Aplicação de algoritmos de Visão Computacional na contagem de gado por meio de processamento de imagens aéreas

Nathan G. V. Ribeiro[1], Gustavo Bartz Guedes[1], Tamires T. Barbieri[1]
[1]Instituto Federal de Educação Ciência e Tecnologia de São Paulo, Campus Hortolândia
CEP: 13183-250 – Hortolândia – SP – Brasil

nathanpirral@gmail.com, {gubartz, tamires.barbieri}@ifsp.edu.br

Abstract. *This paper describes the process of developing a software that makes use of Computational View algorithms in cattle counting through images collected in public image banks. Image processing is performed by computational view and image processing libraries available in the Python programming language. The justification for this work is in the relevance that statistical analysis of data has to the livestock sector when applied to general management, vaccination and disease control processes, especially due to the constant growth of the exports of products generated by livestock in Brazil.*

Resumo. *O presente artigo descreve o processo de desenvolvimento de um programa que aplica algoritmos de Visão Computacional na contagem de gado por meio de imagens aéreas coletadas em bancos de imagens públicas. O processamento das imagens é efetuado por bibliotecas de visão computacional e de processamento de imagens disponíveis para a linguagem de programação Python. A justificativa para desenvolvimento deste trabalho está na relevância da análise estatística dos dados coletados por pecuaristas aplicados na administração dos diversos processos de manejo, vacinação e controle de doenças em gados especialmente devido ao constante crescimento da exportação de produtos gerados pela pecuária no Brasil.*

1. Introdução

Segundo o IBGE (2016) a pecuária no Brasil tem se expandido nos últimos 45 anos e o país se tornou líder mundial em exportação de produtos gerados pela pecuária. Sua representatividade no Produto Interno Bruto (PIB) é de 5,5% sendo maior que os setores de transporte e mineração no último censo realizado em dezembro de 2016.

Segundo os dados divulgados pela Organização das Nações Unidas para Agricultura e Alimentação (FAO), no período de 1970 a 2014 o tamanho de rebanhos no Brasil quadruplicou e se tornou o maior no planeta (desconsiderando a Índia, em que bovinos não são criados com fins comerciais). A produção de carne teve um aumento ainda mais expressivo tendo seu volume sextuplicado de 1970 a 2014.

Apesar da grande expansão do setor de pecuária nos últimos anos, a produtividade tem sido a maior barreira do país para avançar no mercado e gerar lucros maiores. De acordo com Amaral et al. (2012), o Brasil não se encontra entre os 10 países mais produtivos do setor de pecuária de corte (abate de animais para produção de carne) portanto há necessidade de criação de processos de pecuária de precisão dado o tamanho dos rebanhos existentes.

A necessidade global por alimentos vai exigir cada vez mais produtividade nos processos agrícolas e pecuários. Para que isto ocorra, o uso de tecnologias como o processamento de imagem, redes de sensores sem fio e Internet das Coisas são essenciais para aprimorar a gestão operacional de fazendas e garantir o crescimento da produtividade, necessários para a expansão do mercado pecuarista brasileiro.

Este trabalho teve como objetivo desenvolver um programa na linguagem Python, que por meio de aplicação de filtros de processamento de imagem e a utilização de algoritmos de visão computacional, consiga realizar a contagem de gado, tanto em ambientes de confinamento como em campo aberto.

Este artigo é composto de uma apresentação dos trabalhos correlatos descrevendo as soluções a problemas similares na identificação de animais em fotografias, um capítulo com a fundamentação teórica descrevendo todos os processos, filtros afins de contextualizar o leitor sobre os componentes do programa desenvolvido. Uma seção com a descrição da metodologia empregada no desenvolvimento assim como um capítulo para a apresentação dos resultados do software. Após a análise e discussões dos resultados, são apresentadas as conclusões e descobertas da pesquisa, evidenciando as deduções extraídas com o trabalho e ainda introduz possíveis melhorias a serem aplicadas em trabalhos futuros.

