



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

HERCULES RODRIGUES ANSELMO

**APRENDIZAGEM DE MÁQUINA APLICADA AO CONTROLE
DE REBANHO BOVINOS**

CAMPINA GRANDE - PB

2022

HERCULES RODRIGUES ANSELMO

**APRENDIZAGEM DE MÁQUINA APLICADA AO CONTROLE DE
REBANHO BOVINOS**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

Orientadora : Joseana Macêdo Fachine Régis de Araújo

CAMPINA GRANDE - PB

2022

HERCULES RODRIGUES ANSELMO

**APRENDIZAGEM DE MÁQUINA APLICADA AO CONTROLE DE
REBANHO BOVINOS**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

BANCA EXAMINADORA:

**Joseana Macêdo Fechine Régis de Araújo
Orientadora – UASC/CEEI/UFCG**

**Leandro Balby Marinho
Examinador – UASC/CEEI/UFCG**

**Francisco Vilar Brasileiro
Professor da Disciplina TCC – UASC/CEEI/UFCG**

Trabalho aprovado em: 02 de Setembro de 2022.

CAMPINA GRANDE - PB

RESUMO

A pecuária de corte de bovinos é uma das atividades mais importantes para o setor pecuário no Brasil. Neste contexto, proprietários de grandes confinamentos possuem dificuldade em manter o controle sobre a quantidade de animais que possuem e sobre sua segurança antifurto. A proposta deste trabalho consiste em apresentar uma técnica capaz de identificar “faces” de animais da espécie bovina (“cabeças de gado”), em especial, para as raças Nelores, a partir de imagens desses animais, seguida da contagem dessas faces, informando ao proprietário a quantidade de animais do seu rebanho presente na imagem. A técnica foi desenvolvida utilizando conceitos de aprendizagem de máquina, com aprendizagem supervisionada, baseada em classificadores de cascatas de Haar, contando com datasets públicos disponibilizados e obtenção de novos dados amostrais. O modelo possui precisão total acima de 95% para detecção de faces em uma imagem.

Palavras-chave: Aprendizado de máquina, rebanhos bovinos, dataset, detecção e contagem de faces.

APRENDIZAGEM DE MÁQUINA APLICADA AO CONTROLE DE REBANHO BOVINOS

Trabalho de Conclusão de Curso

Hércules Rodrigues Anselmo
Universidade Federal de Campina Grande
Campina Grande, Paraíba
hercules.anselmo@ccc.ufcg.edu.br

Joseana Macêdo Fachine Régis de Araújo
Universidade Federal de Campina Grande
Campina Grande, Paraíba
joseana@computacao.ufcg.edu.br

RESUMO

A pecuária de corte de bovinos é uma das atividades mais importantes para o setor pecuário no Brasil. Neste contexto, proprietários de grandes confinamentos possuem dificuldade em manter o controle sobre a quantidade de animais que possuem e sobre sua segurança antifurto. A proposta deste trabalho consiste em apresentar uma técnica capaz de identificar “faces” de animais da espécie bovina (“cabeças de gado”), em especial, para as raças Nelores¹, a partir de imagens desses animais, seguida da contagem dessas faces, informando ao proprietário a quantidade de animais do seu rebanho presente na imagem. A técnica foi desenvolvida utilizando conceitos de aprendizagem de máquina, com aprendizagem supervisionada, baseada em classificadores de cascatas de Haar, contando com *datasets* públicos disponibilizados e obtenção de novos dados amostrais. O modelo possui precisão total acima de 95% para detecção de faces em uma imagem.

Palavras-chave

Aprendizado de máquina, rebanhos bovinos, *dataset*, detecção e contagem de faces.

Repositório

<https://github.com/herculesra/tcc-detect-face-cow.git>

1. INTRODUÇÃO

A população de gado no Brasil é maior do que a sua própria população humana, com mais de 215 milhões de cabeças de gado registradas no ano de 2020 [1]. Sete países concentram 90% do rebanho mundial: Índia, Brasil, Estados Unidos, China, União Europeia,

¹ **Nelore** é uma raça de gado bovino originária da Índia. Possui a pelagem branca e a pele preta. Fonte: <<https://pt.wikipedia.org/wiki/Nelore>>. Acesso em 10 de ago 2022.

