena, recebe esse sinal e converte a energia recebida para alimentar o chip. Em resposta, o chip transmite os dados armazenados de volta ao leitor por meio de ondas de rádio. Esses dados podem incluir informações específicas sobre o objeto ou animal ao qual a etiqueta está associada. A eficácia do sistema RFID reside na capacidade de realizar leituras rápidas e precisas a distâncias variáveis, dependendo do tipo de chip e do leitor utilizado [40].

A aplicação de redes neurais na rastreabilidade de bovinos é uma abordagem promissora e assertiva em comparação com métodos tradicionais, uma vez que a tecnologia de redes neurais oferece vantagens significativas ao lidar com a complexidade e a dinâmica inerentes à cadeia de produção de bovinos. Ao invés dos métodos convencionais, como a identificação por brincos ou documentos físicos, as redes neurais proporcionam uma melhor automação do processo, reduzindo a dependência de intervenção humana no processo de rastreio.

A principal viabilidade dessa abordagem reside na capacidade das redes neurais em processar grandes volumes de dados de forma simultânea e automática. Elas podem aprender padrões complexos e extrair informações relevantes a partir de conjuntos de

dados extensos, o que é crucial para o rastreamento eficaz ao longo de toda a cadeia de produção. Isso inclui informações sobre a origem do gado, histórico de saúde, movimentação, condições de transporte. Além disso, as redes neurais têm a capacidade de lidar com dados não estruturados, como imagens e vídeos, o que amplia as possibilidades de rastreabilidade. A identificação biométrica por meio de características específicas dos animais, como padrões faciais ou marcações individuais, pode ser facilmente integrada ao processo de rastreamento [41].

A implementação de redes neurais na rastreabilidade de bovinos também contribuiu para a automação do processo, reduzindo a dependência de intervenções manuais. Isso resulta em eficiência operacional e, conseqüentemente, em menor margem de erro nas etapas de identificação e monitoramento. Outro ponto relevante é a escalabilidade dessa abordagem. À medida que a demanda por produtos de origem animal rastreáveis aumenta, as redes neurais podem se adaptar facilmente a diferentes volumes de dados e cenários, tornando-as uma solução versátil para a indústria pecuária [41].

3.4 Trabalhos correlatos

O presente trabalho fundamenta-se na pesquisa intitulada "*Recognition of Pantaneira cattle breed using computer vision and convolutional neural networks*" conduzida por Weber *et al.*, em 2020 [42]. Essa pesquisa adotou em sua abordagem metodológica o desenvolvimento de um *dataset*, com 51 classes de bovinos, e realizou a comparação do desempenho do reconhecimento de bovinos em três arquiteturas de rede, sendo elas a ResNet50 [43], InceptionResNetV2 [44] e DenseNet201 [10]. No artigo foram utilizados os parâmetros descritos na Tabela 3.1 para a construção dos modelos utilizados.

Tabela 3.1: Parâmetros utilizados por Weber *et al.* [42]

Parâmetros	Valores
Dimensões da imagem de entrada	256x256
Tamanho do lote	32
Épocas	50
Otimizador	SGD
Taxa de Aprendizado	0.0001

Na implementação de Weber *et al.*, foi utilizado transferência de aprendizado com o conjunto de dados da ImageNet [32]. Os modelos propostos foram treinados utilizando o conjunto de dados RecBov41c [45], contendo 51 classes de bovinos distribuídas em 27.849

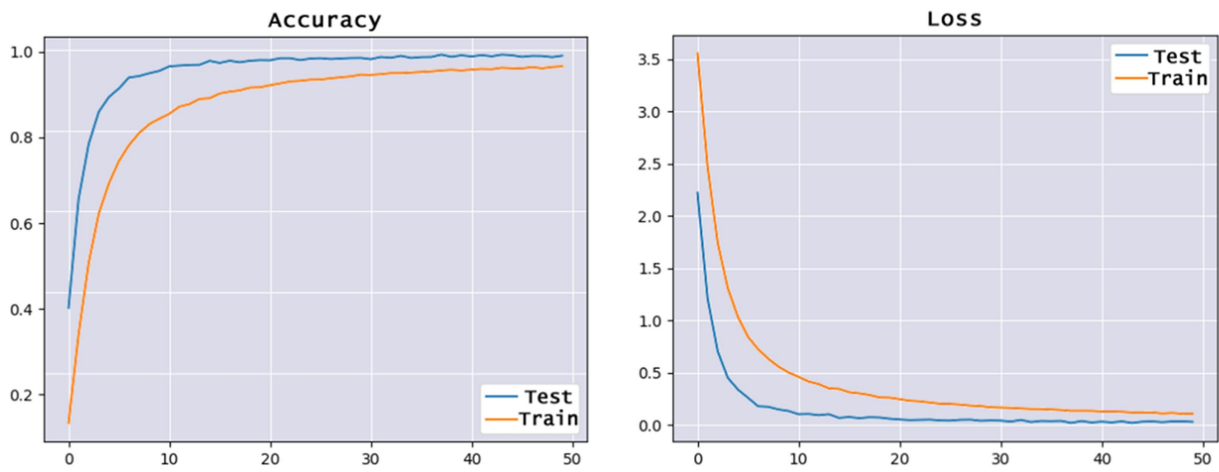
imagens. A divisão do *dataset* foi de 90% para o conjunto de treinamento e 10% para o conjunto de teste. Além disso, aplicou-se a técnica de validação cruzada *k-fold*, com $k = 10$.

Na condução da pesquisa de Weber *et al.*, destacaram-se resultados promissores na avaliação de métricas relacionadas à identificação de bovinos através de CNNs, conforme evidenciado nos dados apresentados na Tabela 3.2 e ilustrado na Figura 3.1. Esses indicadores favoráveis fortalecem a validade e a eficácia da abordagem adotada no contexto do reconhecimento de bovinos, ressaltando o potencial e a robustez das CNNs para esta aplicação específica. O desempenho expressivo das métricas valida a pertinência da escolha metodológica, evidenciando resultados encorajadores no âmbito da identificação precisa de animais bovinos por meio dessa tecnologia avançada de aprendizado profundo.

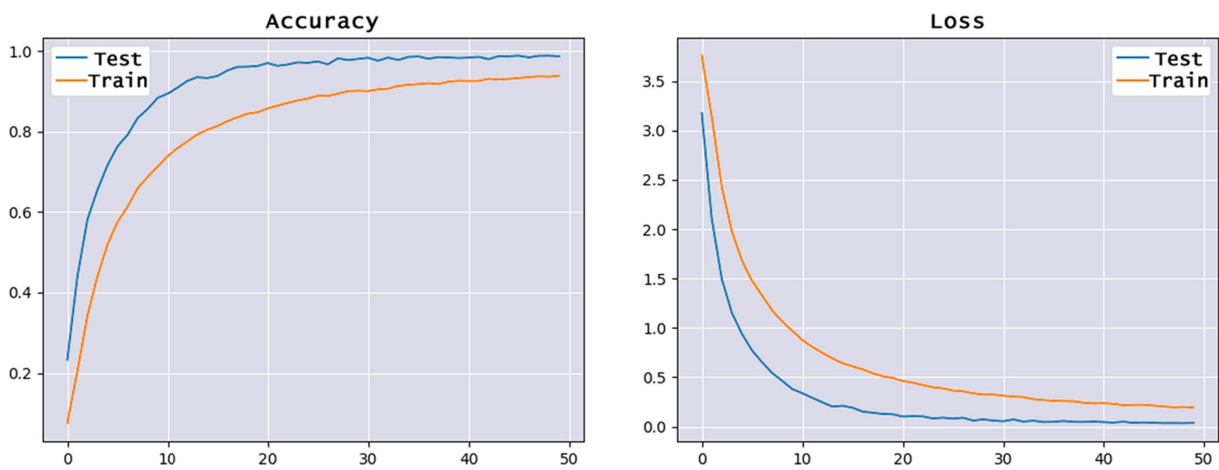
A presente monografia fundamenta-se, predominantemente, nos resultados e metodologias expostos na pesquisa conduzida por Weber *et al.*, que não apenas serviu como ponto de partida para este trabalho, mas também proporcionou *insights* valiosos sobre a utilização de técnicas avançadas de visão computacional e redes neurais convolucionais no reconhecimento de raças bovinas, ampliando assim o conhecimento e as possibilidades dentro desse domínio específico.

