



FUNDAÇÃO UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

BRENDA MEDINA DE OLIVEIRA

**Sistema Automatizado para Identificação de Fenótipo Relacionado a
Precocidade e Fertilidade de Fêmea Bovina**

Campo Grande - MS

2022

BRENDA MEDINA DE OLIVEIRA

**Sistema Automatizado para Identificação de Fenótipo Relacionado a
Precocidade e Fertilidade de Fêmea Bovina**

Versão original

Dissertação apresentada ao Programa de Pós-Graduação em Mestrado Profissional em Computação Aplicada como requisito final para obtenção do título de Mestre em Computação Aplicada.

Área de concentração: Tecnologias Computacionais para Agricultura e Pecuária

Orientador: Prof. Dr. Edson Antonio Batista (UFMS)

Coorientador: Prof. Dr. Eriklis Nogueira (EMBRAPA)

Campo Grande - MS

2022

À Deus, minha família, amigos e professores que sempre me incentivaram a conquistar meus objetivos.

Agradecimentos

Um trabalho de mestrado é uma longa viagem, que inclui uma trajetória permeada por inúmeros desafios, tristezas, incertezas, alegrias e muitos percalços pelo caminho, mas apesar do processo solitário a que qualquer investigador está destinado, reúne contributos de várias pessoas, indispensáveis para encontrar o melhor rumo em cada momento da caminhada. Trilhar este caminho só foi possível com o apoio, energia e força de várias pessoas, a quem dedico especialmente este projeto de vida.

Por fim, o meu profundo e sentido agradecimento a todas as pessoas que contribuíram para a concretização desta dissertação, estimulando-me intelectual e emocionalmente.

"Um computador pode ser chamado de "inteligente" se conseguir fazer uma pessoa pensar que é humana."
(Alan Turing)

Resumo

Oliveira, Brenda Medina. **Sistema Automatizado para Identificação de Fenótipo Relacionado a Precocidade e Fertilidade de Fêmea Bovina**. 2022. 51 f. Dissertação (Mestrado em Computação Aplicada) – Faculdade de Computação, Universidade Federal do Estado de Mato Grosso do Sul, Campo Grande, 2022.

A identificação da precocidade bovina, características reprodutivas, do mesmo são de extrema importância, pois com o diagnóstico correto é possível atingir um aumento de produtividade e conseqüentemente a lucratividade do setor. Porém, este processo é realizado subjetivamente, o que pode acarretar inexatidão, e diferentes resultados de acordo com o avaliador. Com intuito de apresentar uma solução automatizada e com maior objetividade, neste trabalho desenvolveu-se uma metodologia baseada em escores visuais para identificar a precocidade de fêmeas bovinas. A metodologia consiste na captura de imagens do animal, quando transita pelo mangueiro e através de algoritmos de inteligência artificial permite-se identificar sua precocidade a fertilidade. Os algoritmos desenvolvidos permitem a entrada da imagem do bovino e retorna a probabilidade de acurácia e fertilidade na primeira estação reprodutiva. Para realização deste trabalho foram coletadas imagens dos animais na Fazenda Sete Estrelas, que serviram de base para desenvolver o algoritmo. Foram testados o desempenho de 5 algoritmos, que realizam a identificação e classificação de cada imagem de fêmea bovina com intuito de indicar qual técnica é mais indicada para finalidade. Os resultados obtidos através dos algoritmos foram: CNN em conjunto com U-Net, obtiveram uma porcentagem de acertos de 71.42%; O algoritmo Efficientnet obteve 70.46%; Assim como o anterior o algoritmo Twins obteve 70.46%; E o melhor resultado obtido se deu por meio do algoritmo Resnet com 74.92%. Os resultados obtidos são promissores, contudo pode-se melhorar, com a adequada captação das imagens dos animais e aumentando o banco de imagens, apresentando um grande potencial para a resolução do problema.

Palavras-chaves: Identificação de fertilidade bovina. Precocidade de bovinos fêmeas. Algoritmo CNN. Algoritmo U-Net. Algoritmo Resnet. Algoritmo Twins. Algoritmo Efficientnet.

Abstract

Oliveira, Brenda Medina. **Automated System for Identification of Precocity and Fertility Related Phenotype of Female Cattle**. 2022. 51 f. Dissertation (Master in Applied Computing) - Faculty of Computing, Federal University of the State of Mato Grosso do Sul, Campo Grande, 2022.

The identification of bovine precocity, reproductive characteristics are extremely important, because with the correct diagnosis it is possible to achieve an increase in productivity and consequently the profitability of the sector. However, this process is performed subjectively, which can lead to inaccuracy, and different results according to the evaluator. In order to present an automated solution with greater objectivity, this paper developed a methodology based on visual scores to identify the precocity of bovine females. The methodology consists in capturing images of the animal when it passes through the mango tree and through artificial intelligence algorithms it is possible to identify its precocity and fertility. The developed algorithms allow the input of the bovine image and return the probability of accuracy and fertility in the first reproductive season. To accomplish this work, images of the animals were collected from the Sete Estrelas Farm, which served as a basis to develop the algorithm. The performance of 5 algorithms that perform the identification and classification of each female bovine image were tested in order to indicate which technique is more suitable for the purpose. The results obtained through the algorithms were: RCNN in conjunction with U-Net, obtained a hit percentage of 71.42%; the Efficientnet algorithm obtained 70.46%; likewise the Twins algorithm obtained 70.46%; and the best result obtained was through the Resnet algorithm with 74.92%. The results obtained are promising, but it can be improved with the adequate capture of images of the animals and increasing the image bank, presenting a great potential for solving the problem.

Key words: Identification of bovine fertility. Female bovine precocity. Mask RCNN Algorithm. U-Net Algorithm. Resnet Algorithm. Twins Algorithm. Efficientnet algorithm.

Lista de figuras

Figura 1 – Representação de identificação de características do método CPMU, onde as setas na cor amarela indicam a conformação, as verdes, a precocidade, azuis, musculosidade e vermelho, umbigo. Fonte: Adaptado (BeefPoint, 2004)	16
Figura 2 – Representação de identificação de características do método EPMURAS, onde as setas em vermelho são: estrutura corporal, amarela: precocidade, azul: musculosidade, verde: umbigo, e roxo: raça, marrom: aprumos e laranja: sexualidade, que não possuem setas indicativas, pois não inferem em notas. Fonte: Adaptado (FILHO et al., 2015)	17
Figura 3 – Representação do comportamento da técnica de deep learning. Fonte: Adaptado (DRASS et al., 2021).	21
Figura 4 – Representação da operação de convolução: (a) imagem com dimensões de 7x5; (b) imagem de dimensões 3x3; (c) características resultantes do processamento convolucional. Fonte: (SILVA, 2019).	22
Figura 5 – Representação da arquitetura de uma CNN. Fonte: (VARGAS; PAES; VASCONCELOS, 2016).	23
Figura 6 – Representação da segmentação semântica. Fonte adaptado: (LI; JOHNSON; YEUNG, 2017).	24
Figura 7 – Representação da arquitetura Unet. Fonte adaptado: (GEORGIEV; ASENOV, 2018).	24
Figura 8 – Escala do modelo. (a) é um exemplo de rede de base; (b) - (d) são escalas que aumentam apenas uma dimensão da largura, profundidade ou resolução da rede. (e) é o método proposto de escala composto que dimensiona uniformemente as três dimensões com uma relação fixa. Fonte: Adaptado (TAN; LE, 2019).	25
Figura 9 – Representação da aprendizagem residual, bloco de construção. Fonte: Adaptado (HE et al., 2016b)	26
Figura 10 – Representação da estrutura do Twins. Fonte: Adaptado (DOSOVITSKIY et al., 2020).	27

Figura 11 – Padrão de imagem da captura da lateral do animal. Fonte: (Elaborado pelo autor).	30
Figura 12 – Fluxograma do desenvolvimento dos algoritmos. Fonte: Elaborado pelo autor.	31
Figura 13 – Resultado da segmentação semântica, do banco de imagens. Fonte: Elaborado pelo autor.	33
Figura 14 – Representação do comportamento da técnica Cross Validation K Fold. Fonte adaptado: (MAREDDIA, 2020).	34
Figura 15 – Representação da divisão do banco de imagens, seguido o método de Cross Validation. Fonte: Elaborado pelo autor.	35
Figura 16 – Logo SystemPrecFer	39
Figura 17 – Apresentação da tela inicial do web-site. Fonte: Elaborado pelo autor.	40
Figura 18 – Apresentação da tela de instruções, sobre o funcionamento do web-site. Fonte: Elaborado pelo autor.	40
Figura 19 – Resultado do algoritmo treinado e testado na plataforma colab	41
Figura 20 – Resultado do algoritmo treinado e testado na plataforma colab	42
Figura 21 – Melhor resultado obtido pelo algoritmo Resnet. Fonte: Elaborado pelo autor.	44
Figura 22 – Melhor resultado obtido pelo algoritmo Twins. Fonte: Elaborado pelo autor.	44

Lista de tabelas

Tabela 1 – Apresenta-se a metodologia utilizada na pesquisa para revisão bibliográfica.	28
Tabela 2 – Resultado do algoritmo CNN.	41
Tabela 3 – Resultados dos algoritmos, Efficientnet, Resnet e Twins, e as médias e erros.	42
Tabela 4 – Resultado de todos os algoritmos estudados, com relação a acurácia. .	43

Lista de Algoritmos

3.1	Algoritmo para identificação da mascara referente ao item 'bovino' - U-Net .	32
3.2	Algoritmo CNN	32
3.3	Algoritmo de configuração do dataset	35
3.4	Algoritmo de configuração do dataset	36
3.5	Configuração do algoritmo Efficientnet	36
3.6	Configuração do algoritmo Resnet	37
3.7	Configuração do algoritmo Twins	37

Lista de abreviaturas e siglas

CPMU	Conformação, precocidade, musculosidade, umbigo
EPMURAS	Estrutura corporal, musculosidade, umbigo, característica racial, aprumo, sexualidade
MERCOS	Musculosidade, estrutura física, raça, conformação, ônfalo, sexualidade
PGP	Provas de ganho em peso a pasto
PMGZ	Programa de melhoramento genético de zebuínos
IA	Inteligencia artificial
CNN	Rede neural convolucional
ConvNets	Redes neurais convolucionais
IATF	Inseminação artificial por tempo fixo
SystemPrecFer	Sistema de precocidade e fertilidade

Sumário

1	Introdução	13
1.1	<i>Motivação e Justificativa</i>	14
1.2	<i>Objetivo</i>	14
1.2.1	Objetivos Específicos	14
1.3	<i>Organização do texto</i>	15
2	Revisão Bibliográfica	16
2.1	<i>Escores visuais</i>	16
2.2	<i>Precocidade bovina</i>	19
2.3	<i>Deep Learning</i>	19
2.3.1	Rede Neural	21
2.3.2	U-Net	23
2.3.3	EfficientNet	24
2.3.4	ResNet	25
2.3.5	Twins	27
2.4	<i>Trabalhos relacionados</i>	28
3	Materiais e Métodos	30
3.1	<i>Coleta de dados</i>	30
3.2	<i>Algoritmos</i>	31
3.3	<i>Proposta do Front-end</i>	39
4	Resultados	41
5	Discussão	43
6	Conclusões e Considerações Finais	46
6.1	<i>Limitações</i>	46
6.2	<i>Trabalhos Futuros</i>	47
	REFERÊNCIAS	48

1 Introdução

A crescente evolução tecnológica, provoca em todas as áreas a necessidade de atualizações, com intuito de se tornarem eficientes e sustentáveis. Segundo (LASI et al., 2014), o século 21 inicia com a quarta revolução industrial, similarmente denominada indústria 4.0. (DRATH; HORCH, 2014), a qual caracteriza-se principalmente na utilização da internet das coisas e da inteligência artificial, de forma a integrar constantemente o ser humano com a máquina.

O âmbito da pecuária, foi igualmente atingido por esta evolução. Conforme (MÁS-STRUHÁ; LEITE, 2017), foi desenvolvido o termo pecuária 4.0, o qual visa aumentar a produtividade e a qualidade dos produtos, agregar serviços na cadeia produtiva e proporcionar rastreabilidade do processo de produção. O termo advém da indústria 4.0, o qual possui pilares sólidos com a sustentabilidade, confortabilidade dos animais e agilidade na comercialização. De acordo com (CHIARI LUCIMARA, 2020), apresentou uma nova perspectiva de agir no campo, pois não são produzidos apenas produtos e sim serviços, almejando a simplificação do uso da tecnologia para o homem do campo.