Argentina e Austrália. Índia e Brasil representam 55%, respectivamente, 30,6% e 24,7% [2]. Possuir uma fazenda para criação desses animais demanda tempo, gestão e espaço. Dependendo da quantidade de animais da espécie, a criação desses pode trazer grande dificuldade ao criador. Contabilizar a quantidade de animais em segundos e obter uma resposta rápida pode ser uma tarefa árdua, se for feita de forma manual. Porém, tal tarefa pode ser otimizada a partir do uso de um sistema automático e inteligente para colaborar com essa contagem.

Soluções para aumentar a segurança e rastreabilidade do rebanho bovino já foram e estão sendo desenvolvidas. A mais comum dentre essas é a rastreabilidade via chip, responsável por armazenar informações relevantes sobre o animal. Esta técnica é implementada a partir da incisão de um chip no rúmen do animal ou um brinco, que emite um sinal de RFID (*Radio Frequency Identification*), com bits transmitidos por um transponder e interpretados por um transceptor [3], que captura a informação e a disponibiliza no Sistema Brasileiro de Identificação e Certificação de Bovinos e Bubalinos (SISBOV).

Pensando no bem estar dos animais e no quanto a técnica de identificação RFID pode ser agressiva, estressando o animal, estudos foram realizados e técnicas foram desenvolvidas, algumas bem conhecidas como a técnica para detecção bovina, que leva em consideração o focinho do animal [4] e também, detecção por pelagem [5]. Ambas são técnicas relevantes e que possuem bons resultados. No entanto, a técnica de identificação pelo focinho do animal requer uma imagem de alta resolução, além do que, para realizar a verificação, esta parte do animal deve estar limpa, o que não é trivial manter em um ambiente de pastagem, onde os animais estão sujeitos a desasseio. O problema para a técnica de identificação por pelagem acontece em algumas raças bovinas que não possuem distinção na pelagem, inutilizando assim, a técnica.

Como o problema em questão consiste em identificar rapidamente a quantidade de animais presentes em um pasto, por exemplo, o uso de uma imagem com todos os animais se mostra adequada como uma técnica não invasiva. Assim, este trabalho apresenta um classificador para detecção facial frontal bovina, com ênfase para a raça dos Nelores, capaz de identificar e contar a quantidade de faces presentes na imagem.

2. PROBLEMA E SOLUÇÃO

Dependendo do tamanho da criação de bovinos, averiguar e vistoriar os animais pode ser uma tarefa trabalhosa e que demandará tempo dos envolvidos nesse processo, exigindo suporte de colaboradores para assegurar a situação (presença e segurança) do gado. Com isso, o proprietário terá diversas despesas com recursos humanos, podendo existir casos em que serão extraviados recursos da propriedade.

O uso dos brincos com chips atrelados à rastreabilidade do animal pode ser invasivo. Estresse e infecção, podem ser problemas da utilização desta técnica, ocasionando em uma má produção lactínea e perda de peso para o animal, reduzindo assim o lucro para o proprietário.

Logo, uma contagem automática em tempo real, realizada a partir da detecção facial, por meio de imagens, mostra-se promissora, capaz de mitigar problemas como o estresse sobre o animal e minimizar roubos.

3. FUNDAMENTAÇÃO TEÓRICA

O estudo da visão computacional mostra que é possível mapear um dos principais sentidos do Homem, a visão, para valores matemáticos que fazem sentido para a máquina, possibilitando assim, aplicações sobre imagens.

3.1 Reconhecimento de Padrões

O reconhecimento de padrões é um subsídio da visão computacional, responsável por detectar padrões dos pixels em determinada área estabelecida para extração de características [6].

3.2 Detecção Facial x Reconhecimento Facial

É comum que algumas pessoas acabem confundindo detecção facial com reconhecimento facial, ao tentar descrever o propósito de alguma atividade computacional relacionada à imagem. Mas, o objetivo das duas difere na sua aplicabilidade, sendo a detecção facial aplicável no objetivo de localização de rostos, ou características que se assemelham a uma face e o reconhecimento facial tem como objetivo a identificação do indivíduo que pertence

aquela face, ao comparar informações extraídas do rosto com as que estão armazenadas, previamente. O trabalho ora descrito está inserido no contexto da detecção facial.

3.3 Técnicas para Detecção Facial

Dentre as técnicas mais comuns utilizadas para detecção facial, estão: Classificador em Cascata de Haar (*Haar Cascade Classifier*), Histogramas Orientados a Gradientes (*Histogram of Oriented Gradients*) e as Redes Neurais Convolucionais (*Convolutional Neural Networks*).

