Pró-reitoria de Pesquisa e Pós-Graduação
INSTITUTO FEDERAL São Paulo Câmpus Cubatão

12º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2021

USO DA VISÃO COMPUTACIONAL PARA DETECÇÃO DE BEZERRAS LEITEIRAS

JULIO CESAR SCHENDROSKI [1], IRAN J. OLIVEIRA DA SILVA[2], THIAGO LUÍS LOPES SIQUEIRA[3]

[1] Graduando em Engenharia de Computação, PIBIFSP, IFSP Câmpus Piracicaba, julio.schendroski@aluno.ifsp.edu.br
[2] Docente, colaborador, Departamento de Engenharia de Biossistemas, ESALQ/USP, iranoliveira@usp.br
[3] Docente, orientador, IFSP Câmpus Piracicaba, prof.thiago@ifsp.edu.br
Área de conhecimento (Tabela CNPq): 1.03.03.04-9 Sistemas de Informação

RESUMO: Diversificadas soluções tecnológicas para agropecuária têm sido desenvolvidas devido à transformação digital pela qual ela vem passando. Nesse contexto, soluções envolvendo a visão computacional têm relevância. Este artigo aborda o uso de visão computacional para detecção de bezerras leiteiras a partir de imagens obtidas por meio de smartphone. Descreve-se ainda o programa de computador criado e resultados preliminares do processamento das imagens retratando esses animais.

PALAVRAS-CHAVE: processamento de imagens; pecuária leiteira; Python; OpenCV

USING COMPUTER VISION FOR DETECTING HEIFERS

ABSTRACT: Technological solutions for agriculture and livestock have been developed due to the digital transformation happening in the field. In this context, solutions involving computer vision are relevant. This paper describes the use of computer vision for detecting heifers whose pictures were taken using a smartphone. It also details the implemented software and preliminary results obtained from image processing.

KEYWORDS: image processing; dairy cattle; Python; OpenCV

INTRODUÇÃO

A transformação digital da agropecuária vem aumentando a procura por diversificadas soluções que possam ser acessadas via computadores ou *smartphones*. No sentido de suprir essa demanda, plataformas digitais têm sido desenvolvidas por diversificadas instituições, tais como centros de pesquisa, universidades, grandes empresas, *startups*, cooperativas e associações (MASSRUHÁ et al., 2020). Essas soluções integradas e inovadoras são baseadas, dentre outros recursos, na visão computacional (MOTA; CASTRO JUNIOR; SILVA, 2021).

Por exemplo, a detecção de gado bovino em pastagens pode ser viabilizada com acurácia a partir de imagens obtidas por veículo aéreo não-tripulado quadrirrotor, voando em alturas mais elevadas e cobrindo mais áreas em menos tempo, e apesar de situações adversas (e.g. oclusão por árvores, diferenças de iluminação, condições da pastagem e posicionamento dos animais) (BARBEDO et al., 2019). Noutro exemplo, detectam-se bovinos em ambiente não controlado de pastagem, a partir de imagens disponíveis em repositórios *online* (e.g. ImageNet e Google Image) e nas quais apenas o animal foi selecionado para minimizar falsas detecções (ALMEIDA; FREITAS; SIQUEIRA, 2019).

Já neste artigo, objetivou-se aplicar métodos e técnicas da visão computacional sobre imagens de bezerras leiteiras, com intuito de detectar esses animais. As imagens foram obtidas majoritariamente a partir de *smartphone* num bezerreiro experimental. Para implementação do código-fonte foi usada a linguagem de programação Python e a biblioteca OpenCV. Nas seções seguintes, são descritos materiais e métodos empregados, discutidos resultados obtidos e apontadas as conclusões.

MATERIAIS E MÉTODOS

A visão computacional reúne um conjunto de métodos e técnicas que permite ao computador captar e interpretar imagens, utilizando embasamento matemático ao viabilizar a criação de possíveis soluções. Segundo Szeliski (2011), na visão computacional objetiva-se descrever o mundo que se observa em uma ou mais imagens para interpretação de suas propriedades, formatos, iluminações e distribuição de cor utilizando como base modelos físicos. Comumente, a interpretação computacional da imagem permite o reconhecimento de informações não possíveis aos olhos humanos.

O conjunto de dados utilizado foi composto por 10 fotografias de bezerras leiteiras, sendo 8 delas tiradas em um bezerreiro experimental usando-se um *smartphone,* e as outras 2 obtidas a partir de repositório *online*. Os animais estão posicionados de lado nessas imagens. Apenas 3 imagens retratam apenas o animal sem a presença de outros objetos (exceto fundo e sombra).