Tabela 3.2: Resultados obtidos por Weber *et al.* [42]

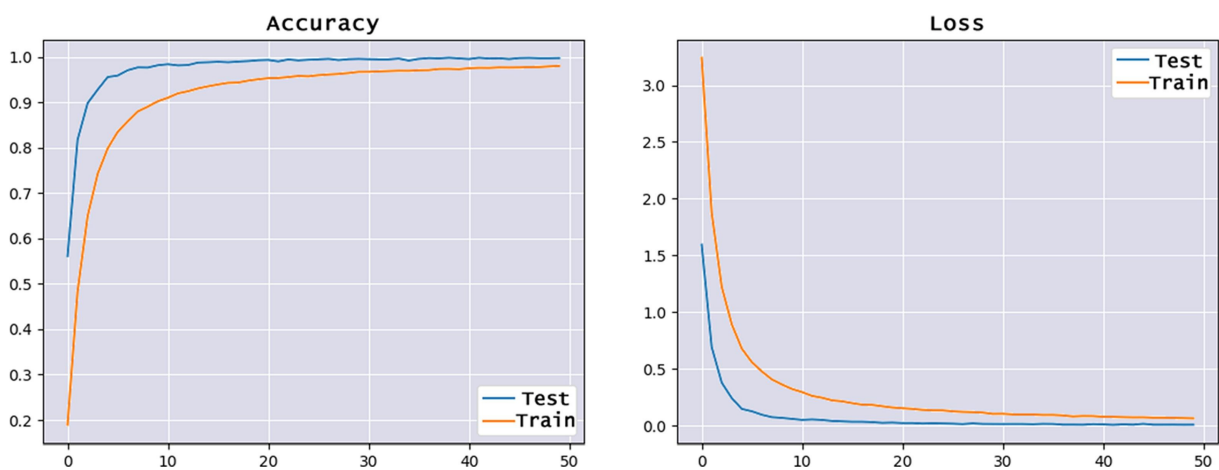
Arquitetura	Acurácia de Treinamento (%)	Acurácia de Teste (%)	Duração de treinamento	Duração de teste
ResNet50	99,20	99,78	52h:41m	2h:20m
InceptionResNetV2	98,87	99,52	29h:29m	6h:42m
DenseNet201	99,74	99,85	7h:26m	5h:48m



(a) Acurácia e função de custo da ResNet50.



(b) Acurácia e função de custo da InceptionResNetV2.



(c) Acurácia e função de custo da DenseNet201.

Figura 3.1: Resultados obtidos por Weber *et al.*. Fonte: [42]

Capítulo 4

Metodologia

No âmbito da revolução tecnológica que permeia o mundo moderno e seus diversos setores econômicos, incluindo o da agropecuária, torna-se fundamental a evolução da pesquisa de técnicas de reconhecimento de bovinos, como aprimoramento de técnicas de rastreamento. O presente capítulo possui o objetivo de oferecer uma visão detalhada das abordagens metodológicas adotadas na implementação de um modelo de CNN voltado para o reconhecimento de bovinos.

A Seção 4.1 proporcionará uma compreensão mais aprofundada da implementação adotada durante o desenvolvimento dos experimentos, abordando detalhes da arquitetura utilizada. Na Seção 4.2, serão discutidos pormenores referentes ao conjunto de dados empregado no treinamento e validação do modelo. Posteriormente, na Seção 4.3, serão delineados os procedimentos aplicados às imagens antes de sua utilização no treinamento do modelo. Por fim, a Seção 4.4 apresentará uma análise detalhada dos experimentos propostos para a condução deste trabalho.

4.1 Implementação

Para a implementação dos experimentos propostos, foi utilizada a linguagem Python¹. Vale também destacar o uso da biblioteca Tensorflow, que disponibilizou uma implementação para modelos pré-treinados que foram utilizados nos experimentos, bem como funções para o pré-processamento das imagens contidas no *dataset*.

¹O código desenvolvido pode ser encontrado no repositório do GitHub: <https://github.com/kallebe/cattle-recognition-cnn>

4.1.1 Arquitetura

Para os experimentos delineados, adotou-se a arquitetura DenseNet201 [10], foi utilizada a técnica de transferência de aprendizado, empregando pesos pré-treinados com o *dataset* da ImageNet [32]. Durante o treinamento do modelo, realizou-se o *fine-tuning* de todas as camadas da rede. Na configuração da arquitetura, optou-se por excluir a última camada de classificação. Como substituição, foi implementada uma nova camada totalmente conectada composta por 50 neurônios, nas primeiras etapas do experimento, e 49 neurônios, nas etapas posteriores, cada neurônio representando uma classe distinta. A escolha da função de ativação para a camada de saída foi a função softmax.

Tabela 4.1: Detalhes da Arquitetura utilizada

Camada (tipo)	Dimensões de saída	Número de parâmetros
DenseNet201 (Funcional)	$8 \times 8 \times 1920$	18.321.984
<i>Global Average Pooling</i> 2D	1×1920	0
Totalmente conectada	1×49	94.129

Total de parâmetros	18.416.113 (70,25 MB)
Parâmetros treináveis	18.187.057 (69,38 MB)
Parâmetros não-treináveis	229.056 (894,75 KB)

A Tabela 4.1 fornece uma análise detalhada das camadas do modelo empregado, detalhando as dimensões de saída de cada camada e o número de parâmetros associados, indicando a quantidade de pesos e vieses que o modelo precisa aprender durante o treinamento. Como mencionado anteriormente, foi utilizada a DenseNet201 [10], que é uma rede neural convolucional (CNN) profunda. Na Tabela 4.1 não foram detalhadas as camadas da DenseNet201 [10]. A Figura 4.1, contudo, apresenta um diagrama das camadas da DenseNet201 [10], já com a camada de classificação adaptada ao modelo utilizado nesta monografia.

Para a transição das camadas convolucionais às camadas totalmente conectadas, foi utilizada a camada de *Global Average Pooling* 2D com o objetivo de reduzir as dimensões espaciais (altura e largura) do tensor, fazendo a média global de cada canal, resultando em um vetor unidimensional de tamanho 1920. Após essa camada, a arquitetura possui uma camada totalmente conectada final, com uma saída para 49 neurônios, indicando a saída para classificação em 49 classes.

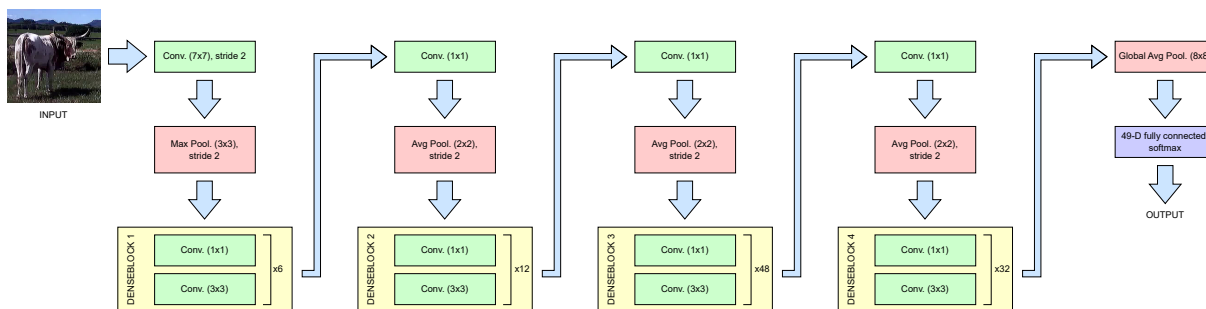


Figura 4.1: Detalhamento da arquitetura DenseNet201 [10], aplicada ao modelo de reconhecimento de bovinos proposto.

Quanto ao otimizador, a escolha recaiu sobre o Gradiente Descendente Estocástico (SGD), configurado com uma taxa de aprendizado de 0,0001. A função de custo adotada foi a entropia cruzada, recomendada para problemas de classificação multi-classe, dada sua eficácia em quantificar a discrepância entre as distribuições de probabilidade previstas e reais.

4.1.2 Validação cruzada

Para a divisão do conjunto de dados, empregou-se a técnica de validação cruzada *k-fold*, onde o k foi definido como 5. Assim, o *dataset* foi, primeiramente, dividido em conjunto de teste, compreendendo 10% do conjunto de dados, enquanto os 90% remanescentes foram alocados para treinamento e validação do modelo. Essa porção de 90% do conjunto de dados foi subsequente e equitativamente particionada em $k = 5$ segmentos. Devido ao desbalanceamento intrínseco no conjunto de dados, buscou-se preservar a proporção de amostras de cada classe em cada partição.

Durante cada uma das 5 iterações da técnica de validação cruzada *k-fold*, o modelo foi treinado utilizando 4 das partições para treinamento e 1 para validação. Cada iteração compreendeu 20 épocas de treinamento. Ao término dessas iterações, calculou-se a média das métricas referentes à acurácia e à função de custo de cada época. Esse procedimento foi realizado com o objetivo de apresentar métricas de desempenho com menor variância, proporcionando uma visão mais estável e robusta em relação à separação do conjunto de dados.