Neste trabalho, foi utilizado o Classificador em Cascata de Haar, proposto no início do século XXI por Paul Viola e Michael Jones [7].

A principal característica do *Haar Cascade Classifier* consiste em gerar os melhores recursos (*features*) fracos a partir dos métodos de Haar [8], recursos esses que são selecionados a partir de um algoritmo AdaBoost para servir de entrada para o próximo “nível” da cascata, gerando assim, ao final, um classificador forte a partir da soma dos fracos [7].

4. METODOLOGIA

A solução para o problema consiste no desenvolvimento de um classificador *Haar Cascade*, baseado na técnica desenvolvida por Paul Viola e Michael Jones, no início do século XXI. No contexto do trabalho ora descrito, o principal objetivo consistiu na detecção facial para a raça Nelore, visto que é a raça predominante no território Brasileiro, com cerca de 80% do gado pertencendo a esta raça [9].

O classificador desenvolvido neste trabalho, passou por 5 etapas de processamento, apelidado por “Praça do Processamento”, em que há 5 (cinco) módulos responsáveis pela geração do resultado final da classificação.

4.1 Ambiente de Desenvolvimento

A técnica de detecção, juntamente com o classificador, foram gerados a partir de um ambiente de desenvolvimento, no qual foi utilizada a linguagem de programação Python e a biblioteca OpenCV (bastante popular na comunidade científica de inteligência artificial e *machine learning*, para trabalhos envolvendo processamento de imagens) [10].

Como o Sistema Operacional utilizado foi o Windows 10 na sua versão PRO, foi utilizada a ferramenta *Haar Training* [11], para facilitar a usabilidade de algumas aplicações do OpenCV, sendo necessário o uso da plataforma de código aberto Anaconda [12], a qual

permite criar um ambiente configurado python necessário para a execução de *scripts* essenciais para a criação do classificador.

O treinamento do classificador foi realizado em uma máquina com as seguintes configurações: Sistema Operacional Windows 10 Pro, processador Core i5 4460, 16 GB RAM DDR3, SSD 512 GB.

4.2 Conjunto de Dados

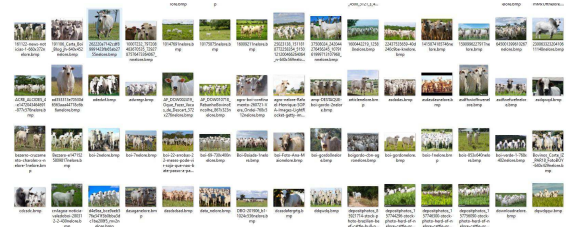
Inicialmente, foram utilizados dois conjuntos de dados disponibilizados pela plataforma Kaggle[13] e um terceiro encontrado no artigo de Brito [14], sendo este descartado por conta da baixa luminosidade das faces, o que prejudica modelos criados com Haar Cascade e também por conta da pouca diversidade dos dados.

O conjunto obtido totalizou 2.087 imagens da espécie bovina. Após a “limpeza” dos dados (removendo duplicatas, prezando por imagens reais e frontais dos animais) restaram 733 imagens.

Com o intuito de aumentar a base amostral para o treinamento, foi dedicado um certo tempo para a obtenção de novas amostras/imagens, prezando pela qualidade e a com o máximo de faces frontais contidas na imagem, sendo estas específicas da raça Nelore, conforme ilustrado na Figura 1. Após o módulo de recorte, explicado mais a frente, a base resultante contou com um total de 1.482 amostras de faces de gado.

Para as imagens negativas, utilizou-se amostras de pastagens/paisagens que não possuíssem nenhum animal da espécie bovina, disponibilizadas também, a partir do Kaggle [15].

Figura 1: Novas imagens da raça Nelore obtidas a partir do Google imagens.

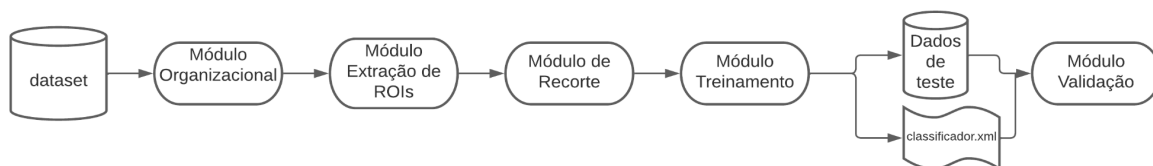


Fonte: Autoria própria, 2022.