Para implementação de algoritmos da visão computacional e o processamento das imagens, foi utilizada a linguagem de programação Python na versão 3.8.8 (ROSSUM et al., 2021) e a biblioteca OpenCV na versão 4.5.2 (BRADSKY, c2021). Essas versões foram escolhidas por serem as mais recentes compatíveis entre si. Além disso, foram empregados também nas implementações o pacote de linguagem Numpy para manipular os conjuntos de matrizes (https://numpy.org/doc), e a biblioteca Matplotlib para permitir a visualização dos resultados do processamento da imagem (https://matplotlib.org). Foi empregado o Windows 10 como sistema operacional de 64 bits, o ambiente de desenvolvimento Visual Studio Code na versão 1.58.2, e o GitHub como repositório para gerenciamento de versões de código-fonte.

A seguir, são sumarizados métodos e técnicas de visão computacional que foram considerados neste trabalho. Para tal, foram consideradas as definições apresentadas por Szeliski (2011) e suas correspondentes aplicações usando Python e OpenCV segundo Bradsky (c2021) e Antonello (2020).

Uma imagem colorida é associada a uma matriz tridimensional e 3 canais (numerados de 0 a 2), ou uma matriz bidimensional no caso da imagem preta e branca com apenas 1 canal (numerado 0). Na linguagem de programação Python, os números inteiros que compõem as matrizes, foram definidos pelo tipo de dados *uint8* já que assumem valores dentro do intervalo de 0 até 255. A representação de um elemento de determinada cor dentro de uma matriz tridimensional foi (0,0,0) para preto, e (255,255,255) para branco em cada canal, ou seja, valores mínimos e máximos de *uint8*, respectivamente. Para obter os canais de cores, empregou-se o sistema BGR (*blue*, *green*, *red*) e valores para cada elemento da lista. Por exemplo, a cor vermelha assumindo (0, 0, 255) e o valor máximo para vermelho (*red*), e a cor magenta com (255, 0, 255) e os valores máximos para azul (*blue*) e vermelho (*red*), respectivamente.

É possível manipular cada elemento na matriz, denominado pixel. Para acessar um pixel, especifica-se as coordenadas específicas da localização na matriz. E os canais da imagem, sendo o primeiro canal representando a altura da imagem, o segundo canal a largura, e o terceiro para representar as cores, respectivamente. Obtido o pixel desejado, é possível alterar a cor e até mesmo seu posicionamento na imagem. O redimensionamento de uma imagem redefine as suas proporções, atribuindo-se novos valores para altura e largura.

A suavização da imagem (*blur*), consiste em aplicar um efeito de desfoque a partir de uma vizinhança que circunda determinado pixel, a fim de reduzir o ruído. O método *Median Blur* realiza uma suavização através do resultado dos valores médios da vizinhança. A binarização é a operação que permite a interpretação da imagem de forma binária. Porém em visão computacional, não é definida por 0 e 1, e sim por 0 e 255, já que trata dos extremos preto e branco, respectivamente.

Limiarização é um processo que usa valores que representam limites mínimo e máximo, e cada pixel é comparado com esses limites. Caso o valor do pixel seja abaixo ou acima dos limites, assumem valores 0 e 255, respectivamente. O valor de limiar pode ser definido de forma automática ou de forma manual (vide seção seguinte). Pode ser definido de forma adaptativa ou de forma global.

O limiar de uma imagem antecede a identificação de contornos de objetos. Um contorno pode ser expresso por uma variação muito alta e de forma repentina entre (o valor de) um pixel para o outro. Ou seja, existe contorno entre pixels com valor de intensidade muito alto e que abruptamente assumem valores baixos. Um método para identificação de contornos comumente usados, e também considerado neste trabalho, é o proposto por Canny (1986). Ele utiliza de vetores gradientes para indicar qual região da imagem sofre maiores variações abruptas. Dois valores, denominados supressão máxima e mínima, são limiares nos quais os pixels identificados por gradientes são comparados. Usando funções do OpenCV é possível desenhar os contornos identificados em uma imagem limiarizada.

Com transformações morfológicas é possível alterar as formas de uma imagem. O fechamento pode eliminar pequenos buracos suavizar contornos (mas fundir partes). A dilatação pode ser usada visando aumentar a área do objeto de interesse.

A segmentação é um método utilizado para identificar objetos baseado em segmentar a imagem por regiões e diferenciá-las através de uma escala de cores. Um dos algoritmos de segmentação é denominado *watershed* (MEYER, 2012) e promove a diferenciação do fundo da imagem e a definição de camadas contendo objetos no plano frontal. As camadas não utilizáveis para reconhecimento dos contornos são definidas como 0. Já as camadas com a mesma tonalidade de cor pertencem ao mesmo objeto.