A presente revolução desperta expectativas nos setores de produção alimentar, os quais são: agricultura, mineração, pesca, pecuária, extrativismo vegetal e caça. (FAO, 2009), afirma que existem contenções de expansão territorial para produção agropecuária, o que afeta diretamente a ampliação da produtividade. Haja vista que como aponta (ONU, 2012), a população mundial atingirá a marca de 9,5 bilhões em 2050, causando grande preocupação com o futuro alimentar mundial. Neste aspecto, o aumento da produtividade deve ser focado na eficiência territorial, ou seja, produzir o máximo com menor espaço e tempo possível.

O presente trabalho visou a utilização dos conceitos da pecuária 4.0, especificamente através de sistema de algoritmos integrados para análise de imagens. A utilização destes, busca identificar quais bovinos fêmeas apresentam maior propensão a fertilidade. Manualmente realizar esta identificação demanda tempo e recursos como: radiografia, palpação retal, ultrassom, entre outros. O mais indicado, para (FREITAS, 2010), é o uso de ultrassom, por apresentar resultados mais satisfatórios, mesmo não sendo o método com maior completude. Exames que apresentam melhores resultados, devido a logística e onerosidade envolvidos, comumente tornam-se inviáveis. Portanto, buscando contornar a

lacuna apresentada, este trabalho busca oferecer precisão ao pecuarista, na determinação de quais bovinos são precoces a fertilidade, através de uma dinâmica integrada aos ambientes atuais e respeitando a confortabilidade do animal.

Em concordância com (JOSAHKIAN; LUCAS; MACHADO, 2009), para identificação de parâmetros reprodutivos, é imprescindível avaliação de determinados pontos específicos para determinar a fertilidade da fêmea, são estes: Estrutura corporal, precocidade, musculosidade, umbigo, aprumos e sexualidade. Em todas as características são atribuídas notas para avaliar a predisposição fértil. Estas características foram observadas assim que, viabilizado a captura das imagens dos animais. Subsequentemente estas imagens foram treinadas e testadas em algoritmos de inteligência artificial, proporcionando resultados promissores, com média de acurácia por volta dos 70 e 74%.

1.1 Motivação e Justificativa

É possível observar, a importância do trabalho, pois com uso da tecnologia proposta, os pecuaristas adquirem agilidade no processo de identificação da precocidade do animal, obtendo maior precisão comparando-se com outros métodos. Apresenta-se um caráter inovador, pois, não foi encontrado na literatura, solução tecnológica como a proposta do presente trabalho.

1.2 Objetivo

O objetivo do presente trabalho, foi o desenvolvimento de algoritmos de processamento de imagens, afim de que, estes processamentos auxiliem na tomada de decisão sobre a precocidade e fertilidade de fêmeas bovinas, baseados nos fenótipos apresentados em cada imagem dos animais.

1.2.1 Objetivos Específicos

- Definição de um sistema de monitoramento, para esse tipo de avaliação;
- Elaboração de um padrão de imagens;

- Estudar algoritmos de *Deep Learning*, técnica que proporciona o treinamento de computadores para realizar as tarefas de identificação e classificação dos bovinos nas imagens;
- Comparar os resultados obtidos entre os algoritmos de *Deep Learning* e indicar o melhor a ser utilizado.

1.3 Organização do texto

O Capítulo 2 apresenta os principais conceitos que serviram de base para o desenvolvimento deste trabalho, ou seja, a fundamentação teórica, abordando conceitos da pecuária, escores visuais relacionado à precocidade bovina e algoritmos de *Deep Learning* utilizados para solução do problema. O Capítulo 3 apresenta os detalhes do desenvolvimento do trabalho e a implementação das técnicas computacionais utilizadas. O Capítulo 4, apresenta os resultados obtidos através das técnicas usadas. O Capítulo 5, apresenta a discussão dos resultados e a sugestão sobre qual a melhor técnica a ser utilizada. O Capítulo 6 apresenta as considerações finais e trabalhos futuros.

2 Revisão Bibliográfica

Nesse capítulo, descreve-se sobre escores visuais, servindo como base para representar a precocidade do animal e a técnica de *Deep Learning*, que será usada para desenvolver o algoritmo de detecção otimizada.

2.1 Escores visuais

O trabalho realizado por (FARIA et al., 2007), apresenta as três maneiras de avaliar animais através de escores corporais, os quais são atribuir nota a determinada área do animal, Figura 1 é apresentado as demarcações de áreas a serem atribuídas notas. Estes métodos são realizados de forma que seja mais viável a avaliação do animal.

A primeira maneira apresentada por (FARIA et al., 2007), trata-se do CPMU, realizado pela empresa GenSys, onde cada inicial da sigla trata de uma característica a ser analisada, sendo:

- (C) Conformação, profundidade, largura e aparência do animal;
- (P) Precocidade, capacidade de armazenamento de gordura;
- (M) Musculosidade, massa muscular contida;
- (U) Umbigo, apresenta o tamanho e posicionamento.

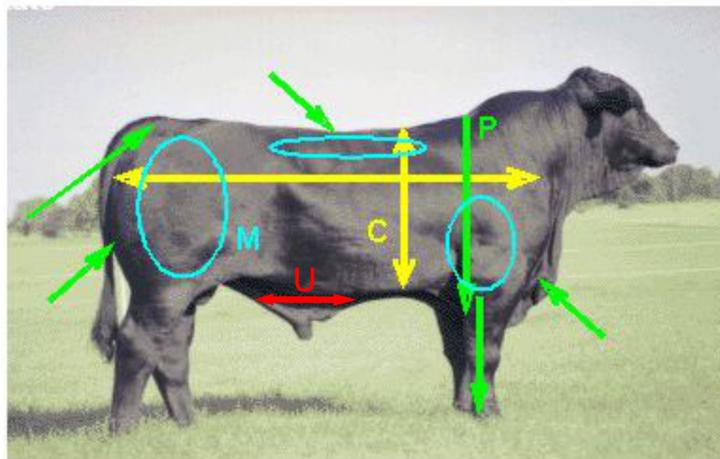


Figura 1 – Representação de identificação de características do método CPMU, onde as setas na cor amarela indicam a conformação, as verdes, a precocidade, azuis, musculosidade e vermelho, umbigo. Fonte: Adaptado (BeefPoint, 2004)

O segundo método apresentado pelo trabalho citado, trata do EPMURAS, assim como o citado anteriormente este analisa o animal por notas atribuídas, o significado da sigla é:

- (E) Estrutura corporal, observa-se a área que o animal atinge, comprimento e profundidade das costelas;
- (P) Precocidade, o desenho do animal corresponda o qual demonstre depositar gordura mais precocemente;
- (M) Musculosidade, massa muscular;
- (U) Umbigo, tamanho e posicionamento;
- (R) Caracterização racial, todas as características previstas na raça;
- (A) Aprumos, são avaliados proporções, direções, angulações e articulações;
- (S) Sexualidade, observa-se a acentuação da idade do animal com relação seu sexo.

Como aponta (FILHO et al., 2015), este método diferentemente do CPMU, engloba mais características a serem analisadas, variando de 1 à 6, onde os escores E, P e M são avaliados em grupo contemporâneo, enquanto que R, A, S são analisados com base em referencia de padrões pré-definidos e por fim U também é atribuído nota em referência definidas anteriormente, como demonstra a Figura 2.

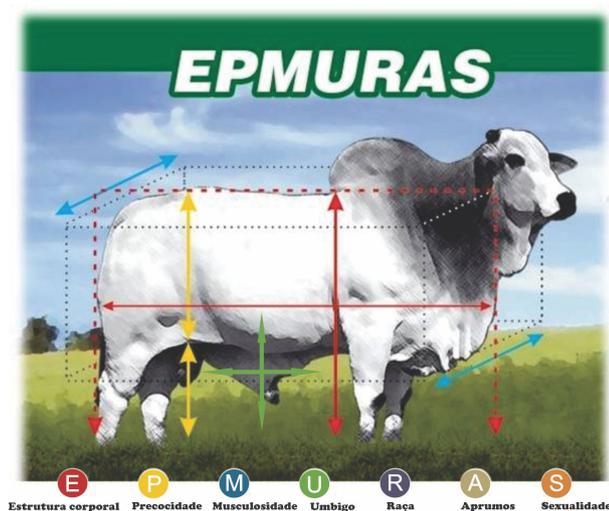


Figura 2 – Representação de identificação de características do método EPMURAS, onde as setas em vermelho são: estrutura corporal, amarela: precocidade, azul: musculosidade, verde: umbigo, e roxo: raça, marrom: aprumos e laranja: sexualidade, que não possuem setas indicativas, pois não inferem em notas. Fonte: Adaptado (FILHO et al., 2015)

Além disso, existe um outro método bastante similar ao EPMURAS, denominado conformação frigorífica, esta avaliação é realizada após as considerações de fertilidade, as quais foram supracitadas no método do EPMURAS.

De acordo com (ROSA; ABREU, 2007), a conformação frigorífica é constituída por três componentes, sendo estes: estrutura, musculatura e precocidade do acabamento. Onde a estrutura se dá pela caixa do animal, ou seja, comprimento, profundidade e arqueamento das costelas. A musculatura indica a quantidade e a forma da massa muscular e a precocidade aponta a cobertura de gordura. Esta análise de conformação frigorífica é imprescindível, pois é necessário que os animais apresentem boa qualidade com relação a recria e engorda.

O terceiro método apresentado é o denomina-se MERCOS, tratando das seguintes características:

- (M) Musculosidade, distribuição muscular;
- (E) Estrutura física, integridade do animal ao tamanho;
- (R) Raça, características que descrevam o animal sendo da raça em questão;
- (C) Conformação, o esqueleto como um todo do animal;
- (O) Ônfalo, trata do umbigo, como está posicionado e o comprimento;
- (S) Aspectos sexuais, funcionalidade genitália coma idade do animal.

MERCOS foi desenvolvido pelo programa nelore Brasil, buscando identificar animais que obtenham características que interferem em sua economia, assim como composição de peso do animal. Os escores recebem de 1 à 5. É realizada a comparação entre o grupo contemporâneo, dentre estes o animal que apresentar escore 3, passará a ser referencia ao restante. Ou seja, é um método comparativo e as notas são relativas.

Os métodos apresentados são as maneiras utilizadas para atribuir notas as características visuais do gado, que resultam no grau de fertilidade do bovino. É possível observar a importância destes, pois baseado nessas atribuições que objetiva-se, o trabalho proposto, trata de uma otimização dos métodos conhecidos. Estas avaliações serão feitas através de um algoritmo de visão computacional.

(COSTA; C.H.C., 2013), realizaram um trabalho baseado em escores visuais, buscando identificar os animais com melhores avaliações e submeter estes a um programa de Provas de Ganho em Peso a Pasto (PGP) do Programa de Melhoramento Genético de Zebuínos (PMGZ).

As características identificadas em cada método, são essenciais. Pois, através destas, é possível inferir a precocidade de determinado bovino a fertilidade.

2.2 Precocidade bovina

Nas palavras de (BRUMATTI et al., 2011), os aspectos reprodutivos são um dos principais pontos chaves para o desenvolvimento econômico em produção de gado de corte. Pois, foi constatado que os animais com maior precocidade a ser fértil, possui maior disposição a demandas de venda ou até mesmo capacidade seletiva.

O trabalho de (TEIXEIRA; ALBUQUERQUE; FRIES, 2002), inclusive buscou evidenciar este fato. De forma que a pesquisa apresenta três maneiras de produção, afim de comparar qual a mais eficiente. Os sistemas em que a produção de novilhas nelore foram testadas são de reprodução aos 15, 18 e 27 meses de idade. Durante o processo houveram novilhas descartadas, pois não obtinham fenótipos de precocidade, e as restantes apresentaram o resultado esperado, constatando o objetivo do trabalho.