4.3 Praça de Processamento

A praça de processamento consiste em módulos, com suas respectivas funções, para a criação do classificador, ilustrada na Figura 2. O processamento se inicia no módulo organizacional, passando pelo módulo de extração da região de interesse (*ROI - Region of interest*), seguindo para o módulo de recorte e treinamento, logo em sequência. O último módulo consiste da validação do classificador obtido no treinamento.

Figura 2: Esteira de processamento do classificador.



Fonte: Autoria própria, 2022.

4.3.1 Módulo Organizacional

Ao receber o *dataset* como entrada, é realizado o primeiro processamento nas imagens, organizando-as de duas formas: renomeando-as e redimensionando seu tamanho.

As imagens foram renomeadas utilizando um contador simples e modificadas para o formato “.bmp”, formato necessário para o software responsável pela extração dos ROIs.

As imagens com tamanho superior a 700x700 pixels, passaram a ficar com tamanho igual a 700x700. Imagens

menores que 700x700 e maiores que 500x500, passaram a ficar com tamanho 500x500 e assim também para as imagens com tamanho 400x400 pixels. Essas dimensões foram definidas com o intuito de diminuir as perdas da qualidade da imagem e também para manter um padrão ao utilizá-las no próximo passo.

4.3.2 Módulo de extração de ROIs

Nesta etapa do processamento, foi utilizado uma ferramenta auxiliadora para a captura das regiões de interesse da imagem, neste caso, a face do animal.

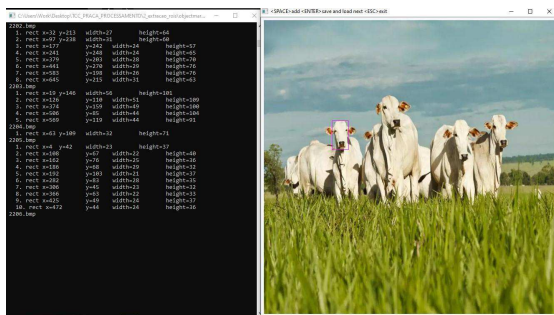
O processo é simples: após o módulo organizacional, são disponibilizadas as imagens no formato “.bmp”,

necessário para que a ferramenta consiga encontrar a imagem; ao executar a ferramenta, o processo de leitura das imagens é inicializado e a imagem é mostrada na tela; com o cursor do mouse é possível desenhar o retângulo para informar que esta é a área de interesse da imagem; após repetir o processo com todas as imagens disponíveis, é gerado um arquivo “info.txt”, em que cada linha do arquivo contém informações sobre a imagem no formato a seguir.

$p \quad nt \quad x_0 \quad y_0 \quad w_0 \quad h_0 \quad x_1 \quad y_1 \quad w_1 \quad h_1 \quad \dots \quad x_i \quad y_i \quad w_i \quad h_i$, sendo $0 \leq i < nt$.

em que, p é a localização da imagem; nt número de retângulos desenhados; x_0 e y_0 são pontos horizontais e verticais iniciais do retângulo; w e h correspondem à largura e à altura do retângulo, respectivamente.

Figura 3: Ferramenta responsável pela extração de ROIs².



Fonte: Autoria própria, 2022.

4.3.3 Módulo de corte

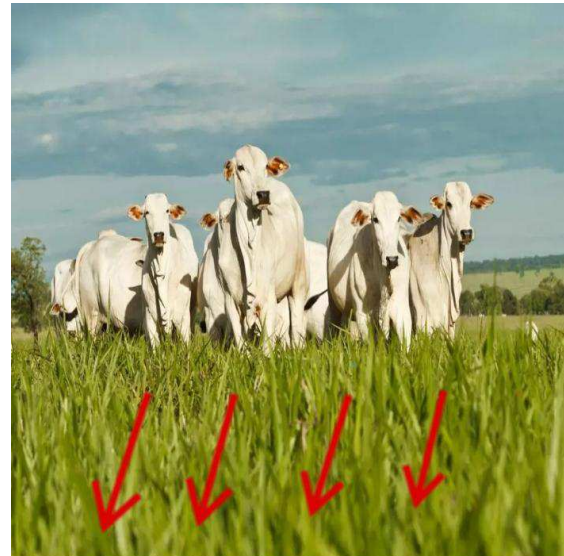
O módulo de corte é responsável pelo corte das imagens organizadas. Ao obter as imagens e o arquivo *info*, o módulo de corte fica responsável por extrair as faces das imagens, originando assim um novo *dataset* de faces.