4.2 Conjunto de dados

Para o treinamento do modelo proposto para reconhecimento de bovinos, foi utilizado o *dataset* RecBov51c [45]. O conjunto de dados foi coletado em 2019 no Núcleo de Conservação de Bovinos Pantaneiros de Aquidauana (NUBOPAN), na Universidade Estadual de

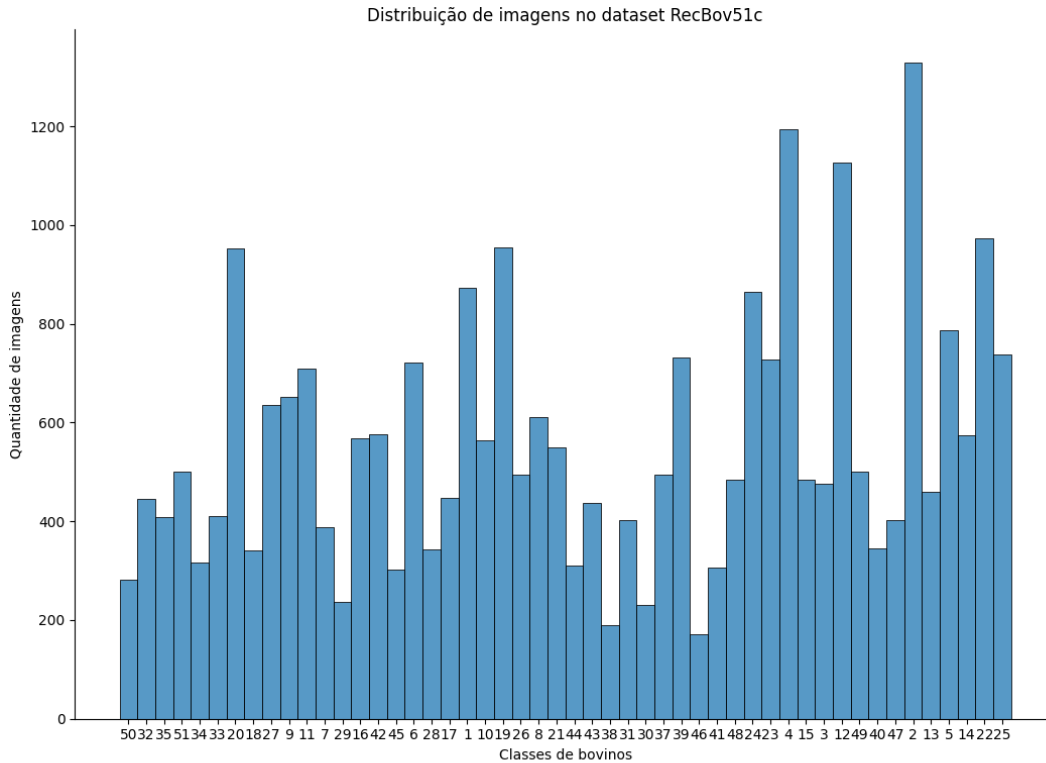


Figura 4.2: Distribuição de imagens no conjunto de dados RecBov51c [45]

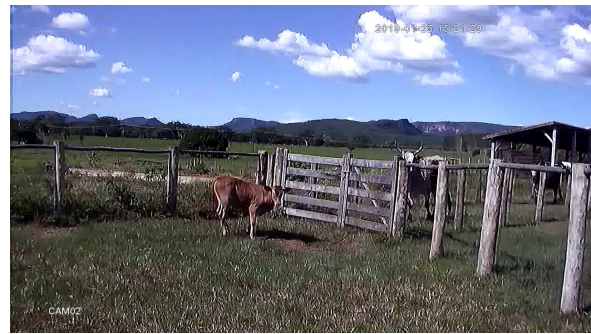
Mato Grosso do Sul (UEMS), por Weber *et al.* [42]. O *dataset* contém 28.025 imagens pertencentes a 50 diferentes classes, onde cada classe representa um bovino da raça Pantaneira. As classes pertencentes ao conjunto de dados incluem bovinos de ambos os sexos, e de variadas idades [42]. A coleta das imagens do conjunto de dados foi realizada a partir da extração de quadros de vídeos capturados de cada bovino, resultando em imagens com a resolução de 1280×720 .

Após uma análise no conjunto de dados utilizado, foram percebidas certas características que devem ser levadas em consideração antes do treinamento do modelo de CNN. Primeiramente, nota-se que a distribuição de imagens entre as classes do *dataset*, como apresentado na Figura 4.2, apresenta um nível de desbalanceamento, onde estão presentes classes com mais de 1200 imagens, enquanto outras possuem menos de 200 imagens.

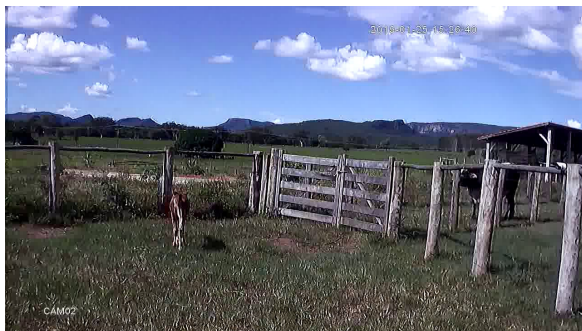
Ainda, na Figura 4.3, são constatados alguns problemas que podem afetar negativamente o desempenho do modelo. A Figura 4.3a é um exemplo de imagens do conjunto de dados que não possuem nenhum gado presente. A Figura 4.3b, por sua vez, apresenta um exemplo em que a imagem possui mais de um gado na imagem. Outro problema identificado é a distância ou posição, em alguns momentos, do bovino em relação à câmera, fazendo com que a presença do bovino na imagem seja de difícil detecção. Exemplos disso estão apresentados nas Figuras 4.3c a 4.3d. Tais problemas identificados devem ser trata-



(a) Imagem sem nenhum boi presente



(b) Imagem com mais de um boi



(c) Boi quase indetectável na imagem



(d) Imagem com boi oculto na imagem

Figura 4.3: Problemas encontrados no conjunto de dados utilizado. Fonte: [45]

dos antes da apresentação do conjunto de dados ao modelo para a etapa de treinamento. Os tratamentos realizados serão abordados na Seção 4.3.

Além dos desafios previamente destacados, identificou-se uma inconsistência adicional no *dataset*, evidenciada pela presença de duas classes relacionadas ao mesmo bovino, conforme ilustrado na Figura 4.4. Ambas as classes compartilhavam exatamente as mesmas imagens. Em virtude dessa irregularidade, os primeiros experimentos foram conduzidos utilizando o *dataset* com as 50 classes originalmente designadas. No entanto, em fases subsequentes da pesquisa, ao identificar essa anomalia, procedeu-se com a exclusão da classe 51, preservando, assim, as 49 classes restantes do conjunto de dados. Esta ação corretiva visou assegurar a integridade e coerência do *dataset*, promovendo uma base mais consistente para análises e interpretações durante os estágios posteriores da pesquisa.

4.3 Pré-processamento das imagens

Com o propósito de abordar as questões identificadas no conjunto de dados discutido na Seção 4.2, realizou-se uma seleção criteriosa das imagens no *dataset*, excluindo aquelas em que a presença de bovinos não era claramente identificável. O *dataset*, após a seleção das

imagens, passou a conter 11.150 imagens. A distribuição do conjunto de dados selecionado está apresentado na Figura 4.5. Após a exclusão da classe 51, como mencionado na Seção 4.2, o *dataset* selecionado passou a ter 10.967 imagens, distribuídas entre 49 classes.

Após a etapa de seleção, executou-se uma operação de recorte nas imagens remanescentes. Os recortes foram realizados com um *aspect ratio* de 1:1, resultando em imagens quadradas. A intenção era manter os bovinos centralizados na imagem, proporcionando uma representação mais focalizada e precisa, visando o melhor desempenho no aprendizado de características pelo modelo. Algumas das imagens recortadas estão apresentadas na Figura 4.6.

4.3.1 Aumento de dados

Além da etapa de seleção e recorte das imagens do dataset, foi aplicada uma técnica de aumento de dados, a fim de aumentar a disponibilidade de dados do *dataset* introduzindo pequenas variações das imagens já presentes e introduzindo novas instâncias com variações sutis em relação ao conjunto de dados original.