Um dos trabalhos pioneiros nesse aspecto de precocidade bovina, com o título de Fatores genéticos e ambientais sobre a probabilidade de prenhez precoce em bovinos Caracu por (MATTAR et al., 2007), onde apresentaram que para se obter a características de precocidade, é preciso submeter o animal a monta, com probabilidade de prenhez aos 14 meses de idade. Para constatar os resultados obtidos foi utilizada uma análise *bayesiana*, e a média foi de 0,13. Embora a média baixa de herdabilidade, é possível observar a influência deste fator na precocidade bovina.

Portanto, as distintas formas de produção e seleção são de suma importância, para sua eficiência econômica (PIRES et al., 2015). Tendo em vista estes aspectos, notamos os principais fatores que devem ser levado em conta, como crescimento, peso corporal possuem maior ênfase nos critérios de seleção.

2.3 Deep Learning

A Inteligência Artificial(IA) busca representar comportamento inteligente, baseado em conhecimento empírico do ser humano. Buscando identificar e reconhecer padrões para treinamento de dispositivos eletrônicos. Na literatura afirmam-se que além da IA

(TEIXEIRA, 2019) buscar reproduzir o comportamento da mente humana, através da IA busca-se compreender seu funcionamento.

A técnica utilizada neste trabalho, trata de uma sub-área da IA, conhecida como *Deep Learning*. Conforme (PONTI; COSTA, 2018) *Deep Learning*, objetiva encontrar um modelo, usando dados de entrada e um método para conduzir o processamento desses dados, afim de retornar como saída o resultado adequado para solução do problema, em questão.

Como aponta (ZHANG, 2020) os modelos existentes são divididos em: Aprendizado supervisionado: consiste em um conjunto de dados rotulados, onde são inseridas as entradas de dados e espera-se um conjunto de saída específica. Ou seja, trata-se de um mapeamento da entrada a saída de dados. Aprendizado não supervisionado: este consiste em um conjunto de dados sem rótulos, em que é preciso encontrar a solução, sem auxílio de rótulos de dados de entrada. Aprendizado semi-supervisionado: dado um conjunto de dados, uma pequena parte destes são rotulados e o restante, maior parte não possuem rótulos. Pode-se haver modelos distintos neste modelo, decorrente da forma em que os dados são rotulados. Aprendizado por reforço: o conjunto de dados são feedback de um agente de IA em um ambiente, onde esse retorno para o sistema acumule experiência e os sistema aprenda a realizar determinada atividade.

A técnica de *Deep Learning* ou aprendizado profundo, advém da técnica conhecida como redes neurais, onde o processamento dos dados ocorrem em camadas da transformação não linear dos dados de entrada para saída. (SCHMIDHUBER, 2015) afirma que, estas camadas de processamento formam hierarquia de conhecimento, de forma que as camadas iniciais são responsáveis por extrair características gerais e subsequentemente as camadas posteriores tem o objetivo de aprofundar-se mais na extração das características de uma imagem, por exemplo.

Na Figura 3, é exposto o funcionamento de *Deep Learning*, onde é possível observar que existe uma grande quantidade de camadas de processamento, sendo esta a principal diferença entre os algoritmos de *Deep Learning* e redes neurais simples, os quais possuem apenas os dados de entrada, processamento e saída, não contendo inúmeras camadas de processamento.

Estas várias camadas ocultas, que permite o algoritmo executar determinada tarefa repetidamente, buscando a cada processamento ajustar-se a fim de melhorar o resultado esperado. Ou seja, para (PONTI; COSTA, 2018) redes neurais simples necessitam de

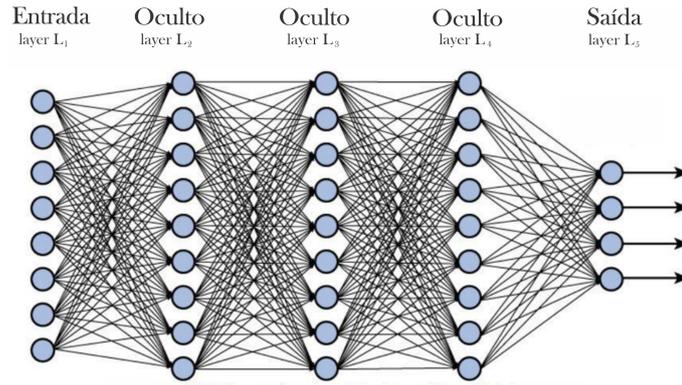


Figura 3 – Representação do comportamento da técnica de deep learning. Fonte: Adaptado (DRASS et al., 2021).

quantidade maior de informações para realizar o processamento dos dados, enquanto que utilizando aprendizado profundo, não é necessário nenhum tipo de informação sobre os recursos a serem utilizados, pois realiza-se o ajuste sem interferência direta.

Como indica (PACHECO; PEREIRA, 2018), esse processamento é realizado através de cálculo de pesos em cada camada. Exemplifica-se na Equação 1, onde admite-se um conjunto de parâmetros, como uma matriz WL , referente a cada fl , sendo que f trata das técnicas usadas por *Deep Learning* e l refere-se as camadas, sendo então uma composição de funções:

$$fl : (...f2(f1(x1, W1); W2)...), WL \quad (1)$$

Sendo $x1$ representa os dados de entrada, cada função faz o uso do próprio conjunto $W1$ de parâmetros e sua saída passa para a função subsequente. Vale ressaltar que esta função é composta por L funções, ou seja, camadas. Este é o conceito amplo de *Deep Learning*, e como funciona seu comportamento. Porém, nas seções seguintes 2.4, 2.5, 2.6, 2.7 e 2,8, são apresentadas distintas maneiras de aprendizado, contudo, advindas do conceito amplo de *Deep Learning*.

2.3.1 Rede Neural

A rede neural convolucional(CNN's), é uma técnica que advém do aprendizado profundo, resultando em uma técnica bastante conhecida pois, além de realizar detecção de objetos, realiza a classificação dos mesmos, os quais essas redes podem extrair descritores da imagem a ser analisada. (SHAHEEN; VERMA; ASAFUDDOULA, 2016).

O funcionamento de uma CNN, é bastante semelhante as redes neurais tradicionais, porém de outro modo utiliza-se núcleos, com objetivo de extrair características de determinada imagem. As convoluções utilizadas reduzem as dimensões buscando, a representação geométrica, bem como linhas e colunas, logo após, as camadas convolucionais geram uma imagem menor, (ERICKSON et al., 2017). Para melhor entendimento este processo é apresentado a Figura 4.

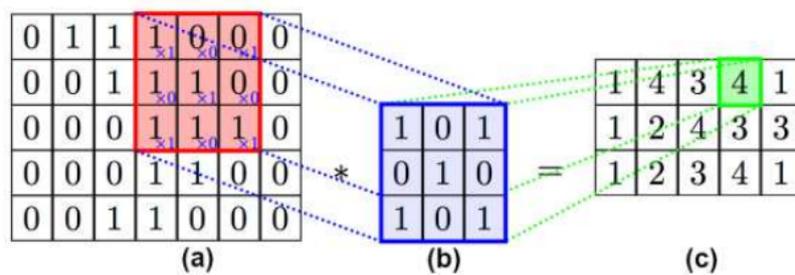


Figura 4 – Representação da operação de convolução: (a) imagem com dimensões de 7x5; (b) imagem de dimensões 3x3; (c) características resultantes do processamento convolucional. Fonte: (SILVA, 2019).

Os núcleos anteriormente citados, podem também ser chamados de filtros. Estes por sua vez, buscam locomover sobre as camadas da imagem, realizando a convolução, por resultado espera-se que cada filtro, crie um mapa de características bidimensionais. (LECUN; BENGIO; HINTON, 2015) Cada um destes filtros almeja calcular distintas características, por exemplo: nitidez, detecção de borda, ou seja, quanto maior a quantidade de filtros, maior a extração de peculiaridades.

Um ponto de extrema importância é a camada de *pooling*, que em suma, realiza a simplificação do mapa das características geradas pelas camadas convolucionais (NIELSEN et al., 2015). Concisamente a camada de *pooling* opera agrupando os dados, ou seja, a entrada é dividida em 4x4, por exemplo e de cada, é selecionado um valor que represente as métricas envolvidas. (PENHA et al., 2018) Existem outras camadas deste processamento como, *max-pooling*, onde o maior valor representa a métrica de cada saída, similar a isto existe a técnica de *average-pooling* que por sua vez a métrica se dá pelo valor médio de cada saída.

Além disso, existem as camadas totalmente conectadas, onde todos neurônios da camada anterior são conectados aos neurônios de saída, os quais representam as classes a serem classificadas (ARAÚJO, 2018). É neste ponto que tudo o que foi aprendido é combinado, afim de gerar os descritores de características. Geralmente utiliza-se o *softmax*, que classifica a saída. Essa classificação permite indicar a possibilidade de uma imagem pertencer a determinada classe. (HIJAZI et al., 2015)

Na Figura 5, é apresentado o funcionamento completo de uma CNN, com todas camadas que anteriormente foram citadas.

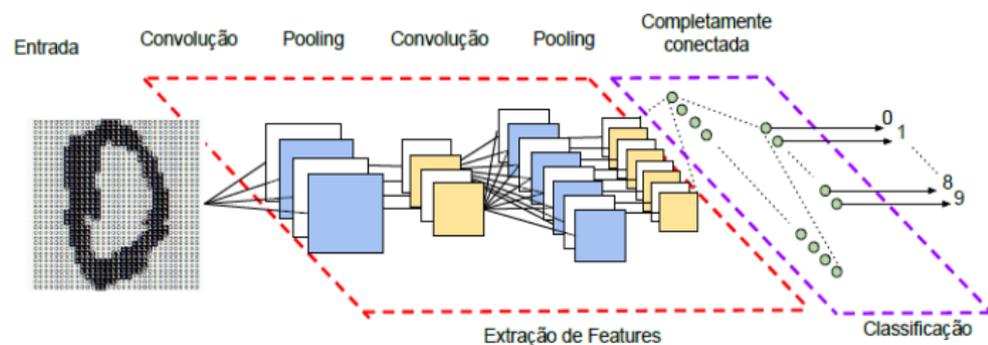


Figura 5 – Representação da arquitetura de uma CNN. Fonte: (VARGAS; PAES; VASCONCELOS, 2016).

2.3.2 U-Net

Segundo (RONNEBERGER; FISCHER; BROX, 2015), trata-se de uma rede de segmentação semântica, elas diferem das redes de classificação, pelo fato de estas, gerarem uma saída com rótulo único para a imagem. Distintamente as redes semânticas buscam rotular não o todo, mas cada pixel da imagem. Isso gera um limite exato, de cada objeto na imagem. Na Figura 6, este processamento fica evidenciado.

Ainda segundo (RONNEBERGER; FISCHER; BROX, 2015), é necessário seguir etapas para a utilização desta segmentação semântica, as quais são, etapas de compressão de sequencia, até que se chegue em um ponto que a descompressão inicia, sendo que para isto é preciso ser realizada a concatenação de dados anteriores.

Na Figura 7, é apresentada a arquitetura da Unet.



Figura 6 – Representação da segmentação semântica. Fonte adaptado: (LI; JOHNSON; YEUNG, 2017).

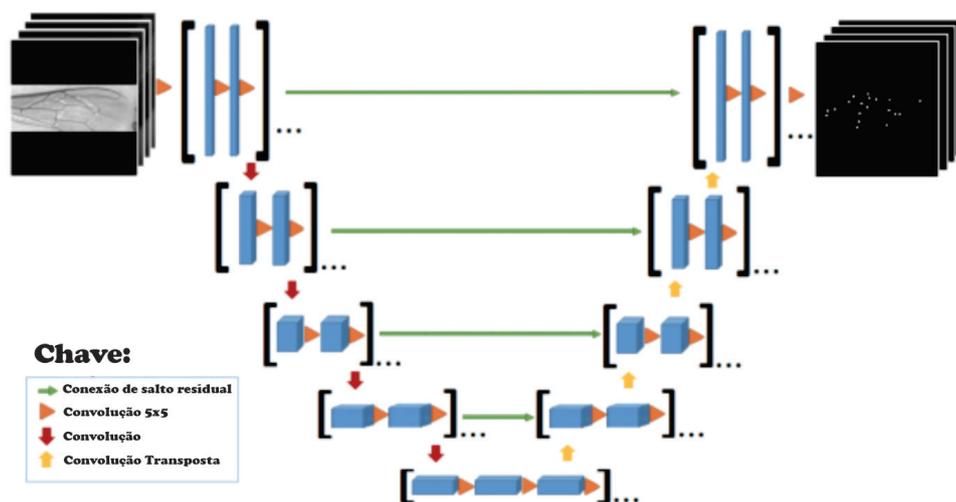


Figura 7 – Representação da arquitetura U-Net. Fonte adaptado: (GEORGIEV; ASENOV, 2018).