O corte é feito ao obter a imagem através do seu caminho (p) e, sabendo a quantidade de retângulos desenhados (nt), os próximos valores (x , y , w e h) serão utilizados para obter o retângulo na imagem. Tomando posse desses valores, são extraídos todos os retângulos para geração das novas imagens e criação do novo *dataset faces*³, conforme as Figuras 4 e 5.

² Vídeo tutorial utilizando a ferramenta. Disponível em: <<https://www.youtube.com/watch?v=POSYDLcspIk>>. Acesso em 18 de ago. de 2022.

³ Nome utilizado neste trabalho para o novo *dataset* gerado a partir das imagens de recorte.

Figura 4: Recorte das regiões de interesse.



Fonte: Autoria própria, 2022.

Figura 5: Dataset gerado a partir da extração das faces.



Fonte: Autoria própria, 2022.

4.3.4 Módulo de Treinamento

Para treinar o classificador, é necessário dispor de um conjunto de amostras positivas (contendo as características desejadas) e um conjunto de imagens negativas (não contendo o que se deseja detectar). O conjunto de amostras negativas, foi preparado manualmente a partir do *dataset* público obtido no kaggle (*landscape pictures*), exemplo Figura 6, enquanto o conjunto de amostras positivas foi criado aplicando o *opencv_createsamples*.

O treinamento do classificador foi baseado no tutorial⁴ disponibilizado pela própria biblioteca python OpenCV, utilizando as aplicações fornecidas por esta biblioteca: *opencv_createsamples* e *opencv_traincascade*.

A aplicação *opencv_createsamples* é responsável pela criação do vetor de amostras de imagens utilizado no treinamento. Para o classificador deste trabalho, as imagens do *dataset faces* foram utilizadas como imagens positivas e os parâmetros do vetor são apresentados na Figura 7.

A inicialização do treinamento se dá a partir da aplicação *opencv_traincascade*, utilizando o arquivo “info.txt”, *dataset faces*, como o conjunto de imagens positivas, o vetor gerado, o conjunto de imagens negativas e outros parâmetros informados, conforme ilustrado na Figura 8.

Após a conclusão do treinamento, foi utilizado um *batch file* responsável por converter todos os “nós” gerados na classificação em um arquivo “.xml”, podendo ser aplicado para detecção de faces bovinas para a raça Nelore e utilizado no próximo módulo, para a validação.

Figura 6: Exemplo de imagem negativa.



Fonte: Kaggle Landscape Pictures, 2022.

Figura 7: Parâmetros para criação do vetor, aplicados no *opencv_createsamples*.

```
Info file name: positive/info.txt
Img file name: (NULL)
Vec file name: vector/all_front_faces.vec
BG file name: (NULL)
Num: 1482
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 1.1
Max y angle: 1.1
Max z angle: 0.5
Show samples: FALSE
Width: 24
Height: 36
```

Fonte: Autoria própria, 2022.

Figura 8: Parâmetros utilizados no *opencv_traincascade*.

```
Data dir name: cascades
Vec file name: vector/all_front_faces.vec
BG file name: negative/bg.txt
Num pos: 1482
Num neg: 741
Num stages: 30
Num splits: 1 (stump as weak classifier)
Mem: 8192 MB
Symmetric: TRUE
Min hit rate: 0.995000
Max false alarm rate: 0.500000
Weight trimming: 0.950000
Equal weights: FALSE
Mode: ALL
Width: 24
Height: 36
Max num of precalculated features: 0
Applied boosting algorithm: GAB
Error (valid only for Discrete and Real AdaBoost): misclass
Max number of splits in tree cascade: 0
Min number of positive samples per cluster: 500
Required leaf false alarm rate: 9.31323e-010

Tree Classifier
Stage
+---+
| 0 |
+---+

Number of features used : 294874
```

Fonte: Autoria própria, 2022.

4.3.5 Módulo de Validação

O módulo de validação é responsável pela validação do classificador construído, utilizando um conjunto de imagens para testes e um arquivo *info_teste.txt*, contendo a informação de quantas faces bovinas há na imagem. O módulo gera, então, um log contendo duas precisões: precisão total e acurácia. A precisão total corresponde às chances de detecção de face bovina na imagem e à acurácia representa as chances do modelo detectar a quantidade de faces que foi anotada para a imagem.