Diversas técnicas de aumento de dados foram utilizados nos experimentos. Dentre as técnicas utilizadas, estão:

- a normalização dos valores dos pixels, através da divisão de cada pixel da imagem por 255 com o objetivo de manter os valores presentes na imagem no intervalo $[0, 1]$;
- o espelhamento horizontal das imagens, introduzindo a invariância em relação à orientação;
- variações de zoom em até 30%;
- deslocamentos horizontais e verticais em até 30% da largura/altura da imagem;
- rotações em até 30°



Figura 4.4: Bovino presente nas classes 39 e 51 do *dataset*. Fonte: [45]

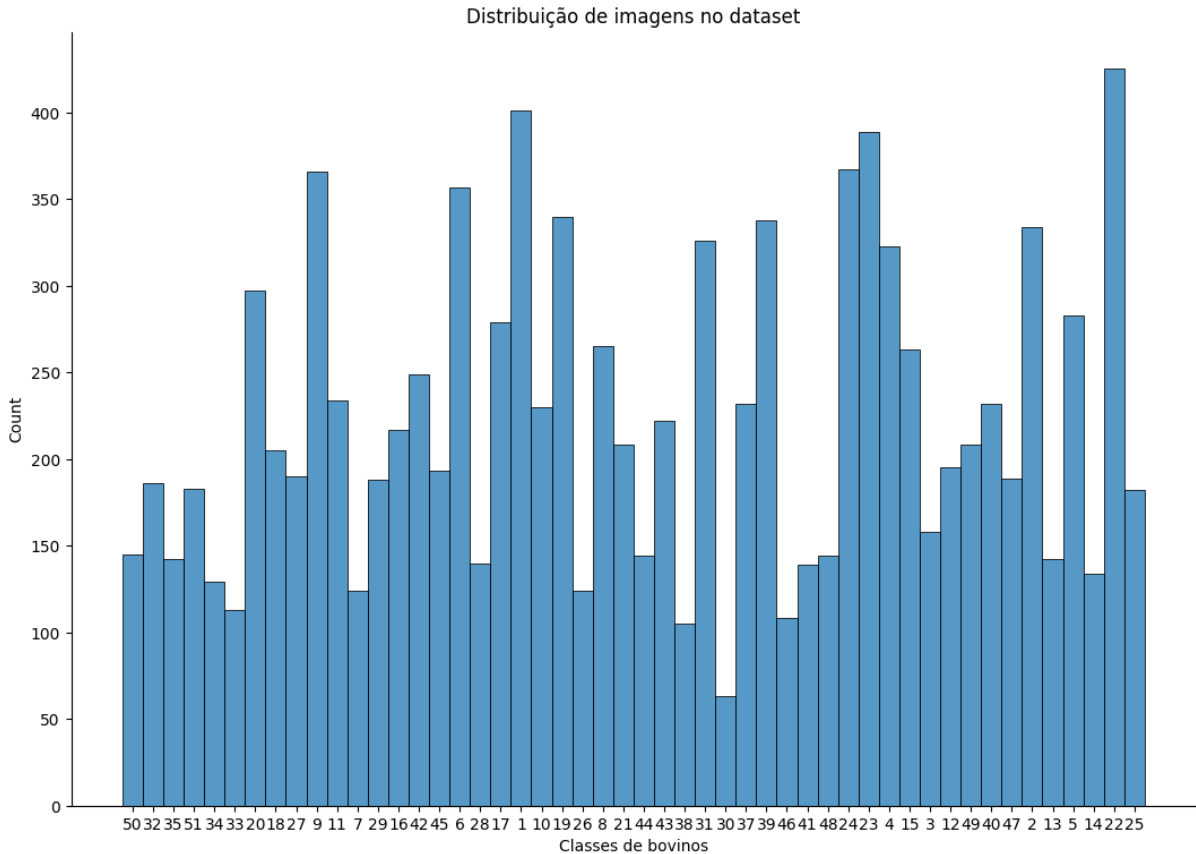


Figura 4.5: Distribuição de imagens no conjunto de dados selecionado.

Tais pré-processamentos foram realizados a fim de enriquecer o conjunto de dados utilizado, contribuindo para uma melhor generalização para dados não vistos na etapa de treinamento do modelo.

4.4 Experimentos

O presente trabalho propõe a realização de experimentos para a avaliação do desempenho de uma Rede Neural Convolutiva, utilizando a arquitetura da DenseNet201 [10], no reconhecimento de bovinos em imagens. Para isso, foram definidos três diferentes experimentos, descritos na Tabela 4.2. O primeiro experimento foi separado em dois momentos diferentes: no primeiro momento, o modelo foi treinado com as 50 classes presentes no *dataset*, incluindo a classe 51 que é uma réplica da classe 49; no segundo o modelo foi treinado apenas para as 49 classes, excluindo a classe 51 do *dataset*.

Os experimentos propostos visam treinar o modelo de CNN utilizando o conjunto de dados RecBov51c [45] em três diferentes etapas e avaliar o desempenho alcançado para cada experimento, a partir da análise de métricas obtidas no treinamento do modelo.



Figura 4.6: Exemplos de imagens recortadas do conjunto de dados. Fonte: [45]

Após a análise dos resultados, espera-se definir a viabilidade do uso de aplicações de Redes Neurais Convolucionais na tarefa de identificação de bovinos.

Tabela 4.2: Descrição dos experimentos propostos

Experimento	Descrição	Quantidade de imagens no <i>dataset</i>	Quantidade de classes
E1a	Treinamento do modelo utilizando o <i>dataset</i> original	28.025	50
E1b	Treinamento do modelo utilizando o <i>dataset</i> original após a remoção da classe duplicada 51	27.524	49
E2	Treinamento do modelo com imagens do <i>dataset</i> selecionadas, excluindo imagens indesejadas para o treinamento e a classe 51	10.967	49
E3	Treinamento do modelo com imagens do <i>dataset</i> selecionadas e cortadas em <i>aspect ratio</i> 1:1 sem a classe 51	10.967	49

Capítulo 5

Resultados

O presente capítulo representa o culminar de uma jornada dedicada à exploração, implementação e avaliação da solução proposta para o reconhecimento de bovinos em imagens por meio de Redes Neurais Convolucionais (CNNs). Ao longo dos capítulos anteriores, foi delineada a motivação, fundamentação teórica e metodologia adotada. Este capítulo, portanto, visa apresentar os resultados de toda a jornada trilhada até o momento.

Esta seção não apenas destaca os resultados quantitativos obtidos em termos de métricas de desempenho, mas também busca oferecer uma compreensão dos fatores que moldaram esses resultados. A avaliação cuidadosa da arquitetura da rede neural, a análise de parâmetros e hiperparâmetros e a exploração visual de resultados formam a essência deste capítulo.

5.1 Resultados

A seguir, apresentamos os resultados obtidos para cada experimento proposto, conforme detalhado na Tabela 5.1. Esta abordagem oferece uma visão direta do desempenho em diferentes configurações experimentais, permitindo uma compreensão clara e abrangente das contribuições específicas de cada abordagem ao escopo da pesquisa.

5.1.1 Experimento E1

Para a condução do Experimento E1a, inicialmente, os hiperparâmetros delineados na Tabela 5.2 foram estabelecidos. Nesta primeira iteração, optou-se por não aplicar as técnicas de aumento de dados mencionadas na Seção 4.3. Considerando ser uma fase preliminar, a técnica de validação cruzada *k-fold* também não foi incorporada, sendo adotada a estratégia de *hold-out*. Nesse cenário, 20% do conjunto de dados foram reservados para a validação do modelo, enquanto os 80% restantes foram destinados ao treinamento. A

Tabela 5.1: Descrição dos experimentos propostos

Experimento	Descrição	Quantidade de imagens no <i>dataset</i>	Quantidade de classes
E1a	Treinamento do modelo utilizando o <i>dataset</i> original	28.025	50
E1b	Treinamento do modelo utilizando o <i>dataset</i> original após a remoção da classe duplicada 51	27.524	49
E2	Treinamento do modelo com imagens do <i>dataset</i> selecionadas	10.967	49
E3	Treinamento do modelo com imagens do <i>dataset</i> selecionadas e cortadas em <i>aspect ratio</i> 1:1	10.967	49

Figura 5.1 ilustra as curvas de acurácia e função de custo para cada época de treinamento, proporcionando uma visão detalhada do desempenho do modelo ao longo do processo.

Tabela 5.2: Principais hiperparâmetros da 1ª implementação do Experimento E1a

Hiperparâmetro	Valor
Épocas	10
<i>Dropout</i>	40%
Taxa de aprendizado	0,0001
Otimizador	Gradiente Descendente Estocástico

Observa-se que, embora a curva de treinamento alcance uma acurácia de 82,43%, o conjunto de validação do modelo atinge apenas 32,2%, sugerindo a presença de *overfitting*. Essa disparidade entre o desempenho nos dados de treinamento e validação sugere que o modelo pode estar se ajustando excessivamente aos padrões específicos do conjunto de treinamento, perdendo sua capacidade de generalização para novos dados. A identificação desse fenômeno é crucial para aprimorar a robustez e a eficácia do modelo, destacando a necessidade de estratégias de regularização ou ajustes nos hiperparâmetros para mitigar o *overfitting*.