2. Trabalhos correlatos

Existem diversos trabalhos na linha de pesquisa focada no reconhecimento de objetos e sua contagem utilizando algoritmos de visão computacional, assim como a aplicação de veículos aéreos não tripulados aplicados a contagem e monitoramento de gado. Chamoso et. Al. (2014) apresenta uma abordagem utilizando redes neurais convolucionais [1]com o objetivo de solucionar problemas de reconhecimento de imagens de diferentes tipos de entrada, problema este que foi detectado durante o desenvolvimento do programa.

A pesquisa de Howse (2014) apresentam uma solução mais genérica em que é executado o processamento de vídeo e de reconhecimento de padrões com a utilização de detectores pré-treinados para identificação de humanos e gatos durante a análise de vídeo.

Outros dois trabalhos relevantes são os estudos de identificação de objetos e de contagem de humanos realizados por Badham (2016) e Bhumika et. Al. (2017) que trouxeram inúmeras informações sobre as ferramentas que a biblioteca OpenCV disponibiliza para os trabalhos de processamento de imagens assim como realizar a contagem de objetos neste contexto.

Por fim a dissertação de mestrado de Balan (2003) propõe a implementação de técnicas de segmentação de imagem para a contagem da população de indivíduos pertencentes a uma determinada população de aves por meio do processamento de imagens aéreas, o método de contagem por segmentação implementado em seu trabalho apresentou resultados consideráveis e sua tese propõe como solução final a utilização de segmentação por textura como o método mais adequado para o processo de contagem.

[1] Redes neurais são grupos de neurônios conectados entre si que formam um sistema nervoso, tal como o cérebro humano. As redes neurais artificiais são abstrações do modelo biológico que permitem representar os neurônios como unidades simplistas de computação. Fonte: Guedes (2017).

3. Fundamentação Teórica

Esta seção descreve os principais conceitos acerca dos métodos e conceitos utilizados no trabalho para o processamento das imagens coletadas, definição dos conceitos de visão computacional e suas áreas de pesquisa, assim como expõe as tecnologias escolhidas para a contagem de gado.

3.1. Visão Computacional

Segundo Klette (2014), atualmente a visão computacional tem se tornado uma das principais tecnologias em diferentes linhas de pesquisa e desenvolvimento, sendo uma das grandes áreas da computação e sua principal função é a extração automática de informações de imagens, estas informações podem ser empregadas dentre as diversas subáreas como apresentado na Figura 1.

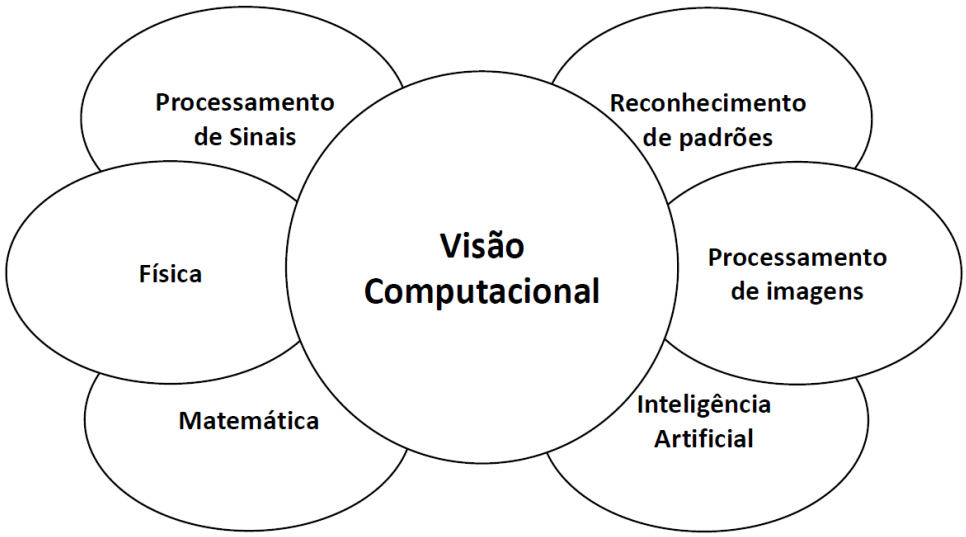

Figura 1. Visão computacional e suas subáreas. Fonte: Klette (2014),elaboração própria.