A precisão total é calculada conforme a Equação 1.

$$\text{precisão total} = \text{qtd} / \text{qti}, \quad (1)$$

em que *qtd* é a quantidade total de detecções e *qti* é a quantidade total de imagens com faces. Essa métrica entrega a porcentagem de detecção para, ao menos, uma face na imagem.

A acurácia é a soma total de acertos de detecção para cada imagem do conjunto de teste, com base no arquivo *info_teste* (Equação 2).

$$\text{acurácia} = \text{qtdr} / \text{qti}, \quad (2)$$

em que *qtdr* representa a quantidade total de detecções reais, ou seja, se uma imagem do teste possui, por exemplo, 4 faces e o modelo acertou essa quantidade, esse acerto é contabilizado. Ao final, a soma de cada detecção real é dividida pela quantidade de imagens presentes no conjunto de testes. Por exemplo, se para um total de 10 imagens, o modelo acertou todas as faces que continham as 9 imagens e errou para 1 delas, sua acurácia é de 90%.

⁴ Tutorial classificador haarcascade do OpenCV. Disponível em: <https://docs.opencv.org/3.4/dc/d88/tutorial_traincascade.html>.

5. RESULTADOS

Para o desenvolvimento de modelos de *machine learning*, um dos principais fatores de aprendizado que influencia o treinamento de um bom modelo é a qualidade e quantidade de dados do *dataset* disponível. Para este trabalho, o conjunto de dados utilizado, por não conter um grande número de amostras, dificultou o aprendizado do modelo.

Apesar do exposto acima, o classificador resultante (`haarcascade_frontface_cownelore_30_24x36.xml`) proporcionou uma precisão total de 96,92% e acurácia de 80,77%, para uma base de testes com um total de 130 imagens (Figura 9).

No primeiro grupo de linhas da Figura 9, são apresentadas as classes (faces) esperadas para a detecção e a quantidade de cada uma das faces contidas no arquivo de teste. Em seguida, são apresentadas as classes

detectadas pelo classificador e o total de faces para cada classe. Classes não encontradas no arquivo de teste, mas detectadas pelo classificador são destacadas, conforme o segundo grupo de linhas da apresentação. Em seguida, é apresentado o nome do classificador. Nas linhas seguintes constam: quantidade total das imagens de testes que foram analisadas, seguida da quantidade total de imagens com faces detectadas e acerto total de imagens com faces detectadas. Essas informações foram utilizadas para o cálculo da precisão total e acurácia.

Alguns exemplos da detecção correta são mostrados na Figura 10 e exemplos de falha de detecção são exibidos na Figura 11.

Figura 9: Resultado do classificador para a base de testes (código disponibilizado no github).

```
=====//=====
Total para a classe 2 é de 32
Total para a classe 1 é de 45
Total para a classe 6 é de 3
Total para a classe 4 é de 15
Total para a classe 3 é de 24
Total para a classe 7 é de 4
Total para a classe 5 é de 5
Total para a classe 8 é de 1
Total para a classe 9 é de 1
Total detectado para a classe 2 é de 31
Total detectado para a classe 1 é de 45
Total detectado para a classe 5 é de 9
Total detectado para a classe 4 é de 14
Total detectado para a classe 3 é de 22
Total detectado para a classe 7 é de 3
Total detectado para a classe 0 é de 3
Total detectado para a classe 8 é de 1
Total detectado para a classe 6 é de 1
Total detectado para a classe 10 é de 1

=====//=====

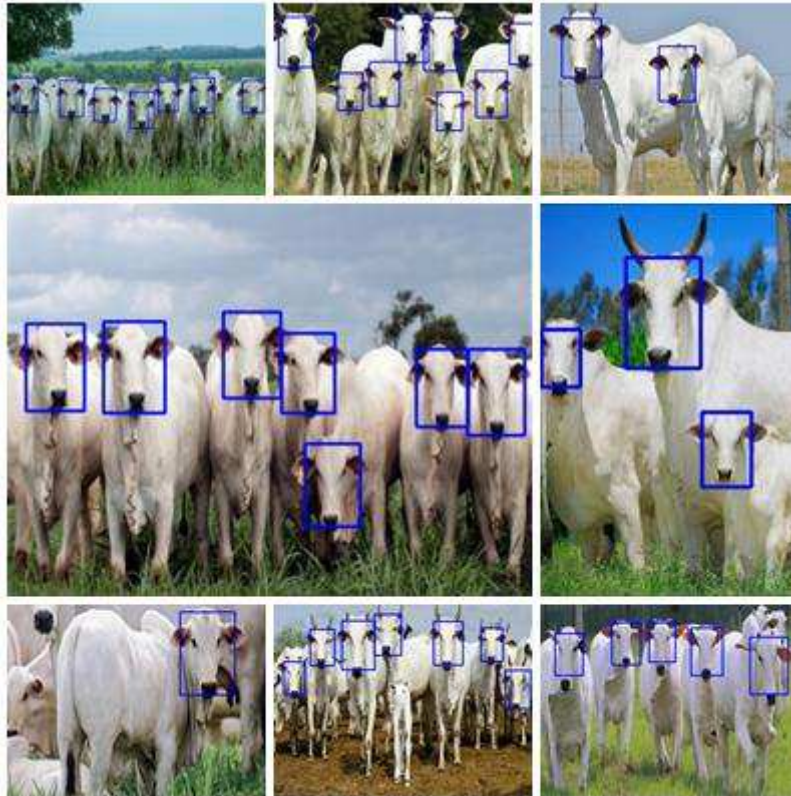
A classe 0 não pertence à classe de teste
A classe 10 não pertence à classe de teste

=====//=====

Classificador utilizado: haarcascade_frontface_cownelore_30_24x36.xml
Quantidade total de imagens analisadas: 130
Quantidade total de imagens com faces detectadas: 126
Acerto total de faces em imagens: 105
Precisão Total = 96.92% (Certeza de detecção para ao menos uma face)
Acurácia = 80.77% (Certeza da quantidade de faces encontradas seja a esperada)
```