2.3.3 EfficientNet

Para (TAN; LE, 2019) trata de um tipo de rede neural convolucional, esta técnica amplia sua precisão a medida que obtêm-se maiores recursos.

A ampliação das ConvNets (Redes neurais convolucionais), é amplamente utilizado, porém a proposta do método EfficientNet, busca além de ampliar a rede pela profundidade como o trabalho de (HE et al., 2016a), por largura como propõe (HAO; YUHANG; LIAN, 2022) ou por resolução de rede como dispõe o trabalho de (HU; SHEN; SUN, 2018). A correlação de todos os métodos anteriores citados, resulta na proposta da técnica aqui

analisada, onde busca-se escalar o todo, realizando o chamado escalonamento composto, largura, profundidade e resolução da rede.

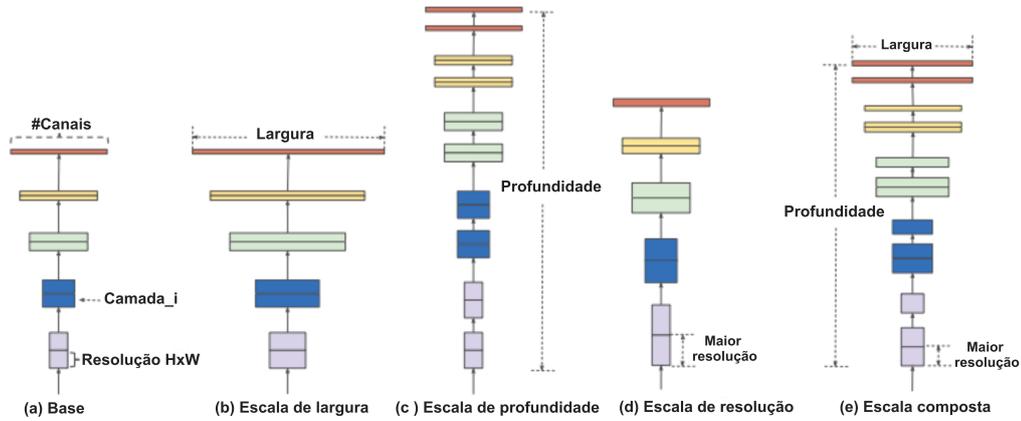


Figura 8 – Escala do modelo. (a) é um exemplo de rede de base; (b) - (d) são escalas que aumentam apenas uma dimensão da largura, profundidade ou resolução da rede. (e) é o método proposto de escala composto que dimensiona uniformemente as três dimensões com uma relação fixa. Fonte: Adaptado (TAN; LE, 2019).

Na Figura 8, mostra-se a diferença entre os métodos convencionais e o proposto. Tendo isto em vista, o método de escala composto, infere que condicionalmente uma imagem de entrada for maior, necessitará mais camadas, aumentando assim seu campo receptivo e mais canais para realizar a granulação mais fina na imagem.

(TAN; LE, 2019) em suma este método objetifica o equilíbrio entre largura, profundidade e resolução da rede, baseado nisto o EfficientNet é capaz de escalonar eficientemente superando a precisão de ponta com uma ordem de magnitude de menos parâmetros.

(TAN; LE, 2019) Após o dimensionamento, a arquitetura neural com multi-objetivo busca otimizar tanto a precisão quanto os FLOPS, usando o mesmo espaço de busca, sendo $ACCm^X [FLOPS(m)/T]^w$, onde $ACCm [FLOPS(m)]$ denotam acurácia e m, T FLOPS alvo e $w = 0.07$ é um hiperparâmetro para controlar o trade-off entre precisão e FLOPS.

2.3.4 ResNet

Refere-se à uma técnica de redes neurais convolucionais profundas. Para (ZEILER; FERGUS, 2014) redes deste tipo interligam características em níveis, sendo estes baixo, médio e alto, além disso possui classificadores em camadas, os níveis de características, podem ser melhorados através do número de camadas. A profundidade tem se tornando um

ponto extremamente importante no desenvolvimento desta técnica como diz, (HE et al., 2015).

Quanto maior a profundidade das redes, estas tem maior capacidade de convergir ao problema de degradação, pois com sua profundidade crescente, a precisão torna-se saturada, levando a degradação. Contudo esta degradação não é causada por *overfitting* e como aponta (HE; SUN, 2015), acrescentando mais camadas ocasiona um erro de treinamento mais alto. Esta degradação infere que nem sempre e nem todos são simples de otimizar.

Como diz, (HE et al., 2016b) existe uma solução para este problema, sendo uma arquitetura mais rasa e sua contraparte mais profunda, a qual acrescenta-se mais camada a ela, ou seja, as camadas adicionais são mapeamento de identidade e as outras são as copias do modelo raso treinado. Resultando em um modelo que não irá produzir erro maior que seu homólogo mais raso, assim como demonstra a Figura 9.

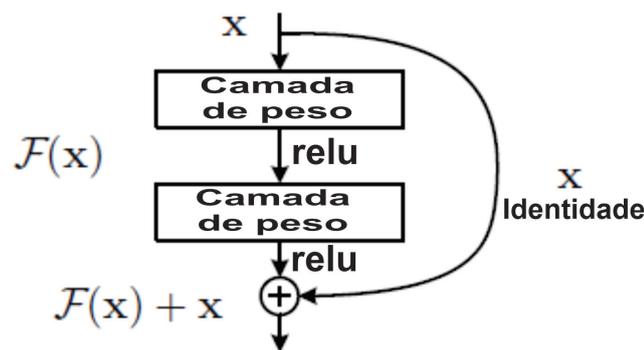


Figura 9 – Representação da aprendizagem residual, bloco de construção. Fonte: Adaptado (HE et al., 2016b)

Para atingir o objetivo de introduzir a aprendizagem residual profunda, seguindo os passos do trabalho desenvolvido por (HE et al., 2016a). Ao contrário de pressupor que as camadas justaponham-se subjacente ao desejado, foi induzido explicitamente estas camadas encaixarem um mapeamento residual.

Sendo que denota-se $H(x)$ deixa-se as camadas não lineares fit outro mapeamento $F(x) := H(x) - x$. O qual seu mapeamento original se dá por, $F(x) + x$, pois estima-se que realizar a otimização do mapeamento residual seja mais compreensível, do que realizar a otimização do mapeamento original.

2.3.5 Twins

Nas palavras de (DOSOVITSKIY et al., 2020) a técnica Twins, concerne a transformadores de visão, os quais estão recebendo rigoroso interesse acadêmico, pois apresentam flexibilidade e facilidade na modelagem de dependências de longo alcance em atividades visão, possuem trajetória menos indutiva, além disso podem facilmente processar informações de entrada de múltiplas categorias, imagens, vídeos, e textos. Tornando-se uma alternativa ao uso das CNN's.

Existem ainda problemas relacionados a esta técnica, o maior se dá ao fato de a complexidade computacional ser muito alta, ocasionada pela operação de auto-atenção espacial nos transformadores, estes ampliam quadraticamente no número de pixels da imagem de entrada. Segundo (LIU et al., 2021), a alternativa para contornar isto é a auto atenção agrupada localmente, assim a entrada é agrupada em janelas não sobrepostas e a auto-atenção é computada apenas dentro de cada subjanela.

Esta alternativa apresentada, pode solucionar porém, ainda assim falta conectividade entre distintas janelas, causando um campo receptivo limitado. Como apresenta (CHEN et al., 2017), um campo receptivo grande é essencial para uma ótima atuação, especialmente para atividades de previsão densa. Para contornar este ponto, dois métodos são apresentados, o primeiro desenvolvido (LIU et al., 2021) sugere que o processamento da janela se desloque para solucionar o problema, o qual o limite local da janela em questão sejam gradativamente deslocados a medida que a rede avança. A segunda solução é proposta por (WANG et al., 2021), onde cada consulta computa a atenção com uma variante da subamostra dos tokens de entrada. Na Figura 10, apresenta como é o comportamento do algoritmo Twins.

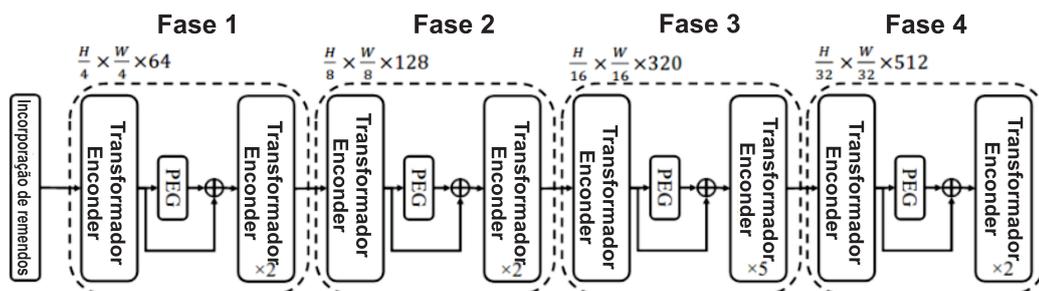


Figura 10 – Representação da estrutura do Twins. Fonte: Adaptado (DOSOVITSKIY et al., 2020).

2.4 Trabalhos relacionados

Nessa seção, são apresentados alguns trabalhos usando a técnicas de inteligência artificial, voltado para pecuária e agricultura. Na busca por selecionar trabalhos mais adequados com o tema proposto, foi realizada uma revisão bibliográfica conforme os passos apresentados por (NEIVA; SILVA, 2016). Sendo o primeiro passo, definição de palavra-chave, para isto foi analisado todas as questões de pesquisa.

Bases de busca foram definidas, estas são: IEEE Xplore, Google Acadêmico e Embrapa. Os resultados encontrados foram:

	S1	S2	S3	S4	Selecionados	Usados
IEEE	52.858	3.078	3.655	179	24	5
Google acadêmico	4.780.000	18.700	81.000	20.300	80	10
Embrapa	1.800	3.150	270	921	55	21

Tabela 1 – Apresenta-se a metologia utilizada na pesquisa para revisão bibliográfica.

Tabela 1, demonstra-se os resultados obtidos nas buscas realizadas nos repositórios de arquivos acadêmicos. As palavras-chaves utilizadas nas buscas foram: *Deep learning* (S1); Escores visuais (S2); Industria 4.0 (S3); Pecuária 4.0 (S4).

A primeira etapa se deu por refinar estes resultados, para realização disto os textos foram lidos as seções resumo e introdução, caso ao final da leitura o texto apresentasse cenário compatível com o do presente trabalho, este passava a compor os artigos selecionados, para que posteriormente fosse lido completamente e obter a conclusão de que está nos âmbitos da pesquisa, e passar a compor a lista de artigos usados, como é apresentado na Tabela 1.

Após o refinamento dos resultados e revisão destes, foram encontrados os seguintes trabalhos. O trabalho de (BRAGAMONTE, 2019), tem por objetivo realizar a avaliação corporal de gordura em bovinos, utilizando redes neurais convolucionais, foram obtidas 7.951 imagens, 6.758 foram destinadas ao treinamento da rede e 1.193 para testes. O resultado era de que, a correlação das classificações e das previsões deveria estar acima de 0.85. E os resultados obtidos foram de que a correlação da rede com os resultados reais foi de 0.97. Comprovando assim a eficácia do uso de Redes Neurais Convolucionais para automatizar o processo de classificação de gordura subcutânea em imagens ultrassonográficas de bovinos.

(OLIVEIRA et al., 2021) propuseram um estudo em pequenos ruminantes, para avaliação da carcaça destes, utilizando *deep learning*. Em geral a avaliação é realizada por

um especialista que utiliza de aspectos visuais para tomar uma decisão. Considerando isso, o estudo propôs que a arquitetura *PSPNetCNN*, alcançou resultados significativos com uma taxa de 0.89, o que permite o produtor determinar com maior precisão as medidas do animal e em menos tempo.