RESULTADOS E DISCUSSÃO

Considerando-se os métodos e as técnicas mencionadas na seção anterior, foi implementado um programa de computador usando-se Python e OpenCV (junto às bibliotecas supracitadas). Uma representação gráfica desse programa é ilustrada na FIGURA 1, considerando-se todas as etapas de processamento pelas quais cada imagem é submetida.

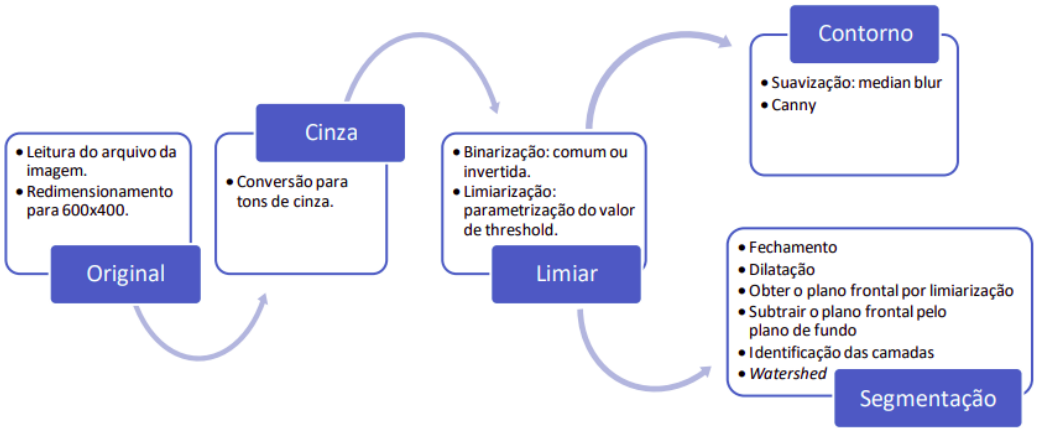

FIGURA 1. Representação gráfica do programa de computador implementado.

Primeiramente, o arquivo da imagem é lido a partir do armazenamento local e redimensionado para 600x400 pixels. Numa segunda etapa, essa imagem, originalmente RGB, é convertida para tons de cinza com intuito de reduzir a quantidade de informações redundantes na imagem, facilitando-se identificar regiões com intensidades de cores menores ou maiores.

Na terceira etapa, emprega-se a limiarização da imagem com o principal intuito de separar o objeto de interesse do plano de fundo, transformando a imagem em cores binárias (preto e branco). Tal etapa é necessária para identificação de contornos e objetos na imagem. Para realizar a limiarização foram propostos valores de parâmetros para o limite mínimo conforme mostrado na TABELA 1. Esses valores foram ajustados com intuito de beneficiar as etapas seguintes e obter o mínimo de objetos de desinteresse na imagem. Esses valores foram selecionados verificando-se a tonalidade dos pixels nas em regiões que separavam o animal e o fundo da cena, e são associados a características das imagens tais como cor da pelagem dos animais, reflexo da luz do ambiente sob a pelagem, e existência de outros objetos. Valores abaixo de 50 foram associados a imagens com pouca influência dessas características, já os maiores foram àquelas com maior influência dessas características.

Após a limiarização são possíveis duas alternativas para prosseguimento: a identificação de contornos ou a segmentação. Na identificação de contornos, almeja-se contornar o corpo do animal e separá-lo dos demais objetos da imagem. A partir da imagem limiarizadas poder reconhecer contornos

verdadeiros que pertencem ao animal, ou contornos falsos que fazem parte de ruídos e objetos indesejados. Na segmentação, deseja-se preencher o corpo do animal e separá-lo dos demais objetos da imagem. Considerando que o animal está no plano frontal da imagem e os objetos desinteressantes no plano de fundo, uma subtração é feita (frontal - fundo). Então são obtidas camadas para cada objeto na imagem, buscando-se determinar qual camada pertence ao animal. Por fim, uma cor específica distingue a bezerra dos demais objetos.

TABELA 1. Valores de parâmetros para limite mínimo na limiarização.

Imagem	1	2	3	4	5	6	7	8	9	10
Valor	15	40	30	50	40	50	68	30	80	47

A execução do programa considerando-se as dez imagens fornecidas como entrada gerou, como saída, uma imagem referente a cada etapa e conforme mostrado na FIGURA 2. São discutidos principalmente, a seguir, os resultados da identificação de contorno e da segmentação.