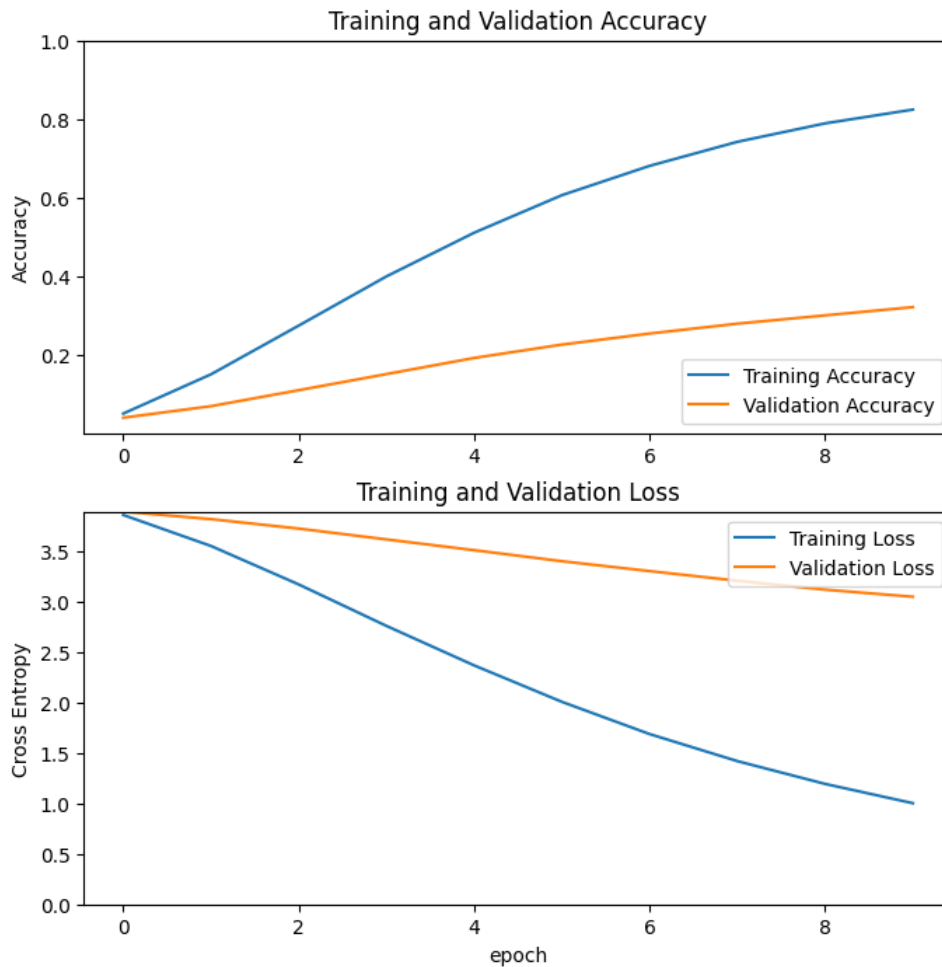


Figura 5.1: Curvas de acurácia e função de perda para a primeira implementação do Experimento E1a.

Após a análise do primeiro teste de implementação, foram implementadas modificações na arquitetura da rede, visando aprimorar o desempenho do modelo de reconhecimento de bovinos. Após efetuar as alterações necessárias, chegou-se à arquitetura detalhada na Seção 4.1. No processo de treinamento e avaliação dessa arquitetura, a classe duplicada 51 foi removida do conjunto de dados. Portanto, os resultados apresentados nas Figuras 5.2 a 5.3 correspondem ao Experimento E1b. Além disso, empregou-se o método de validação cruzada k -fold, conforme descrito na Seção 4.1.2, para uma avaliação mais abrangente e confiável do desempenho do modelo. Essas adaptações representam uma abordagem iterativa e informada para refinamento da arquitetura, refletindo um comprometimento com a melhoria contínua do modelo proposto.

Evidencia-se uma notável melhoria no desempenho do modelo, conforme ilustrado na Figura 5.2, em comparação ao desempenho anterior evidenciado na Figura 5.1. A Tabela 5.4 sumariza as medidas de desempenho obtidas no conjunto de teste, revelando

Tabela 5.3: Arquitetura utilizada para a primeira implementação do Experimento E1a

Camada	Dimensões de saída	Número de parâmetros
DenseNet201 (Funcional)	$8 \times 8 \times 1920$	18.321.984
Convolução 2D	$6 \times 6 \times 32$	552.992
<i>Dropout</i>	$6 \times 6 \times 32$	0
<i>Global Average Pooling 2D</i>	1×32	0
Totalmente conectada	1×50	1.650

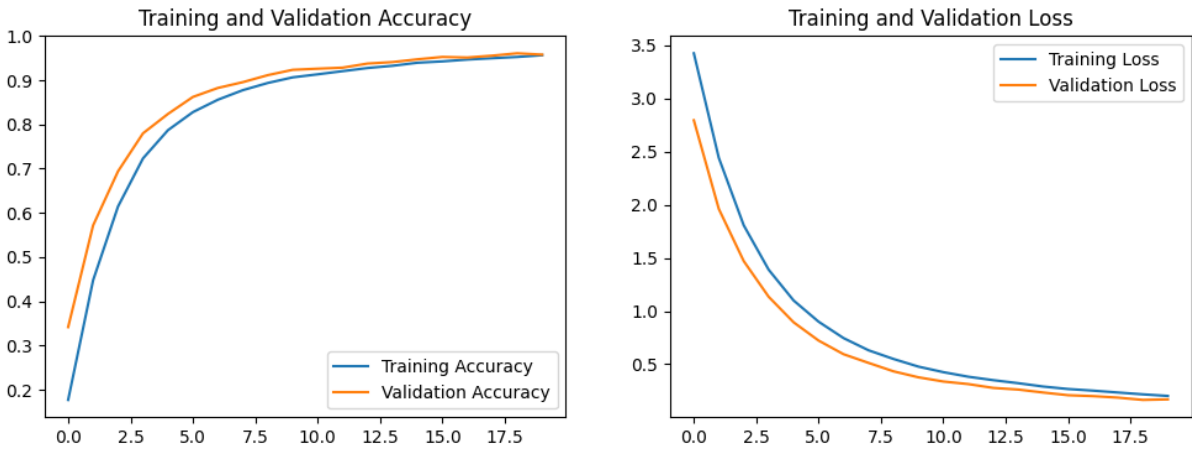


Figura 5.2: Métricas de treinamento e validação em cada época do Experimento E1b.

resultados mais promissores. Ademais, para uma análise mais visual e detalhada do desempenho do modelo, a matriz de confusão na Figura 5.3 oferece uma representação gráfica esclarecedora. Essa evolução positiva destaca o impacto das modificações implementadas na arquitetura da rede, evidenciando um progresso substancial em direção aos objetivos propostos.

Tabela 5.4: Acurácia e função de perda do Experimento E1b no conjunto de teste

Medida	Valor
Acurácia	98,51%
Função de perda	0.07125

desse processo de seleção.

O treinamento do modelo foi conduzido utilizando a técnica de validação cruzada k -fold, com $k = 5$, durante 20 épocas. Essa escolha busca garantir uma avaliação abrangente, abordando diferentes combinações de conjuntos de treinamento e validação. As curvas de acurácia e função de perda geradas após o treinamento são apresentadas na Figura 5.4, proporcionando *insights* visuais sobre a evolução do modelo ao longo do tempo.

Após a conclusão do treinamento, procedemos à avaliação do desempenho do modelo no conjunto de teste. As métricas resultantes, apresentadas na Tabela 5.5, oferecem uma visão quantitativa do sucesso do modelo ao generalizar para dados não vistos. Essa avaliação de desempenho em um conjunto de dados ainda não apresentado ao modelo é essencial para compreender a capacidade do modelo de reconhecimento de bovinos em um cenário mais amplo.

O Experimento E2 não apenas contribui para a evolução contínua do modelo, mas também destaca a importância da seleção de dados e da estratégia de treinamento na eficácia do reconhecimento de bovinos. Os resultados obtidos servirão como base para ajustes e otimizações adicionais, conforme buscamos aprimorar ainda mais a precisão e robustez do modelo.

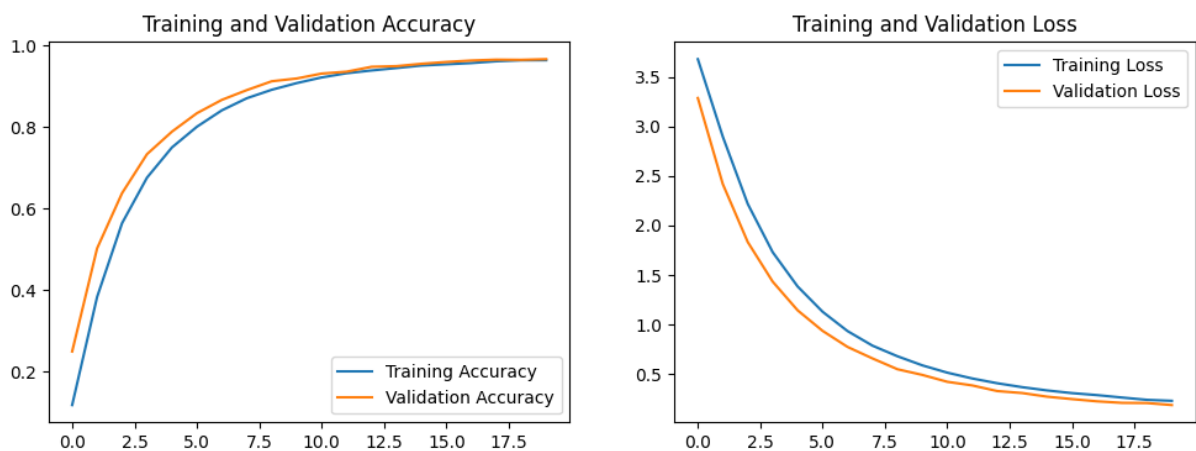


Figura 5.4: Métricas de treinamento e validação em cada época do Experimento E2.