O trabalho proposto por (De La Iglesia et al., 2020), foi utilizado sensoração para classificação em tempo real dos bovinos. Durante a fase inicial, tratando-se do pré-processamento e seleção do contorno, cada imagem do banco foi adaptada para manter um padrão, assim foi possível selecionar os dados do contorno da carcaça e extrair os pixels, o próximo passo dividir a carcaça em diferentes regiões. Essa divisão é formada por linhas imaginárias, após a detecção completa de gordura presente na carcaça, as divisões são calculadas estimando o percentual de gordura existente no bovino. Os resultados apresentados foram que o sistema pode prever os resultados de classificação realizadas por um conjunto de 6 especialistas.

(BEDOYA; CASTRILLÓN, 2019) Realizaram uma pesquisa exploratória, devido ao aumento em roubo de bovinos na Colômbia, os quais geraram perdas milionárias no setor. O manejo de pastoreio é feito a campo aberto, o que dificulta o monitoramento de cada animal. O trabalho proposto construiu um sistema para realizar isto através de drones, foram capturadas 13.000 imagens para construção do modelo, de forma que este consiga detectar o gado utilizando redes neurais profundas, denominada yolo. Os resultados do treinamento do modelo foram uma precisão de 85% com níveis de sensibilidade de 75%, comprovando o potencial de utilização desta ferramenta.

O trabalho realizado por (KUMAR et al., 2017), apresenta um enfoque no reconhecimento biométrico dos bovinos. Para realização do mesmo foi utilizado aprendizado profundo para detecção, através de imagens do focinho primário, padrão de imagem possibilitando a abordagem do problema de animais perdidos ou por vezes trocado. O trabalho foi dividido em três etapas, primeira, consiste na preparação das imagens; segunda, extração das características do focinho, aplicando redes neurais convolucionais baseada na aprendizagem profunda. Nesta etapa da segunda fase foi aplicada juntamente a técnica de *denoising*, usada para codificar as características extraídas da imagens; e por fim a terceira, os resultados, os quais apresentaram a eficácia da utilização desta técnica, obtendo 98,99% de precisão da identificação.

3 Materiais e Métodos

A metodologia do presente trabalho, divide-se nas seguintes atividades, as quais estão apresentadas neste capítulo, no item 3.1:

3.1 Coleta de dados

A pesquisa foi conduzida na fazenda Sete Estrelas, localizada no município de Terenos - MS, que em parceria com a UFMS, Eng soluções tecnológicas e EMBRAPA, viabilizou os recursos como apoio técnico.

Foram inseridos 118 bovinos fêmeas em IATF (Inseminação Artificial por Tempo Fixo), com idade entre 15 e 18 meses, estes ficaram do mês de setembro de 2021 à fevereiro de 2022 no ciclo da IATF, ao final de fevereiro foi possível constatar os bovinos que obtiveram sucesso na prenhez.

Inicialmente, foram capturadas as imagens dos bovinos em formato de vídeo, utilizando uma câmera GoPro, e logo após foi realizada a segmentação das imagens *frame a frame*. Constatando-se a perda de algumas imagens, devido ao fato dos bovinos por vezes se assustarem com a câmera, e não se adequarem ao padrão de captura de imagem adotado. Logo, o número realmente utilizado do conjunto de bovinos foi de 91.

O padrão supracitado, dispõe-se da seguinte maneira: Capturar a lateral do animal. Como apresentado na Figura 11.



Figura 11 – Padrão de imagem da captura da lateral do animal. Fonte: (Elaborado pelo autor).

Após a prévia seleção das imagens, como anteriormente supracitado precisou-se aguardar o período de finalização da IATF para correlacionar o fenótipo com a precocidade e

fertilidade. Dos 91 bovinos identificados, 63 tornaram-se prenhas e 28 vazias (não atingiram prenhez), perfazendo uma taxa de prenhez de 69%.

Observa-se que para relacionar com absoluta certeza foram precisos 5 meses, durante este período foram capturadas imagens 3 vezes. Ou seja, 3 vezes este conjunto de bovinos transitou pelo mangueiro, afim de que, caso o bovino não tenha atingido prenhez na última tentativa de inseminação, este possa ter nova oportunidade.

Assim sendo, constata-se que manualmente todo o processo é lento e trabalhoso, por outro lado caso este fosse otimizado como proposto, o pecuarista apenas adicionaria os bovinos com que o algoritmo inferiu que fosse fértil.

Mais detalhes estão descritos sobre como esta proposta de algoritmo foi realizada no item 3.2.

3.2 Algoritmos

A primeira técnica utilizada foi o CNN em conjunto com a técnica U-Net. A rede U-Net foi usada para identificar o bovino na imagem, ou seja, executar a segmentação semântica. Esta técnica foi treinada e testada na plataforma online denominada *Colab*. A plataforma citada, é dividida em planos, sendo estes gratuitos e pagos, o plano utilizado foi o gratuito, pois apenas foi publicado um web-site simples para proposta do mesmo.

Todos os algoritmos estudados, foram desenvolvidos segundo um passo a passo. Apresentado pelo fluxograma na Figura 12.



Figura 12 – Fluxograma do desenvolvimento dos algoritmos. Fonte: Elaborado pelo autor.

O código identificado em Algoritmo 3.1, apresenta como é realizada a segmentação, assim sendo a identificação do bovino na imagem.

```
1  treino = [n_pre, pre]
2  lista_recortes, lista_ = [], []
3
4  for j in treino:
5      for i in j[2]:
6          image = np.array(cv2.imread(j[0] +str(i) + '.png'))
7          image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
8          results = modelo_segmentacao.detect([image], verbose=1)
9          r = results[0]
10
11  try:
12      local = np.where(r['class_ids'] == 20)[0][0]
13
14  if len(r['class_ids']) == 1:
15      cortar = r['rois'][local]
16      nova_mask = np.copy(r['masks'])
17      nova_mask[np.where(r['masks'] == False)] = 0
18      nova_mask[np.where(r['masks'] == True)] = 1
19  else:
20      cortar = r['rois'][local]
21      nova_mask = np.copy(r['masks'][:, :, local])
22      nova_mask[np.where(r['masks'][:, :, local] == False)] = 0
23      nova_mask[np.where(r['masks'][:, :, local] == True)] = 1
```

Algoritmo 3.1 – Algoritmo para identificação da mascara referente ao item 'bovino' - U-Net

Após a realização da segmentação, o banco de imagens foi plotado assim como na Figura 13.

O algoritmo CNN, é apresentado em Algoritmo 3.2, onde na linha de código 1, é segmentado o banco de imagens em 70% para treino e 30% para teste. E entre as linhas 3 e 18, é realizada a estrutura do modelo CNN. Por fim, na linha 20 apresenta o treinamento do modelo, em 20 passos por épocas.

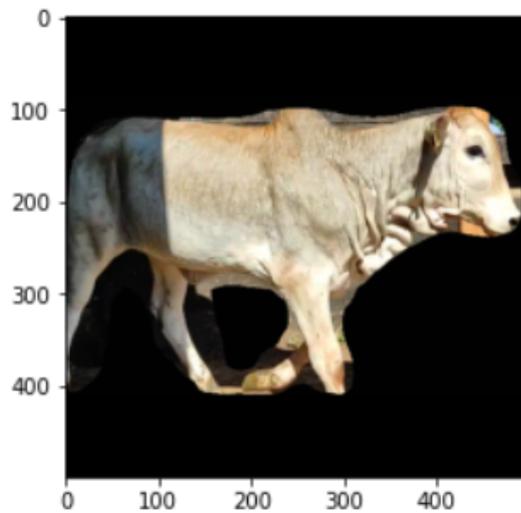


Figura 13 – Resultado da segmentação semântica, do banco de imagens. Fonte: Elaborado pelo autor.

```
1 x, x_t, y, y_t = train_test_split(lista_r, lista_, test_size =
    0.3, random_state=10)
2
3 model_classificacao = tf.keras.models.Sequential([
4 tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape
    =(500, 500, 3)),
5 tf.keras.layers.MaxPooling2D(2, 2),
6 tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
7 tf.keras.layers.MaxPooling2D(2,2),
8 tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
9 tf.keras.layers.MaxPooling2D(2,2),
10 tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
11 tf.keras.layers.MaxPooling2D(2,2),
12 tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
13 tf.keras.layers.MaxPooling2D(2,2),
14 tf.keras.layers.Flatten(),
15 tf.keras.layers.Dense(512, activation='relu'),
16 tf.keras.layers.Dense(1, activation='sigmoid')])
17
18 model_classificacao.compile(loss='binary_crossentropy', optimizer=
    'adam', metrics=['accuracy'])
```

```

19
20 history = model_classificacao.fit(x, y, steps_per_epoch=20,
    epochs=40, verbose=1, validation_data = (x_t, y_t),
    validation_steps=8)

```

Algoritmo 3.2 – Algoritmo CNN

Ao término dessa sequencia de passos, foi obtido o resultado da técnica em questão, que será apresentado no capítulo 4.

Para o treinamento das seguintes técnicas, Efficientnet, Resnet e Twins, foi utilizado o método *cross validation k fold* (REFAEILZADEH; TANG; LIU, 2009). Seu comportamento é da seguinte maneira, o conjunto de imagens foi segmentado assim como na Figura 13, logo após, as imagens foram separadas aleatoriamente em 5 pastas, das quais, uma sempre será a utilizada para teste e o restante para treino, de maneira que todas as imagens sejam treinadas e testadas. Para melhor compreensão a Figura 14, apresenta como este método funciona. É uma técnica, comumente utilizada em situações que o conjunto de dados tem um tamanho reduzido, evitando assim desperdício de informações.

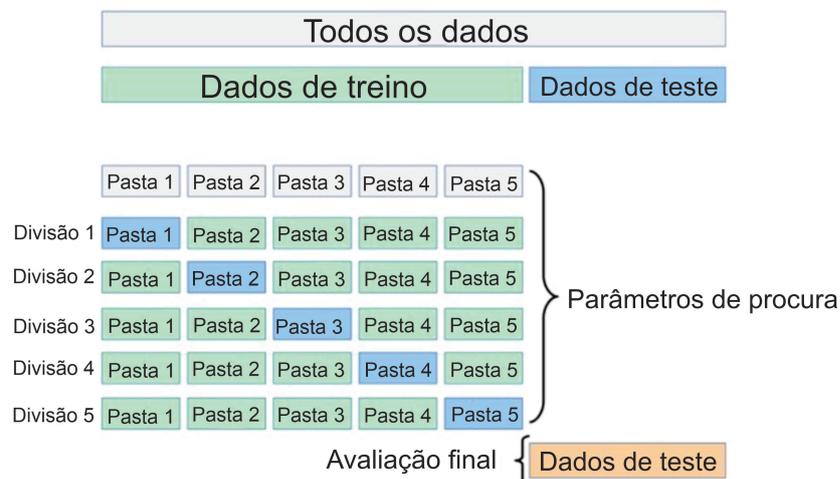


Figura 14 – Representação do comportamento da técnica Cross Validation K Fold. Fonte adaptado: (MAREDDIA, 2020).

Estas técnicas foram treinadas e testadas em computador de alta performance, que fica a disposição dos alunos da UFMS. Na Figura 15, é apresentada como as imagens foram particionadas em pastas, tanto de treino como para teste.