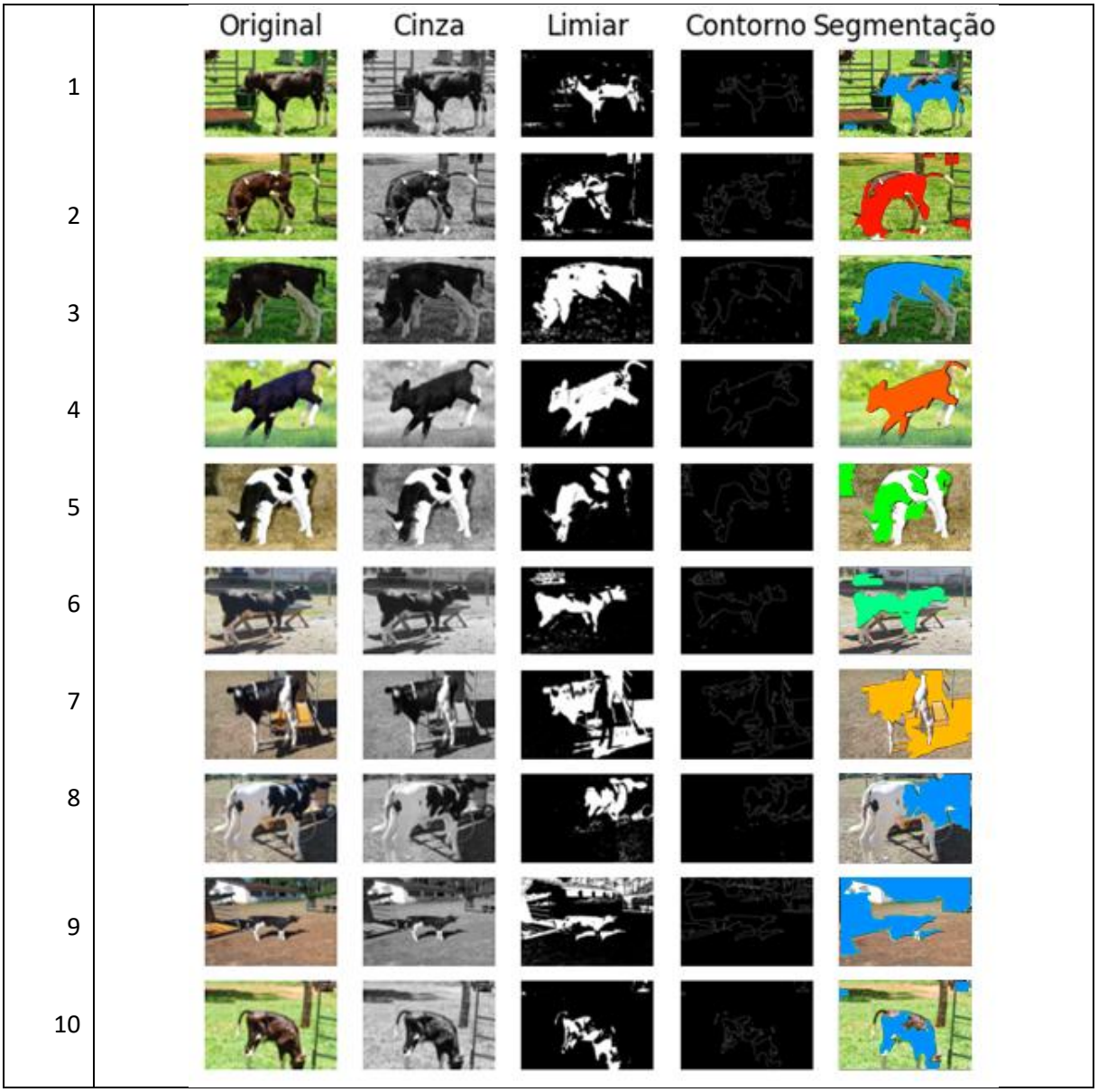

FIGURA 2. Imagens originais e processadas.