Fonte: Autoria própria, 2022.

Figura 10: Resultado do classificador da detecção esperada em algumas imagens de teste.



Fonte: Autoria própria, 2022.

Figura 11: Falha do classificador na detecção de faces bovinas.



Fonte: Autoria própria, 2022.

6. EXPERIÊNCIA

Nesta seção, será discutida a experiência adquirida ao longo do processo de desenvolvimento da técnica e do classificador, assim como os desafios encontrados, as limitações do classificador e as melhorias futuras propostas.

6.1 Processo de Desenvolvimento

Inicialmente, foi estudado sobre o tema de detecção facial aplicado em humanos, visto que é bastante popular e utilizado em diversas aplicações, com o intuito de analisar as técnicas já existentes para tomar como base. Também foi realizado estudo de artigos acadêmicos que possuíam ligações com a técnica. Após o período de entendimento, foi necessário um tempo para a coleta e manipulação dos dados, utilizando das tecnologias conhecidas. Finalizando assim, no tempo do treinamento, utilizando diversos parâmetros para encontrar a melhor acurácia do modelo.

Infelizmente, o aprimoramento do classificador ficou limitado a um máximo de 30 estágios, pois o poder computacional disponível não era suficiente para aumentar o número de estágios, pois cada estágio adicional aumentava o tempo necessário de processamento exponencialmente.

6.2 Principais desafios

Por nunca ter feito algo relacionado à detecção de faces, entender o processo de desenvolvimento levou algum tempo, estudando a melhor metodologia e a melhor técnica base, dadas às informações disponíveis no momento, para o treinamento do modelo.

Outro desafio foi a montagem do ambiente de desenvolvimento, pois poucos trabalhos envolvendo a biblioteca OpenCV foram desenvolvidos no sistema operacional Windows.

Em relação ao classificador, o principal desafio foi encontrar o *dataset* específico para o problema, contendo amostras com boa qualidade e divergentes entre si.

6.3 Limitações

Apesar dos resultados positivos do classificador, é importante ressaltar que o mesmo possui limitações, destacadas a seguir.

- A raça do animal bovino a ser analisado deve ser, preferencialmente, Nelore.
- A face dos animais a serem analisadas devem estar totalmente viradas para a câmera de captura (imagens frontais).

- A posição da câmera deve estar relativamente na altura do animal.
- Os animais devem estar a uma distância média da fonte de captura (A área da face não pode representar mais que 70% da imagem e menos que 2,5% dela, contando que o tamanho mínimo da imagem seja de 300x300)

Estas limitações/restrições são necessárias para garantir a precisão da detecção das faces, caso contrário a técnica não retratará o esperado.

Mesmo diante desses desafios e limitações, a abordagem proposta se mostrou eficiente para a detecção e contagem de faces bovinas, de forma automática, o que poderá representar uma ferramenta de auxílio importante a ser usada na contagem de rebanhos bovinos.