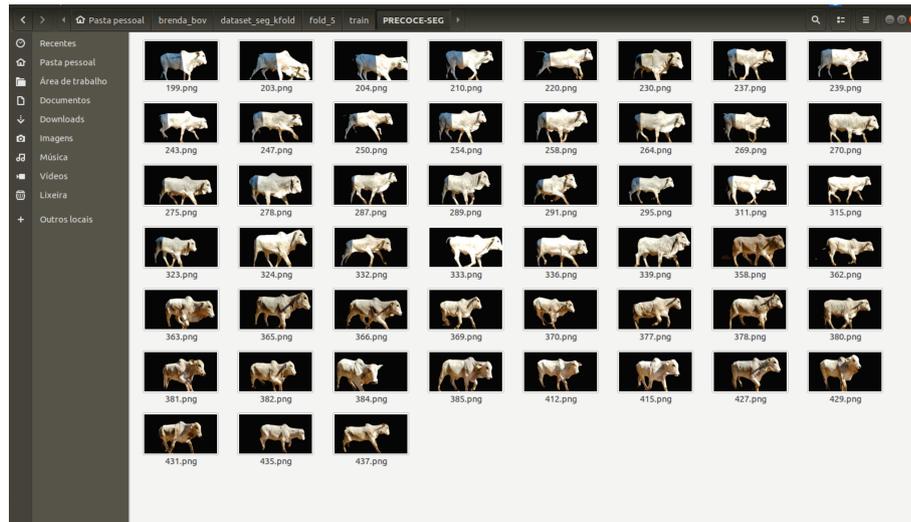


Figura 15 – Representação da divisão do banco de imagens, seguido o método de Cross Validation. Fonte: Elaborado pelo autor.

A configuração do *dataset*, é apresentada em Algoritmo 3.3, onde nas linhas iniciais do código, são configurados o conjunto de dados. Na linha 4, apreseta-se a configuração da normalização da imagem, afim de normalizar as imagens de entrada, também é configurado os valores médios utilizados para pré treinar o modelo. Na sequência, desde a linha 6 a 14, é configurado o *pipeline* de treino, e 16 a 22 *pipeline* de teste.

```

1  dataset_type = 'ImageNet'
2  classes = ('NAO-PRECOCE', 'PRECOCE')
3  img_size = 512
4  img_norm_cfg = dict(
5  mean=[123.675, 116.28, 103.53], std=[58.395, 57.12, 57.375],
6  to_rgb=True)
7  train_pipeline = [
8  dict(type='LoadImageFromFile'),
9  dict(type='Resize', size=img_size),
10 #dict(type='RandomCrop', size=img_size),
11 dict(type='RandomFlip', flip_prob=0.5, direction='horizontal'),
12 dict(type='Normalize', **img_norm_cfg),
13 dict(type='ImageToTensor', keys=['img']),
14 dict(type='ToTensor', keys=['gt_label']),
15 dict(type='Collect', keys=['img', 'gt_label'])
16 ]

```

```
16 test_pipeline = [  
17     dict(type='LoadImageFromFile'),  
18     dict(type='Resize', size=img_size),  
19     #dict(type='CenterCrop', crop_size=img_size),  
20     dict(type='Normalize', **img_norm_cfg),  
21     dict(type='ImageToTensor', keys=['img']),  
22     dict(type='Collect', keys=['img'])  
23 ]
```

Algoritmo 3.3 – Algoritmo de configuração do dataset

Finalizando a configuração do *dataset* geral, em Algoritmo 3.4, é configurado a métrica de avaliação do algoritmo, sendo esta métrica a acurácia.

```
1     evaluation = dict(  
2         interval=1,  
3         metric=['accuracy', 'precision', 'recall', 'f1_score'],  
4         metric_options={'topk':1},  
5         save_best='accuracy')
```

Algoritmo 3.4 – Algoritmo de configuração do dataset

Na sequência apresenta-se a configuração de cada um dos algoritmos usados no método *cross validation*, configurações estas que são o ponto principal, em que cada um se difere. O primeiro é o algoritmo Efficientnet, sua configuração é apresentada em Algoritmo 3.5

```
1     model = dict(  
2         type='ImageClassifier',  
3         backbone=dict(type='EfficientNet', arch='b7'),  
4         neck=dict(type='GlobalAveragePooling'),  
5         head=dict(  
6             type='LinearClsHead',  
7             num_classes=2,  
8             in_channels=2560,  
9             loss=dict(type='CrossEntropyLoss', loss_weight=1.0),  
10            topk=(1,),
```

```
11 ))
```

Algoritmo 3.5 – Configuração do algoritmo Efficientnet

```
1  model = dict(  
2  type='ImageClassifier',  
3  backbone=dict(  
4  type='ResNet',  
5  depth=50,  
6  num_stages=4,  
7  out_indices=(3, ),  
8  style='pytorch'),  
9  neck=dict(type='GlobalAveragePooling'),  
10 head=dict(  
11 type='LinearClsHead',  
12 num_classes=2,  
13 in_channels=2048,  
14 loss=dict(type='CrossEntropyLoss', loss_weight=1.0),  
15 topk=(1, ),  
16 ))
```

Algoritmo 3.6 – Configuração do algoritmo Resnet

```
1  model = dict(  
2  type='ImageClassifier',  
3  backbone=dict(  
4  type='SVT',  
5  arch='base',  
6  in_channels=3,  
7  out_indices=(3, ),  
8  qkv_bias=True,  
9  norm_cfg=dict(type='LN'),  
10 norm_after_stage=[False, False, False, True],  
11 drop_rate=0.0,  
12 attn_drop_rate=0.,  
13 drop_path_rate=0.3),  
14 neck=dict(type='GlobalAveragePooling'),
```

```

15  head=dict(
16  type='LinearClsHead',
17  num_classes=2,
18  in_channels=768,
19  loss=dict(
20  type='LabelSmoothLoss', label_smooth_val=0.1, mode='original'),
21  cal_acc=False),
22  init_cfg=[
23  dict(type='TruncNormal', layer='Linear', std=0.02, bias=0.),
24  dict(type='Constant', layer='LayerNorm', val=1., bias=0.)
25  ],
26  train_cfg=dict(augments=[
27  dict(type='BatchMixup', alpha=0.8, num_classes=2, prob=0.5),
28  dict(type='BatchCutMix', alpha=1.0, num_classes=2, prob=0.5)
29  ]))
30
31  data = dict(samples_per_gpu=128)
32
33  paramwise_cfg = dict(_delete=True, norm_decay_mult=0.0,
    bias_decay_mult=0.0)

```

Algoritmo 3.7 – Configuração do algoritmo Twins

A configuração de cada algoritmo, Efficientnet, Resnet e Twins, são descritas em Algoritmos 3.5, 3.6 e 3.7, respectivamente. Como são algoritmos diferentes, as configurações são específicas para cada um, seja com relação a normalização das imagens, em quantas classes possuem e em como será realizado a profundidade de cada de cada janela para treino. Ou seja, o comportamento dos mesmos são peculiares a cada um.

Com relação aos resultados, são gerados arquivos em *Json*, contendo a acurácia obtida. Portanto, sempre que o algoritmo Resnet por exemplo, é treinado e testado em determinada pasta do conjunto de imagens do banco, um arquivo é gerado contendo o resultado da porcentagem da acurácia obtida, para aquela pasta. E assim, sucessivamente para todos os algoritmos usados no método do *cross validation*.

Os algoritmos apresentados, são o *back-end* de uma possível ferramenta que, integrará com o *front-end*, afim de que seja acessível para os usuários que pretendam fazer uso. Na seção 3.3, é descrito o comportamento deste *front-end*

3.3 Proposta do Front-end

Neste trabalho foi desenvolvido um web-site, afim de proporcionar um ambiente de interface gráfica amigável em que seja possível, inserir imagens dos bovinos e obter o resultado desejado, seja ele positivo para precocidade a fertilidade ou não. Contudo, houveram contratempos com integração do melhor algoritmo de detecção destas características, e o web-site, resultando apenas em uma proposta de interface.

As linguagens utilizadas no desenvolvimento foram *HTML*, *CSS*, *JavaScript* e *PHP*, a integração destas linguagens resulta em um web-site responsivo, ou seja, ele se adapta a determinado dispositivo que o usuário esteja utilizando. Para que o usuário tenha acesso deste web-site, este foi armazenado em uma plataforma que permite acessar as páginas gratuitamente denominado *GitHub*, seguindo o cenário apresentado. O link para o web-site: <https://brendammedina.github.io/SystemPrecFer/>. O nome proposto foi, SystemPrecFer, sigla advinda de sistema de precocidade e fertilidade. Em conjunto, foi desenvolvida uma logo, apresentada na Figura 16.



Figura 16 – Logo SystemPrecFer. Fonte: Elaborado pelo autor.

A seguir será descrito como é o comportamento do web-site, onde o pecuarista deverá inserir as imagens do bovino, e após seu processamento será apresentado para ele a resposta com relação se o bovino está precoce a ser fértil ou não.



Figura 17 – Apresentação da tela inicial do web-site. Fonte: Elaborado pelo autor.

A Figura 17, demonstra a tela inicial do web-site, ao lado esquerdo dispõe-se do menu onde cada um tem uma função específica. Na tela de instruções apresenta o posicionamento do bovino na imagem, na tela de processamento apresenta como será inserida a imagem para o processamento e contato, onde pode-se enviar alguma sugestão, juntamente com e-mail do autor para realização disto.

A Figura 18, apresenta a tela de instruções, onde o pecuarista seguindo os passos indicados através de seta, poderá realizar o processamento da imagem que deseja.

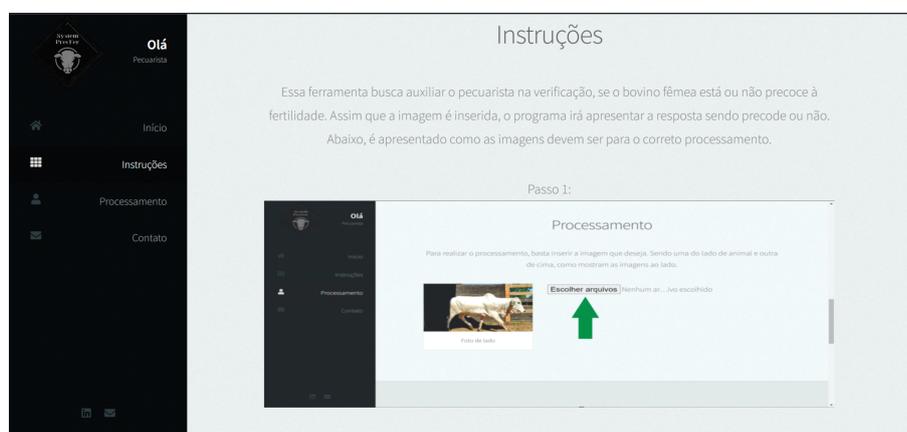


Figura 18 – Apresentação da tela de instruções, sobre o funcionamento do web-site. Fonte: Elaborado pelo autor.

Portanto, resulta em uma proposta para possível integração futura, do web-site e o algoritmo de detecção e classificação, das características buscadas.

4 Resultados

Nesta seção, são demonstrados os resultados obtidos dos algoritmos.

Após o treinamento e teste, o conjunto de técnicas CNN e a U-Net, apresentaram na Tabela 2, o seguinte resultado:

Algoritmo	Acurácia	Erro
CNN	71.42%	36.31%

Tabela 2 – Resultado do algoritmo CNN.

Na Figura 19, apresenta-se como este resultado é exposto na plataforma colab, foi plotado a matriz de confusão, onde: 0 são os animais não precoces a fertilidade e 1 os precoces a tal.

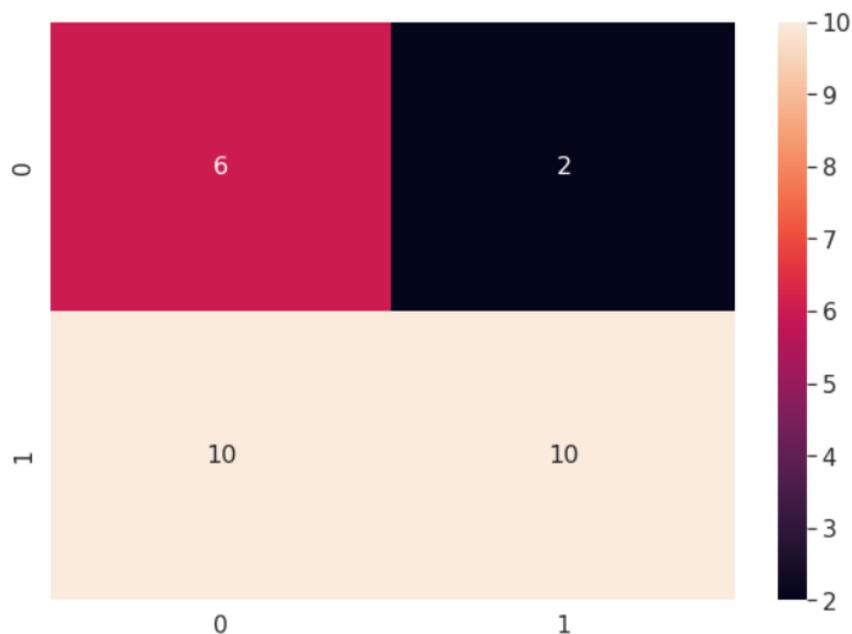


Figura 19 – Resultado do algoritmo treinado e testado na plataforma colab. Fonte: Elaborado pelo autor.