O uso de um limiar próximo a tonalidade da cor preto pode interferir quando há sombras do animal ou da cena. Por exemplo, nas imagens 6, 7, 8 e 9, em que o valor mais escuro da pelagem

desses animais é menor que os valores das sombras. Contudo, os valores não são uniformes em cada animal, pois há incidência de luz nele, e/ou sua pelagem tem mais de uma cor. Com relação aos contornos, aqueles identificados pelas imagens 3, 4 e 6 são fechados e incluem parte significativa dos corpos. Na imagem 8, o corpo e sua sombra são contornados, mas também existem inúmeros contornos que não pertencem ao corpo. Na imagem 10, embora parte do corpo tenha sido contornada, partes do corpo onde a luz refletiu não foram contornadas. Já na segmentação das imagens 1, 2, 3, 4, 6 e 10 notou-se o preenchimento dos corpos dos respectivos animais, distinguindo-os razoavelmente dos demais objetos. Por sua vez, nas imagens 5, 7 e 8 não foram alcançados resultados satisfatórios para detectar os animais, devido às cores (5) e à presença de outros objetos na cena (7 e 8).

CONCLUSÕES

Considerando a transformação digital do campo e sua demanda por soluções tecnológicas, neste artigo foi abordado o uso da visão computacional para detecção de bezerras leiteiras. Foram usadas imagens obtidas num bezerreiro experimental e capturadas a partir de smartphone. Descreveu-se também o programa de computador implementado e foram apresentados resultados obtidos a partir do processamento das imagens. Considerando-se o conjunto de dados existente, a detecção pôde ser considerada satisfatória em 70% das imagens. São indicativos de trabalhos futuros: aperfeiçoamento dos algoritmos implementados; aumentar o volume do conjunto de dados (e.g. quantidade de imagens); triagem ou recorte ou rotulação das imagens (TREVISAN; SIQUEIRA, 2019); uso da matriz de confusão para detectar as porcentagens de acerto, erro, falso positivo e falso negativo. Uma vez viabilizada maior acurácia no processamento das imagens, poderá ser investigada a viabilidade de associação a métodos aprendizado de máquina (LUGER, 2013).

AGRADECIMENTOS

Ao PIBIFSP 2021 pela bolsa discente. Ao NUPEA/ESALQ-USP por obter e ceder imagens.

REFERÊNCIAS

ALMEIDA, A. A.; FREITAS, P. D.; SIQUEIRA, A. A. **Detecção de bovinos em ambiente de pastagem através de visão computacional**. 2019. 101 f. TCC (Graduação) - IFG, Inhumas, 2019.

ANTONELLO, R. **Introdução a Visão Computacional com Python e OpenCV**. Disponível em: http://cv.antonello.com.br/visao-computacional/. 2018. Acesso em: 02 ago. 2021.

BARBEDO, J. G. A.; KOENIGKAN, L. V.; SANTOS, T. T.; SANTOS, P. M. A study on the detection of cattle in UAV images using deep learning. **Sensors**, v. 19, n. 24, 5436, Dec 2019.

BRADSKI, G. Image Processing. c2021. Disponível em: https://docs.opencv.org/4.5.2/d7/dbd /group__imgproc.html. Acesso em: ago. 2021

CANNY, J. A Computational approach to edge detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 8, n. 6, 1986. p. 679-698.

MASSRUHÁ, S. M. F. S.; LEITE, M. A. de A.; OLIVEIRA, S. R. de M.; MEIRA, C. A. A.; LUCHIARI JUNIOR, A.; BOLFE, E. L. (Ed.). **Agricultura digital**: pesquisa, desenvolvimento e inovação nas cadeias produtivas. Brasília, DF: Embrapa, 2020.

LUGER, G. Inteligência artificial. 6. ed. Pearson, 2013.

MEYER, Fernand. The watershed concept and its use in segmentation: a brief history. **arXiv preprint arXiv:1202.0216**, 2012. Cornell University.

MOTA, M. O.; CASTRO JUNIOR, S. L.; SILVA, I. J. O. da. Avaliação automática da porosidade de ovos utilizando técnicas de processamento digital de imagens. *In:* AVESUI, 19., 2020. Evento virtual. **Anais do 5º Congresso Brasileiro de Zootecnia de Precisão e 19º Seminário Técnico-Científico de Aves, Suínos, Bovinos e Peixes**, 2021. 4 p. Disponível em: https://data.avesui.com/file/2021/04/ 19/H140700-F00000-V362.pdf . Acesso em: ago 2021.

ROSSUM, G. van et al. **Python Tutorial**: release 3.8.12. Disponível em: https://docs.python.org/3.8/ archives/python-3.8.12-docs-pdf-a4.zip . Acesso em: set. 2021.

SZELISKI, R. **Computer Vision**: algorithms and applications. Springer-Verlag London. 2011. 812 p.

TREVISAN, V.; SIQUEIRA, T. L. L.. Labeling images with the Labellt! app. *In*: SBIAgro, 12, 2019. Indaiatuba. **Anais do XII Congresso Brasileiro de Agroinformática**, 2019. p. 272-281.