6.4 Propostas para Melhorias Futuras

Para o classificador atual, uma melhoria estaria relacionada justamente à obtenção de novos dados amostrais, com boa qualidade e diversidade.

Para a técnica de detecção facial bovina, seria pertinente também criar outros classificadores capazes de detectar as faces laterais do animal, tanto a direita como a esquerda, assim seria possível não limitar a contagem apenas a uma posição do animal. Outra característica de melhoria consistiria em expandir a base amostral para outras raças bovinas, não limitando apenas às raças Nelore.

Para a validação do classificador, uma melhoria seria sua implementação com base nas anotações realizadas pelo desenvolvedor, de forma subjetiva, que iria observar o retângulo desenhado na imagem e verificar se a detecção do classificador se aproxima ou está contida na área anotada.

7. AGRADECIMENTOS

Gostaria de agradecer, primeiramente, a Deus por ter permitido que este trabalho fosse realizado, pois sem ele nada seria. A minha família, principalmente aos meus pais José Ramalho e Neide Rodrigues, por todo o amparo e dedicação, a minha noiva Mabel Myslane, pela motivação incondicional. A minha excelentíssima professora orientadora Joseana Macêdo Fachine Régis de Araújo, por aceitar esse desafio e me auxiliar sempre que requisitado. Gostaria de agradecer também a todos os meus amigos da graduação, que contribuíram de alguma forma com a minha formação, em especial: Paulo Mendes, Arthur Dantas, Gutemberg da Silva, Samuel Vasconcelos, Abel Antunes e Matheus Araujo.

8. REFERÊNCIAS

[1] IBGE - Painel de Indicadores - Indicadores Agropecuários - Bovinos. Disponível em: <<https://www.ibge.gov.br/indicadores#variacao-do-pib>> Último acesso em 20 de agosto, 2022.

[2] CICARNE - A pecuária de corte mundial em números - Boletim 48. Disponível em: <<https://www.cicarne.com.br/wp-content/uploads/2021/07/Boletim-CiCarne-48-2021.pdf>>. Último acesso em 20 de agosto, 2022.

[3] ABNT - Norma Técnica - NBR 14766:2012 - Identificação de animais por radiofrequência <https://www.abntcatalogo.com.br/norma.aspx?Q=VDV0ZU5aWm16Z3NYM3poUDAxYm1rdDE2YTVkOCtTTBxk3U4Vjk4M0piYz0=> Último acesso em 26/03/2022.

[4] Zin. T.T. Image Technology based Cow Identification System Using Deep Learning, 2018.

[5] Kumar, S., Singh, S.K., Singh, R.S. et al. Real-time recognition of cattle using animal biometrics. J Real-Time Image Proc 13, 505–526 (2017).

[6] Reconhecimento de padrões. Disponível em: <https://en.wikipedia.org/wiki/Pattern_recognition>. Acesso em 22 de agosto, 2022.

[7] Paul Viola and Michael J. Jones. Robust real-time face detection. International Journal of Computer Vision, 57(2):137–154, 2004.

[8] Transformada de Haar. Disponível em: <https://pt.wikipedia.org/wiki/Transformada_de_Haar>. Acesso em 19 de agosto, 2022.

[9] Histórico Nelores. Disponível em: <<http://www.nelore.org.br/Raca/Historico>>.. Acesso em 21 de agosto, 2022.

[10] OpenCV introduction. Disponível em: <<https://docs.opencv.org/4.x/d1/dfb/intro.html>>. Acesso em 03 de junho, 2022.

[11] Haar Cascade Training Materials. Disponível em: <<https://www.youtube.com/watch?v=POSYDLcspk>>. Último acesso em 21 de agosto, 2022.

[12] ANACONDA DISTRIBUTION. Disponível em: <<https://www.anaconda.com/products/distribution>>. Acesso em 01 de julho, 2022.

[13] Animals classifier & cow-images. Disponível em: <<https://www.kaggle.com/datasets/alessiocorrado99/animals10>> <<https://www.kaggle.com/datasets/afnanamin/cow-images>>.

Último acesso em 20 de agosto, 2022.

[14] BRITO, A. Reconhecimento facial bovino: Uma alternativa aos métodos tradicionais de rastreamento. Trabalho de Conclusão de Curso. Universidade de Caxias do Sul, 2021.

[15] Landscape Pictures. Disponível em: <<https://www.kaggle.com/datasets/arnaud58/landscape-pictures>>. Acesso em 15 de julho, 2022.