Tabela 5.5: Acurácia e função de perda do Experimento E2 no conjunto de teste

Medida	Valor
Acurácia	98,63%
Função de perda	0.1001

5.1.3 Experimento E3

No Experimento E3, aprimoramos o modelo através de uma abordagem focada no recorte das imagens, uma extensão lógica dos ajustes feitos nos experimentos anteriores. Este procedimento visa potencializar a identificação do bovino nas imagens, enfatizando regiões de interesse e contribuindo para uma representação mais precisa dos padrões visuais distintivos de cada classe.

O gráfico apresentado na Figura 5.5, que destaca as curvas de acurácia e função de custo ao longo das épocas de treinamento, evidencia uma evolução positiva do modelo. Ao comparar com os experimentos anteriores (E1 e E2), observa-se uma progressão contínua, indicando que os ajustes feitos, particularmente o recorte aprimorado, contribuíram para um aprendizado mais eficaz.

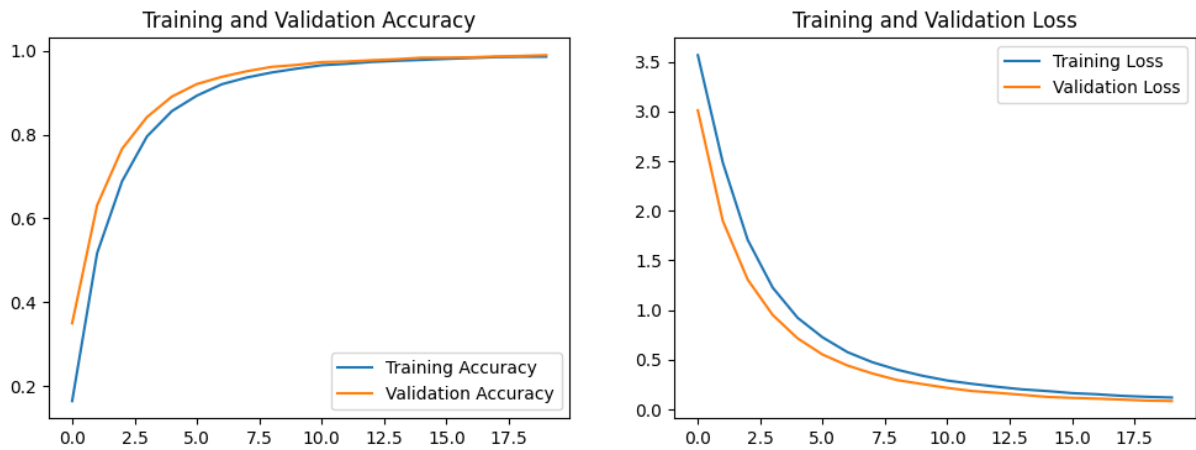


Figura 5.5: Métricas de treinamento e validação em cada época do Experimento E3.

Para uma avaliação mais robusta, recorreremos ao conjunto de teste inicialmente segregado. As medidas obtidas da avaliação do modelo neste conjunto estão apresentadas na Tabela 5.6. As predições do modelo neste conjunto permitiram a criação da matriz de confusão, ilustrada na Figura 5.6. Os resultados apresentados são encorajadores, evidenciando um valor reduzido de erro de classificação. Adicionalmente, as métricas detalhadas de precisão, *recall* e F1, apresentadas nas Figuras 5.7 a 5.9 para cada classe, oferecem uma compreensão mais refinada do desempenho diferenciado do modelo em diversas categorias.

Tabela 5.6: Acurácia e função de perda do Experimento E3 no conjunto de teste

Medida	Valor
Acurácia	99,73%
Função de perda	0.0547

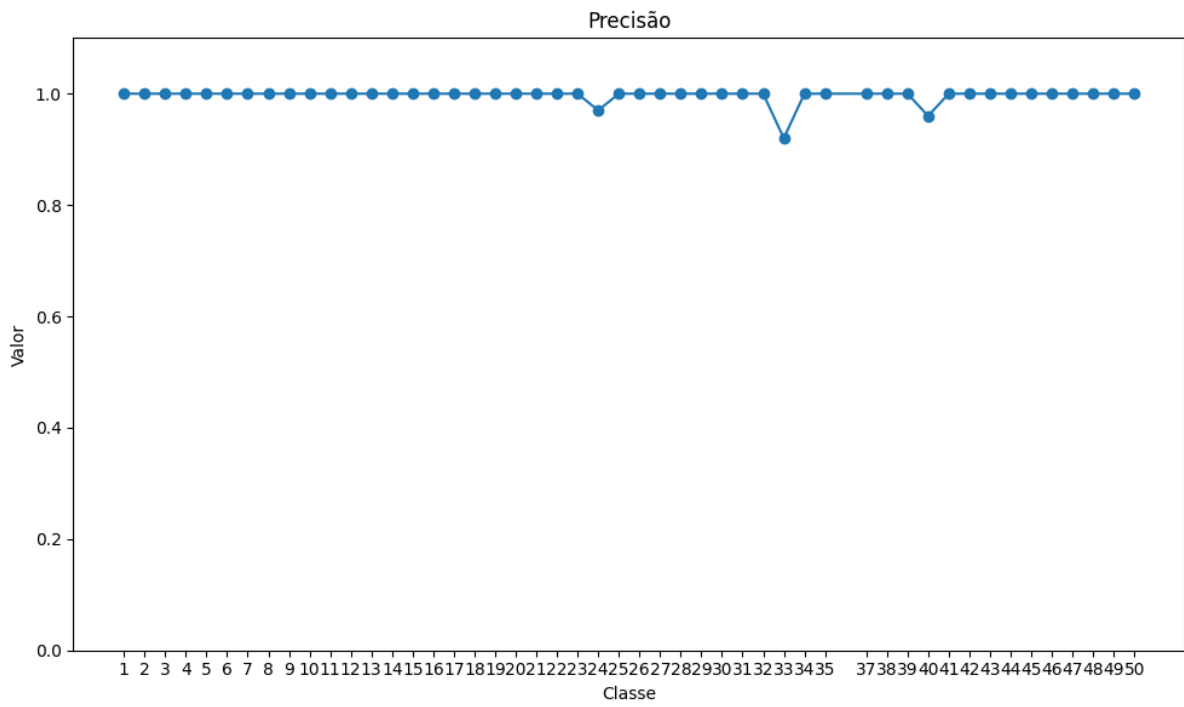


Figura 5.7: Métricas de precisão alcançadas por cada classe no Experimento E3.

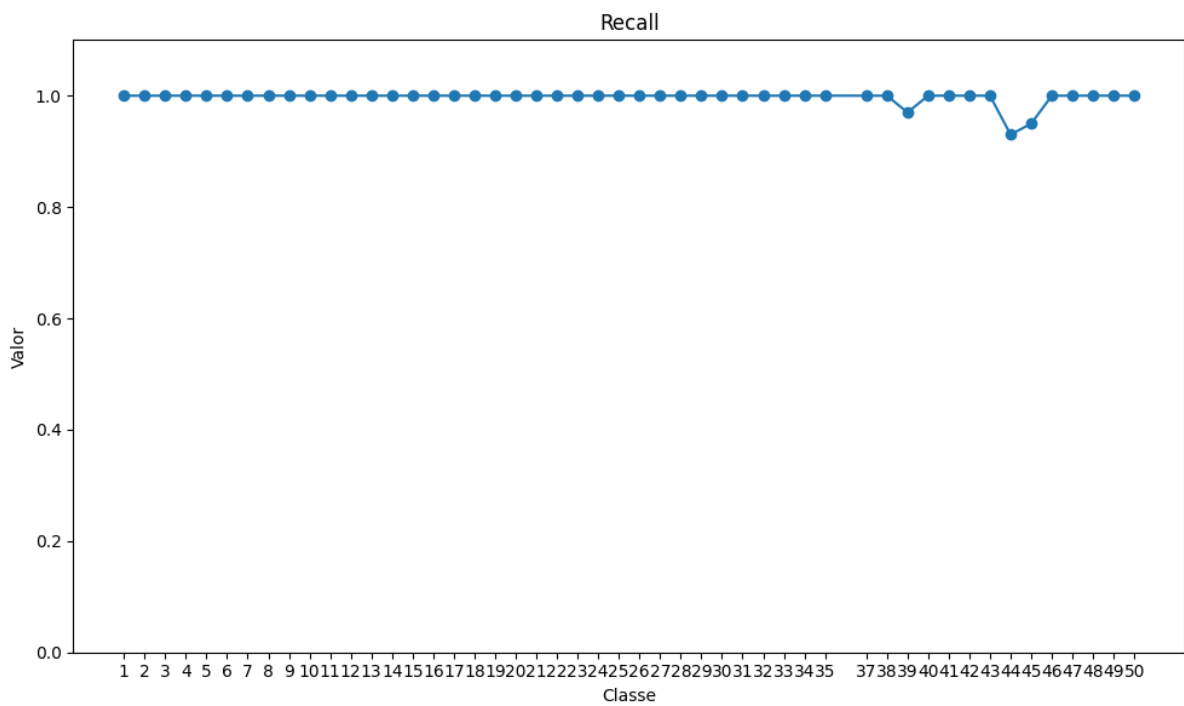


Figura 5.8: Métricas de *recall* alcançadas por cada classe no Experimento E3.