Logo abaixo, apresenta-se os resultados de cada pasta das técnicas restantes, Efficientnet, Resnet e Twins. Estes resultados, após treinamento e teste, são gerados arquivos *Json*, que contém a porcentagem de acurácia de cada técnica, o resultado das porcentagens de acurácia e erro são apresentados na Tabela 3.

Na Figura 20, apresenta-se como estes resultados são gerados. A seta azul, mostra a seguinte informação: Best accuracy is 82.3529 at 15 epoch. Resultado o qual foi apresentado, da pasta 5 do algoritmo resnet, na Tabela 3.

	Efficientnet	Resnet	Twins
Acurácia Pasta 1	73.68%	68.42%	73.68%
Erro Pasta 1	26.32%	31.58%	26.32%
Acurácia Pasta 2	68.42%	78.95%	68.42%
Erro Pasta 2	31.58%	21.05%	31.58%
Acurácia Pasta 3	78.95%	68.42%	68.42%
Erro Pasta 3	21.05%	31.58%	31.58%
Acurácia Pasta 4	76.47%	76.47%	70.58%
Erro Pasta 4	23.53%	23.53%	29.42%
Acurácia Pasta 5	70.59%	82.35%	70.59%
Erro Pasta 5	29.41%	17.65%	29.41%
Média Acurácia	73.62%	74.92%	70.46%

Tabela 3 – Resultados dos algoritmos, Efficientnet, Resnet e Twins, e as médias e erros.

```

2022-07-15 11:14:26,209 - mmcls - INFO - Now best checkpoint is saved as best_accuracy_epoch_2.pth.
2022-07-15 11:14:28,269 - mmcls - INFO - Best accuracy is 70.5882 at 2 epoch.
2022-07-15 11:14:28,269 - mmcls - INFO - Epoch(val) [2][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:14:32,673 - mmcls - INFO - Epoch(val) [3][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:14:37,040 - mmcls - INFO - Epoch(val) [4][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:14:41,478 - mmcls - INFO - The previous best checkpoint /home/wesley/colabo/mclassification/work_dirs/resnet101_2xb32_brenda/best_accuracy_epoch_2.pth was removed
2022-07-15 11:14:42,082 - mmcls - INFO - Now best checkpoint is saved as best_accuracy_epoch_5.pth.
2022-07-15 11:14:42,082 - mmcls - INFO - Best accuracy is 76.4706 at 5 epoch.
2022-07-15 11:14:42,082 - mmcls - INFO - Epoch(val) [5][5] accuracy: 76.4706, precision: 87.5000, recall: 60.0000, f1_score: 59.5238
2022-07-15 11:14:46,470 - mmcls - INFO - Epoch(val) [6][5] accuracy: 64.7059, precision: 57.5000, recall: 57.5000, f1_score: 57.5000
2022-07-15 11:14:50,866 - mmcls - INFO - Epoch(val) [7][5] accuracy: 29.4118, precision: 14.7059, recall: 50.0000, f1_score: 22.7273
2022-07-15 11:14:55,242 - mmcls - INFO - Epoch(val) [8][5] accuracy: 58.8235, precision: 53.0303, recall: 53.3333, f1_score: 52.9644
2022-07-15 11:14:59,586 - mmcls - INFO - Epoch(val) [9][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:15:03,988 - mmcls - INFO - Epoch(val) [10][5] accuracy: 41.1765, precision: 66.6667, recall: 58.3333, f1_score: 39.2857
2022-07-15 11:15:08,417 - mmcls - INFO - Epoch(val) [11][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:15:12,757 - mmcls - INFO - Epoch(val) [12][5] accuracy: 70.5882, precision: 61.6667, recall: 55.8333, f1_score: 55.0265
2022-07-15 11:15:17,175 - mmcls - INFO - Epoch(val) [13][5] accuracy: 29.4118, precision: 38.3333, recall: 44.1667, f1_score: 27.1429
2022-07-15 11:15:21,540 - mmcls - INFO - Epoch(val) [14][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:15:25,998 - mmcls - INFO - The previous best checkpoint /home/wesley/colabo/mclassification/work_dirs/resnet101_2xb32_brenda/best_accuracy_epoch_5.pth was removed
2022-07-15 11:15:26,625 - mmcls - INFO - Now best checkpoint is saved as best_accuracy_epoch_15.pth.
2022-07-15 11:15:26,625 - mmcls - INFO - Best accuracy is 82.3529 at 15 epoch.
2022-07-15 11:15:26,625 - mmcls - INFO - Epoch(val) [15][5] accuracy: 82.3529, precision: 70.0000, recall: 70.0000, f1_score: 73.0159
2022-07-15 11:15:30,998 - mmcls - INFO - Epoch(val) [16][5] accuracy: 76.4706, precision: 71.6667, recall: 71.6667, f1_score: 71.6667
2022-07-15 11:15:35,386 - mmcls - INFO - Epoch(val) [17][5] accuracy: 58.8235, precision: 62.8571, recall: 65.0000, f1_score: 58.2456
2022-07-15 11:15:39,769 - mmcls - INFO - Epoch(val) [18][5] accuracy: 58.8235, precision: 57.6389, recall: 59.1667, f1_score: 56.4103
2022-07-15 11:15:44,140 - mmcls - INFO - Epoch(val) [19][5] accuracy: 70.5882, precision: 65.9091, recall: 67.5000, f1_score: 66.4032
2022-07-15 11:15:48,144 - mmcls - INFO - Saving checkpoint at 20 epochs
2022-07-15 11:15:49,142 - mmcls - INFO - Epoch(val) [20][5] accuracy: 76.4706, precision: 71.6667, recall: 71.6667, f1_score: 71.6667
2022-07-15 11:15:53,511 - mmcls - INFO - Epoch(val) [21][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:15:57,859 - mmcls - INFO - Epoch(val) [22][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:16:02,269 - mmcls - INFO - Epoch(val) [23][5] accuracy: 70.5882, precision: 35.2941, recall: 50.0000, f1_score: 41.3793
2022-07-15 11:16:06,693 - mmcls - INFO - Epoch(val) [24][5] accuracy: 76.4706, precision: 72.6190, recall: 65.8333, f1_score: 67.3077
2022-07-15 11:16:11,099 - mmcls - INFO - Epoch(val) [25][5] accuracy: 76.4706, precision: 87.5000, recall: 60.0000, f1_score: 59.5238
2022-07-15 11:16:15,491 - mmcls - INFO - Epoch(val) [26][5] accuracy: 58.8235, precision: 62.8571, recall: 65.0000, f1_score: 58.2456
2022-07-15 11:16:19,847 - mmcls - INFO - Epoch(val) [27][5] accuracy: 41.1765, precision: 52.8846, recall: 52.5000, f1_score: 40.9722
2022-07-15 11:16:24,229 - mmcls - INFO - Epoch(val) [28][5] accuracy: 82.3529, precision: 90.0000, recall: 70.0000, f1_score: 73.0159
2022-07-15 11:16:28,638 - mmcls - INFO - Epoch(val) [29][5] accuracy: 82.3529, precision: 79.8077, recall: 75.8333, f1_score: 77.3333
2022-07-15 11:16:32,644 - mmcls - INFO - Saving checkpoint at 30 epochs
2022-07-15 11:16:33,648 - mmcls - INFO - Epoch(val) [30][5] accuracy: 47.0588, precision: 67.8571, recall: 62.5000, f1_score: 46.3158

```

Figura 20 – Resultado do algoritmo treinado e testado na plataforma colab. Fonte: Elaborado pelo autor.

Os resultados de todos os algoritmos aqui estudados, buscam inferir qual foi o melhor algoritmo. Mesmo que estes tenham comportamentos diferentes e maneiras diferentes de configurações, são utilizados para o mesmo objetivo. No capítulo 5, será apresentada uma discussão sobre os resultados apresentados.

5 Discussão

Neste capítulo, apresenta-se a discussão e o porque determinado algoritmo, se sobressaiu sobre os demais.

Para identificar o algoritmo com maior porcentagem de acurácia obtida, foi realizada média aritmética simples, dos resultados de cada treinamento e teste das pastas do *cross validation*, sendo assim os algoritmos que utilizaram desta técnica, seus resultados foram somados e divididos pelo número total de pastas, o qual é cinco.

Tendo em vista que o algoritmo CNN e U-net não foram utilizados no *cross validation*, estes apresentam apenas o resultado da acurácia obtida, sem a necessidade de realizar algum calculo matemático.

$$\bar{X} = \frac{X_1 + X_2 + X_3 + \dots + X_n}{n} \quad (2)$$

A equação de número de 2, demonstra como é calculado esta fórmula, sendo que o \bar{X} é o número final ao qual buscamos encontrar; os X são os conjunto de números que somados dividimos por n , que é o valor total de conjunto de números somados.

Algoritmo	Acurácia	Erro
CNN	71.42%	28.58%
Efficientnet	73.62%	26.38%
Resnet	74.92%	25.08%
Twins	70.46%	29.54%

Tabela 4 – Resultado de todos os algoritmos estudados, com relação a acurácia.

Na Tabela 4, expõe os resultados de todos os algoritmos estudados, estes mesmo que distintos, atingiram acurácias bastante similares. Foi calculado o desvio padrão, para indicar o dispersão do conjunto de dados, o qual quanto mais próximo de 0, mais homogêneo são os dados. A equação 3, descreve a fórmula do desvio padrão.

$$S : \sqrt{\frac{\sum(x_i - M_A)^2}{n - 1}} \quad (3)$$

A equação de 3, apresenta como é calculado o desvio padrão, onde:

- \sum somatória de todos os termos da primeira posição ($i = 1$) até a última posição (n).
- x_i valor do dado obtido na primeira posição, caso i seja igual a 1.
- M_A : média aritmética dos dados obtidos.

- n quantidade de dados obtidos.

O resultado obtido do desvio padrão foi de 1.83, o que mostra a uniformidade da acurácia atingida pelos algoritmos.

Como é possível observar o algoritmo Resnet obteve melhor resultado com relação aos demais, a Figura 21 apresenta o melhor resultado classificado por este, com 82.35% de acurácia.



Figura 21 – Melhor resultado obtido pelo algoritmo Resnet. Fonte: Elaborado pelo autor.

Realizando um comparativo com o algoritmo que obteve resultado inferior, Twins, a Figura 22 apresenta o pior resultado classificado por este, com 68.42% de acurácia .



Figura 22 – Melhor resultado obtido pelo algoritmo Twins. Fonte: Elaborado pelo autor.

Observando as imagens supracitadas, é possível perceber que o visualmente o bovino da Figura 21, apresenta musculatura melhor com relação ao bovino da Figura 22. Vale ressaltar o posicionamento do bovino na imagem, sendo que o da Figura 22, aparentemente

assustou-se com algo no transito do mangueiro, o que pode influenciar no treinamento e teste do algoritmo, afim de classificar se este é precoce ou não a fertilidade.

No capítulo 6, será apresentada considerações sobre o funcionamento de todo o trabalho, incluindo possíveis propostas futuras.

6 Conclusões e Considerações Finais

Neste trabalho foi abordado o desenvolvimento de um sistema para identificação de fertilidade bovina, relacionado com a precocidade dos mesmos, no qual analisou-se o desempenho de diferentes algoritmos de Inteligência Artificial. Apesar de não possuir a integração do *back-end* com o *front-end*, os resultados obtidos através dos algoritmos de IA, são atrativos. Resultados que causam expectativas para finalização da integração entre *back e front-end*.

Conclui-se que o melhor algoritmo aqui estudado, foi o Resnet, demonstrando acurácia de 74.92%. Esta porcentagem é promissora, porém é possível elevar esta porcentagem realizando alguns ajustes, sendo estes, melhorar a captação de imagens, ou seja, padronizar o dispositivo utilizado para captura e aumentar o número de imagens no banco.

Alguns fatores influenciaram diretamente no desenvolvido de todo o trabalho, causando obstáculos no decorrer do mesmo. Estes são descritos na seção 6.1.

6.1 Limitações

As limitações encontradas no decorrer da pesquisa, estão relacionados principalmente à fatores decorridos da pandemia do covid-19. Como apresenta (CIOTTI et al., 2020), a OMS(Organização mundial da saúde), declarou pandemia global, e protocolos de biossegurança foram adotados por todo o mundo. Devido a este cenário, não foi possível coletar as imagens dos bovinos no primeiro ano da pesquisa.

Com este atraso na captura das imagens, o desenvolvimento do restante do trabalho, não foi realizado de maneira linear, pois mesmo depois de capturada as imagens, foi preciso aguardar o final da IATF, para poder correlacionar as imagens com sucesso na fertilidade.

Todo o processo do trabalho, foi lento. Devido aos fatores supracitados, a integração do *back e front-end* não foi realizada, por não compor o escopo do trabalho, no presente momento. Na seção 6.2, é apresentado trabalhos futuros, com relação a continuação do presente trabalho.

6.2 *Trabalhos Futuros*

O *front-end*, apenas foi apresentado como uma proposta. E para trabalhos futuros, estima-se concluir esta etapa, para que realmente o usuário seja possibilitado que fazer o uso desta ferramenta, online.

Os algoritmos apresentam resultados promissores em poupar tempo e investimentos dos pecuaristas, principalmente com relação a identificação de animais mais precoces e férteis dentro de um rebanho. Gerando ainda a possibilidade de desenvolver uma API, buscando integração do algoritmo estudado em uma ferramenta acoplada no mangueiro, a qual realizaria a captura da imagem em tempo contínuo de forma a entregar o resultado de precocidade, sempre que o animal transitar pelo mangueiro.

Referências

- ARAÚJO, F. H. D. d. Convnets na caracterização, recuperação e ranqueamento de células. 2018. Citado na página 23.
- BEDOYA, D. I. G.; CASTRILLÓN, R. Reconocimiento automático de ganado bovino a partir de imágenes aéreas tomadas con drones: Un enfoque exploratorio. In: *Memorias de Congresos UTP*. [S.l.: s.n.], 2019. p. 32–39. Citado na página 29.
- BeefPoint. *Escores Visuais de CPM: avaliação da qualidade de carcaça*. 2004. Disponível em: <https://www.beefpoint.com.br/escores-visuais-de-cpm-avaliacao-da-qualidade-de-carcaca-21690/>. Citado 2 vezes nas páginas 7 e 16.
- BRAGAMONTE, J. S. *Mensuração automática de espessura de gordura subcutânea a partir de imagens ultrassonográficas de bovinos utilizando deep learning*. 2019. Monografia (Bacharel em Engenharia da Computação), URCAMP (Universidade Federal do Pampa), Bagé, Brazil. Citado na página 28.
- BRUMATTI, R. et al. Desenvolvimento de índice de seleção em gado corte sob o enfoque de um modelo bioeconômico. *Archivos de Zootecnia*, v. 60, p. 205–213, 06 2011. Citado na página 19.
- CHEN, L.-C. et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 40, n. 4, p. 834–848, 2017. Citado na página 27.
- CHIARI LUCIMARA. *Embrapa Gado de Corte: pesquisas e parcerias com a UFMS*. 2020. Disponível em: <https://www.ufms.br/embrapa-gado-de-corte-pesquisas-e-parcerias-com-a-ufms/>. Entrevista concedida PIMENTA Paula Acesso em: 03 de dezembro 2020. Citado na página 13.
- CIOTTI, M. et al. The covid-19 pandemic. *Critical reviews in clinical laboratory sciences*, Taylor & Francis, v. 57, n. 6, p. 365–388, 2020. Citado na página 46.
- COSTA, R. D.; C.H.C., M. Correlações fenotípicas entre as características morfológicas e zootécnicas de bovinos da raça nelore participantes de provas em ganho de peso a pasto. 2013. Citado na página 18.
- De La Iglesia, D. H. et al. Non-invasive automatic beef carcass classification based on sensor network and image analysis. *Future Generation Computer Systems*, v. 113, p. 318–328, 2020. ISSN 0167-739X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X19317492>. Citado na página 29.
- DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. Citado 2 vezes nas páginas 7 e 27.
- DRASS, M. et al. Semantic segmentation with deep learning: detection of cracks at the cut edge of glass. *Glass Structures & Engineering*, Springer, v. 6, n. 1, p. 21–37, 2021. Citado 2 vezes nas páginas 7 e 21.

- DRATH, R.; HORCH, A. Industrie 4.0 – hit or hype? *IEEE Industrial Electronics Magazine*, p. 56–58, 2014. Citado na página 13.
- ERICKSON, B. J. et al. Machine learning for medical imaging. *Radiographics*, Radiological Society of North America, v. 37, n. 2, p. 505, 2017. Citado na página 22.
- FAO. How to feed the world in 2050. high level expert forum. *Convened at FAO Headquarters in Rome*, 2009. Citado na página 13.
- FARIA, C. U. de et al. *Utilização de Escores Visuais de Características Morfológicas de Bovinos Nelore como Ferramenta para o Melhoramento Genético Animal*. [S.l.: s.n.], 2007. Citado na página 16.
- FILHO, W. K. et al. Avaliação visual - epmuras descritivo. *Caderno de Ciências Agrárias*, v. 7, p. 12–21, 2015. Citado 2 vezes nas páginas 7 e 17.
- FREITAS, V. J. de F. Eficiência reprodutiva em bovinos. *UNIVERSIDADE ESTADUAL DO CEARÁ*, 2010. Citado na página 13.
- GEORGIEV, N.; ASENOV, A. Automatic segmentation of lumbar spine mri using ensemble of 2d algorithms. In: SPRINGER. *International Workshop and Challenge on Computational Methods and Clinical Applications for Spine Imaging*. [S.l.], 2018. p. 154–162. Citado 2 vezes nas páginas 7 e 24.
- GOMES, W. B. S. *Detecção de landmarks e classificação de subespécies de abelhas através de asas com deep learning*. Tese (Doutorado) — Instituto Politecnico de Braganca (Portugal), 2018. Nenhuma citação no texto.
- HAO, S.; YUHANG, X.; LIAN, C. Facial expression recognition based on multi-scale feature fusion and attention mechanism. v. 39, n. 3, p. 34–40, 2022. Citado na página 24.
- HE, K.; SUN, J. Convolutional neural networks at constrained time cost. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 5353–5360. Citado na página 26.
- HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 1026–1034. Citado na página 26.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado 2 vezes nas páginas 24 e 26.
- HE, K. et al. Identity mappings in deep residual networks. In: SPRINGER. *European conference on computer vision*. [S.l.], 2016. p. 630–645. Citado 2 vezes nas páginas 7 e 26.
- HEROKU. *Heroku*. 2007. Disponível em: <https://heroku.com/>. Acesso em: 15 de maio 2021. Nenhuma citação no texto.
- HIJAZI, S. et al. Using convolutional neural networks for image recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*, v. 9, 2015. Citado na página 23.
- HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 7132–7141. Citado na página 24.

- JOSAHKIAN, L. A.; LUCAS, C. H.; MACHADO, C. H. C. Manual do serviço de registro genealógico das raças zebuínas e pmgz. p. 190, 2009. Citado na página 14.
- KUMAR, S. et al. Deep learning framework for recognition of cattle using muzzle point image pattern. *Measurement*, v. 116, p. 1–17, 10 2017. Citado na página 29.
- LASI, H. et al. Industry 4.0. v. 6, n. 4, p. 239–242, 2014. Citado na página 13.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, 2015. Citado na página 22.
- LI, F.-F.; JOHNSON, J.; YEUNG, S. *Lecture 11: Detection and segmentation*. [S.l.]: Stanford university, 2017. Citado 2 vezes nas páginas 7 e 24.
- LIU, Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2021. p. 10012–10022. Citado na página 27.
- MAREDA, R. Analysis of google play store data set and predict the popularity of an app on google play store. 2020. Citado 2 vezes nas páginas 8 e 34.
- MASSRUHÁ, S. M. F. S.; LEITE, M. d. A. Agro 4.0-rumo à agricultura digital. In: IN: MAGNONI JÚNIOR, L.; STEVENS, D.; SILVA, WTL DA; VALE, JMF DO; PURINI, SR *Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE)*. [S.l.], 2017. Citado na página 13.
- MATTAR, M. et al. Fatores genéticos e ambientais sobre a probabilidade de prenhez precoce em bovinos caracu. *Ciência Rural*, 2007. Citado na página 19.
- NEIVA, F.; SILVA, R. *Revisão Sistemática da Literatura em Ciência da Computação - Um Guia Prático*. 2016. Citado na página 28.
- NIELSEN, B. et al. Entropy-based adaptive nuclear texture features are independent prognostic markers in a total population of uterine sarcomas. *Cytometry Part A*, Wiley Online Library, v. 87, n. 4, p. 315–325, 2015. Citado na página 22.
- OLIVEIRA, T. et al. Segmentação de imagens de ultrassonografia da carcaça em pequenos ruminantes utilizando deep learning. In: *Anais do XIV Encontro Unificado de Computação do Piauí e XI Simpósio de Sistemas de Informação*. Porto Alegre, RS, Brasil: SBC, 2021. p. 144–151. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/enucompi/article/view/17765>>. Citado na página 28.
- ONU. United nations, department of economic and social affairs. *The United Nations, Population Division, Population Estimates and Projections Section*, 2012. Citado na página 13.
- PACHECO, C.; PEREIRA, N. Deep learning conceitos e utilização nas diversas Áreas do conhecimento. v. 2, p. 34–49, 12 2018. Citado na página 21.
- PENHA, D. d. P. et al. Rede neural convolucional aplicada à identificação de equipamentos residenciais para sistemas de monitoramento não-intrusivo de carga. Universidade Federal do Pará, 2018. Citado na página 22.

- PIRES, A. V. et al. Precocidade reprodutiva em bovinos de corte. *Caderno de Ciências Agrárias*, 2015. Citado na página 19.
- PONTI, M. A.; COSTA, G. B. P. da. Como funciona o deep learning. *arXiv preprint arXiv:1806.07908*, 2018. Citado na página 20.
- REFAEILZADEH, P.; TANG, L.; LIU, H. Cross-validation. *Encyclopedia of database systems*, Springer, v. 5, p. 532–538, 2009. Citado na página 34.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241. Citado na página 23.
- ROSA, A. d. N.; ABREU, U. de. Características que devem ser avaliadas para seleção ou compra de touros. Corumbá: Embrapa Pantanal, 2007., 2007. Citado na página 18.
- SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, v. 61, p. 85–117, 2015. Published online 2014; based on TR arXiv:1404.7828 [cs.NE]. Citado na página 20.
- SHAHEEN, F.; VERMA, B.; ASAFUDDOULA, M. Impact of automatic feature extraction in deep learning architecture. In: *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. [S.l.: s.n.], 2016. p. 1–8. Citado na página 21.
- SILVA, A. B. *Métodos Computacionais para Análise e Classificação de Displasias em Imagens da Cavidade Bucal*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, 2019. Citado 2 vezes nas páginas 7 e 22.
- TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2019. p. 6105–6114. Citado 3 vezes nas páginas 7, 24 e 25.
- TEIXEIRA, J. *O que é inteligência artificial*. [S.l.]: E-Galáxia, 2019. Citado na página 20.
- TEIXEIRA, R. A.; ALBUQUERQUE, L. G.; FRIES, L. A. Comparações bioeconômicas entre três idades à primeira cobertura em novilhas nelore. *Ars Veterinaria*, 2002. Citado na página 19.
- VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: SN. *Proceedings of the xxix conference on graphics, patterns and images*. [S.l.], 2016. v. 1, n. 4. Citado 2 vezes nas páginas 7 e 23.
- WANG, W. et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2021. p. 568–578. Citado na página 27.
- ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 818–833. Citado na página 25.
- ZHANG, X. *Machine learning. A Matrix Algebra Approach to Artificial Intelligence*, 2020. Citado na página 20.