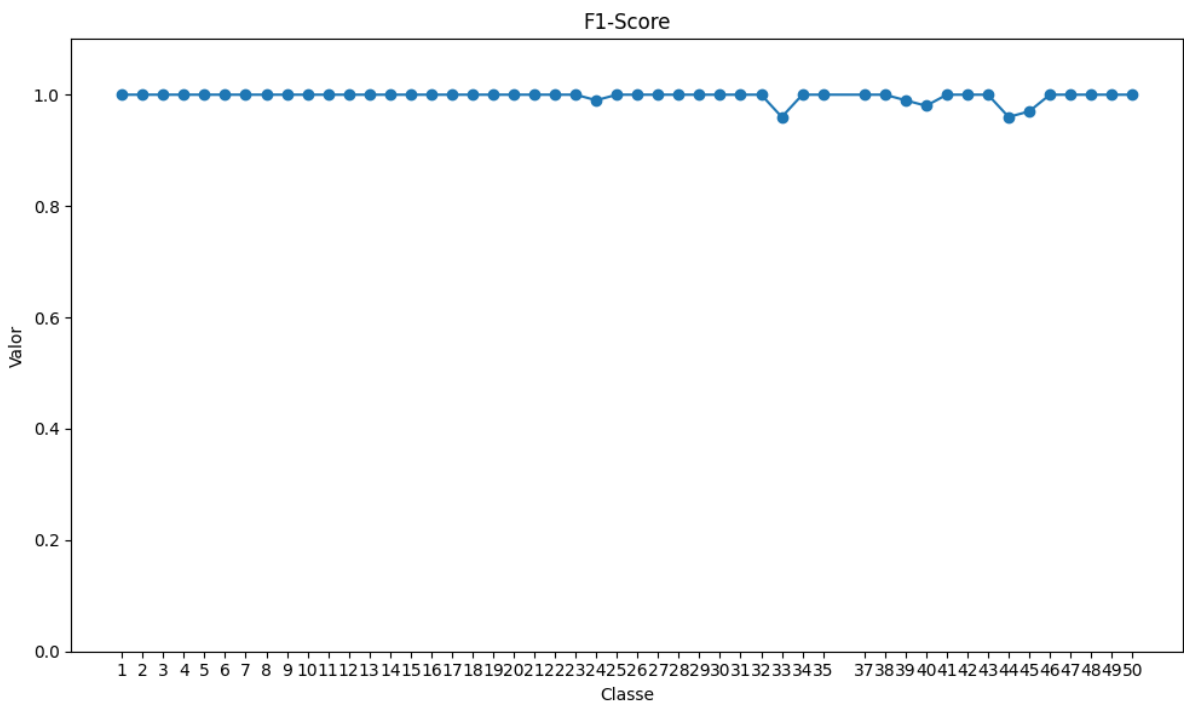


Figura 5.9: Medidas F1 alcançadas por cada classe no Experimento E3.

Capítulo 6

Conclusão

O presente trabalho busca avaliar a aplicabilidade do reconhecimento de bovinos em imagens utilizando Redes Neurais Convolucionais (CNNs), apresentando uma solução tecnológica que atenda à demanda de rastreabilidade de gado presente no setor pecuário, oferecendo uma alternativa a métodos tradicionais de identificação, como brincos auriculares, tatuagens e marcações de ferro.

Para alcançar esse propósito, propõe-se a implementação de uma arquitetura baseada na DenseNet201 [10], cujos pesos são inicializados pelo treinamento prévio do modelo com o conjunto de dados ImageNet [32]. O treinamento, validação e teste do modelo, foram conduzidos utilizando um conjunto de dados que abrange 49 classes de bovinos da raça Pantaneira, desenvolvido por Weber *et al.* [42] em 2020. O trabalho focou em desenvolver uma arquitetura capaz de reproduzir os resultados promissores obtidos por Weber *et al.* por meio de experimentos iterativos no referido conjunto de dados. Isso incluiu uma cuidadosa seleção e recorte nas imagens presentes no *dataset*, visando um aperfeiçoamento no desempenho do modelo.

A análise dos resultados dos experimentos propostos E1a, E1b, E2 e E3, apresentados no Capítulo 5, fornece uma visão esclarecedora do aprimoramento gradual do modelo de reconhecimento de bovinos. Cada iteração desempenhou um papel crucial no desenvolvimento e refinamento do sistema.

No Experimento E1a, a detecção de *overfitting* foi uma observação valiosa, impulsionando melhorias subsequentes. O Experimento E1b, resultante dessa identificação, trouxe ajustes na arquitetura e a exclusão da classe 51, duplicada no conjunto de dados, culminando em um desempenho mais interessante. As métricas do conjunto de teste, apresentadas na Tabela 5.4, demonstraram uma melhoria substancial, com a função de perda em 0.07125 e uma acurácia de 98.51%.

O Experimento E2 expandiu as melhorias, incorporando uma seleção mais cuidadosa de dados. As métricas no conjunto de teste, exibidas na Tabela 5.5, indicaram uma

resiliência adicional do modelo, com um leve aumento na função de perda para 0.1001, mas mantendo uma alta acurácia de 98.63%.

No Experimento E3, uma abordagem refinada de recorte de imagens foi adotada. Os resultados, apresentados na Tabela 5.6, revelaram um desempenho robusto, com a função de custo em 0.0547 e uma acurácia de 99.73%.

Os experimentos propostos, cada um construindo sobre os aprendizados do anterior, destacam a importância de uma abordagem iterativa e estratégica no desenvolvimento de modelos de aprendizado de máquina. A evolução progressiva das métricas de acurácia e da função de custo valida os ajustes realizados, reforçando a capacidade do modelo de generalizar eficientemente para novos dados e realçando sua potencial aplicabilidade em cenários práticos de reconhecimento de bovinos.

Ainda, nota-se que os resultados obtidos se aproximam com as conquistas de Weber *et al.* discutidos na Seção 3.4. Na pesquisa de Weber *et al.*, o modelo DenseNet201 alcançou uma acurácia de 99,85% no conjunto de teste [42]. Dessa forma, conclui-se que o presente trabalho conseguiu reproduzir, de maneira aproximada, os resultados obtidos por Weber *et al.*, validando a consistência do modelo proposto.

Contudo, apesar desses avanços, é crucial reconhecer os desafios potenciais na aplicação de Redes Neurais Convolucionais (CNNs) no reconhecimento de bovinos. Um desafio significativo é a necessidade de volumes substanciais de dados, especialmente considerando a diversidade nas características físicas entre diferentes raças de bovinos. A obtenção e rotulação de conjuntos de dados representativos podem ser trabalhosas e dispendiosas.

Além disso, futuros trabalhos podem incluir a detecção de múltiplos animais em uma única imagem. Abordar o desafio intrínseco de identificar e localizar diferentes bovinos em cenários complexos proporcionaria uma contribuição significativa para a eficácia prática do sistema. Ainda, explorar técnicas de detecção de objetos pode aprimorar a capacidade do modelo de lidar com situações realistas de pecuária, onde diversas instâncias de bovinos podem coexistir em uma única cena.

Os resultados alcançados, contudo, reforçam que a aplicação de CNNs para o reconhecimento de bovinos se apresenta não apenas como uma solução viável, mas também altamente eficaz. As CNNs mostraram-se capazes de aprender representações complexas e hierárquicas de características visuais, proporcionando uma abordagem promissora para problemas de classificação de imagens, como o reconhecimento de bovinos.

Referências

- [1] IBGE: *Rebanho de Bovinos (Bois e Vacas)*, 2022. <https://www.ibge.gov.br/explica/producao-agropecuaria/bovinos/br>. 1
- [2] Santos, Heloisa Oliveira: *Competitividade da carne bovina brasileira no mercado exterior*. Monografia, Universidade Federal do Rio Grande do Sul, 2023. 1
- [3] Albuquerque, Maria do Socorro Maues e Patrícia Ianella: *Inventário de recursos genéticos animais da Embrapa*. EMBRAPA, 2016. <http://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/1059875>. 1
- [4] Bergamini, Luca, Angelo Porrello, Andrea Capobianco Dondona, Ercole Del Negro, Mauro Mattioli, Nicola D’alterio e Simone Calderara: *Multi-views embedding for cattle re-identification*. Em *2018 14th international conference on signal-image technology & internet-based systems (SITIS)*, páginas 184–191. IEEE, 2018. 1, 21
- [5] Neto, Crespo e Sergio Arthur de La Hidalga: *Reconhecimento de tumores cerebrais utilizando redes neurais convolucionais*. Monografia, Universidade Federal do Pampa, 2017. <https://repositorio.unipampa.edu.br/jspui/handle/rii/1911>. 2
- [6] Badue, Claudine, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz *et al.*: *Self-driving cars: A survey*. *Expert Systems with Applications*, 165:113816, 2021. 2
- [7] Jones, Valerie K: *Voice-activated change: Marketing in the age of artificial intelligence and virtual assistants*. *Journal of Brand Strategy*, 7(3):233–245, 2018. 2
- [8] Brito, Alexandre: *Reconhecimento facial bovino: uma alternativa aos métodos tradicionais de rastreamento*. Monografia, Universidade de Caxias do Sul, 2021. <https://repositorio.ucs.br/11338/9364>. 2, 14, 21
- [9] Lopes, Marcos Aurélio, Lucio Violin Junqueira, Fábio Raphael Pascoti Bruhn, Andreia Alves Demeu e Marilane das Dores Silva: *Technical efficiency and economic viability of different cattle identification methods allowed by the brazilian traceability system*. *Semina: Ciências Agrárias*, 38(1):467–480, 2017. 2
- [10] Huang, Gao, Zhuang Liu, Laurens van der Maaten e Kilian Q. Weinberger: *Densely connected convolutional networks*. Em *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3, 12, 13, 20, 25, 29, 30, 34, 47

- [11] Jordan, Michael I e Tom M Mitchell: *Machine learning: Trends, perspectives, and prospects*. Science, 349(6245):255–260, 2015. 3
- [12] Hu, Guosheng, Yongxin Yang, Dong Yi, Josef Kittler, William Christmas, Stan Z. Li e Timothy Hospedales: *When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition*. Em *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015. 3
- [13] Lecun, Y., L. Bottou, Y. Bengio e P. Haffner: *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 86(11):2278–2324, 1998. 3, 12, 17
- [14] Abdel-Hamid, Ossama, Abdel rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn e Dong Yu: *Convolutional neural networks for speech recognition*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(10):1533–1545, 2014. 3
- [15] Goodfellow, Ian, Yoshua Bengio e Aaron Courville: *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>. 4, 5, 6, 9
- [16] Alpaydin, Ethem: *Introduction to Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 3ª edição, 2014, ISBN 978-0-262-02818-9. 4, 5
- [17] Silva, Josenildo C. e Raimundo Osvaldo Vieira: *Introdução às Redes Neurais Profundas com Python*, capítulo 4. Sociedade Brasileira de Computação, 2022. 7, 8
- [18] Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do e Kaori Togashi: *Convolutional neural networks: an overview and application in radiology*. Insights Imaging, 9(4):611–629, agosto 2018. 7, 14, 15, 16, 17
- [19] Jeczminek, Ernest e Piotr A. Kowalski: *Flattening layer pruning in convolutional neural networks*. Symmetry, 13(7), 2021, ISSN 2073-8994. <https://www.mdpi.com/2073-8994/13/7/1147>. 10
- [20] Fleck, Leandro, Maria Hermínia Ferreira Tavares, Eduardo Eyng, Andrieli Cristina Helmann e Minéia Aparecida de Moares Andrade: *Redes neurais artificiais: Princípios básicos*. Revista Eletrônica Científica Inovação e Tecnologia, 1(13):47–57, 2016. 11
- [21] Spörl, Christiane, Emiliano Castro e Aílton Luchiari: *Aplicação de redes neurais artificiais na construção de modelos de fragilidade ambiental*. Revista do Departamento de Geografia, 21:113–135, 2011. 12
- [22] Jabbar, H e Rafiqul Zaman Khan: *Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)*. Computer Science, Communication and Instrumentation Devices, 70(10.3850):978–981, 2015. 14, 15
- [23] Koehrsen, Will: *Overfitting vs. underfitting: A complete example*. Towards Data Science, 405, 2018. 15
- [24] Reitermanova, Zuzana *et al.*: *Data splitting*. Em *WDS*, volume 10, páginas 31–36. Matfyzpress Prague, 2010. 15, 16

- [25] Mitchell, Tom M: *Machine learning*. McGraw-Hill, 1997. 15
- [26] Academy, Data Science: *Deep Learning Book*. Data Science Academy, 2022. <http://www.deeplearningbook.com.br>. 17, 19
- [27] Zhou, XueFei: *Understanding the convolutional neural networks with gradient descent and backpropagation*. Journal of Physics: Conference Series, 1004(1):012028, apr 2018. <https://dx.doi.org/10.1088/1742-6596/1004/1/012028>. 17
- [28] Brownlee, Jason: *What is the difference between a batch and an epoch in a neural network*. Machine Learning Mastery, 20, 2018. 18
- [29] Kumar, Siddharth Krishna: *On weight initialization in deep neural networks*. arXiv preprint arXiv:1704.08863, 2017. 18
- [30] Neyshabur, Behnam, Hanie Sedghi e Chiyuan Zhang: *What is being transferred in transfer learning?* Advances in neural information processing systems, 33:512–523, 2020. 18
- [31] Cireşan, Dan C, Ueli Meier e Jürgen Schmidhuber: *Transfer learning for latin and chinese characters with deep neural networks*. Em *The 2012 international joint conference on neural networks (IJCNN)*, páginas 1–6. IEEE, 2012. 18
- [32] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein *et al.*: *Imagenet large scale visual recognition challenge*. International journal of computer vision, 115:211–252, 2015. 18, 25, 29, 47
- [33] Huh, Minyoung, Pulkit Agrawal e Alexei A Efros: *What makes imagenet good for transfer learning?* arXiv preprint arXiv:1608.08614, 2016. 18
- [34] Filho, Câmara e José Augusto: *Um guia prático para apoiar tarefas preditivas em Ciência de Dados*. Monografia, Universidade Federal do Ceará, 2022. <http://repositorio.ufc.br/handle/riufc/71031>. 19
- [35] Dalla Rosa, Francisco, Laura Dall’Igna Favretto, Vítor Borba Rodrigues e Nasir G Gharaibeh: *Avaliação do desempenho de redes neurais convolucionais para detecção automática de trincas em pavimentos*. TRANSPORTES, 28(5):267–279, 2020. 19, 20
- [36] Martins, Fernando Marassi e Marcos Aurélio Lopes: *Rastreabilidade bovina no Brasil*. Lavras: UFLA, 2003. 22, 23
- [37] McKean, JD: *The importance of traceability for public health and consumer protection*. Revue Scientifique et Technique-Office International des Épizooties, 20(2):363–369, 2001. 22
- [38] Lopes, Marcos Aurélio e Glauber dos Santos: *Custo da implantação da rastreabilidade em bovinocultura utilizando os diferentes métodos de identificação permitidos pelo SISBOV*. Ciência Animal Brasileira/Brazilian Animal Science, 8(4):657–664, 2007. 23, 24

- [39] Abastecimento, Ministério da Agricultura Pecuária e: *Instrução Normativa no 21, de 2 de abril de 2004*. Diário Oficial da União, 2004. seção 1, p. 1. 23, 24
- [40] Leal, Allan Glauber Anchieta: *Tecnologias para gestão pecuária: um estudo sobre a utilização de sistemas RFID*, 2010. <http://repositorio.unifesspa.edu.br/handle/123456789/203>. 24
- [41] Kumar, Santosh, Sanjay Kumar Singh, Ravi Shankar Singh, Amit Kumar Singh e Shrikant Tiwari: *Real-time recognition of cattle using animal biometrics*. Journal of Real-Time Image Processing, 13:505–526, 2017. 25
- [42] Weber, Fabricio de Lima, Vanessa Aparecida de Moraes Weber, Geazy Vilharva Menezes, Adair da Silva Oliveira Junior, Daniela Arestides Alves, Marcus Vinicius Morais de Oliveira, Edson Takashi Matsubara, Hemerson Pistori e Urbano Gomes Pinto de Abreu: *Recognition of pantaneira cattle breed using computer vision and convolutional neural networks*. Computers and Electronics in Agriculture, 175:105548, 2020. 25, 26, 27, 31, 47, 48
- [43] He, Kaiming, Xiangyu Zhang, Shaoqing Ren e Jian Sun: *Deep residual learning for image recognition*. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 770–778, 2016. 25
- [44] Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke e Alexander Alemi: *Inception-v4, inception-resnet and the impact of residual connections on learning*. Em *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 25
- [45] Weber, Fabricio de Lima: *Recbov51c*, 2019. <https://data.mendeley.com/datasets/8ysxtyf8p2/1>. 25, 30, 31, 32, 33, 34, 35