Na prática, o objetivo da visão computacional é utilizar imagens para a extração de informações e por meio da análise e processamento dos resultados, transformar cenários do mundo real em uma abstração digital do que foi coletado.

Um exemplo, são as interações com a realidade virtual, jogos, aplicações de realidade aumentada, sistemas de assistência de direção de veículos autônomos, segurança inteligente de ambientes, aplicações de gestão de processos industriais, entre outras.

3.2. Imagens digitais

Para as análises realizadas no âmbito da visão computacional, é necessário a utilização de imagens digitais cuja responsabilidade é fornecer a representação do espaço do mundo real em uma representação gráfica digital.

As imagens digitais são compostas pela integração de amostras contínuas e analógicas de informações coletadas do domínio espacial, informações estas, que são interpretadas por uma interface digital responsável por processar o que foi observado no mundo real e as transformar em valores capazes de serem processados por computadores para a geração da representação digital do que foi coletado.

De acordo com Leighton et Al. (1969), o processamento realizado com o objetivo de criar uma imagem digital é feito pela criação de uma matriz composta de pixels (termo que abrevia a denominação para elemento de imagem), sendo estes elementos utilizados para representar a menor porção de informação extraída do mundo real.

Segundo a definição de Gonzalez e Woods (2008), uma imagem pode ser descrita matematicamente por uma função de duas dimensões exemplificada pela Equação 1, em que x e y representam o domínio espacial de uma imagem que é representado pelo símbolo ômega (Ω), as colunas no eixo x contém pontos da matriz $\{(1,x),(2,x), \ldots, (N\mathit{colunas},x)\}$ que representam a posição do pixel cujo valor deve seguir a regra: $1 \leq x \leq N\mathit{colunas}$, e as linhas no eixo y contém os pontos da matriz correspondentes a $\{(1,y),(2,y), \ldots, (N\mathit{linhas},y)\}$ que representam pixels cujo valor segue a regra: $1 \leq y \leq N\mathit{linhas}$. A Figura 2 simboliza a equação graficamente por meio da disposição das duas funções visualmente.

Equação 1. Fórmula matemática de definição de imagem em domínio espacial.

$$\Omega = \{(x,y) : 1 \leq x \leq N_{colunas} \ \wedge \ 1 \leq y \leq N_{linhas}\} \subset \mathbb{Z}^2$$

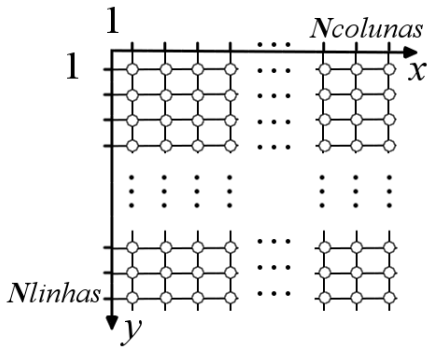

Figura 2. Disposição de uma imagem por meio de sistema de coordenadas.
Fonte: Klette, 2014, tradução própria.

As coordenadas espaciais $(x, y) \in \mathbb{Z}^2$, são compostas de valores pertencentes ao conjunto numérico dos números inteiros e indicam os índices de posição do pixel na imagem.

Ainda é possível associar a este píxel um valor (u) que descreve a cor que o mesmo deve apresentar na imagem a qual ele compõe. A Figura 3 é um exemplo de imagem binária. Neste tipo de imagem digital o valor de um píxel (u) pode ser representado somente por dois valores [0,1], sendo zero a cor preta e um a cor branca.

Figura 3. A noite estrelada de Vincent Van Gogh em imagem binária.
Fonte: Van Gogh, V. (1889). A Noite Estrelada, sob filtro.

Em imagens monocromáticas ou de escalas de cinza este valor de cor (u) é expandido a todas as variações de cor entre o preto e o branco, desta maneira consegue

gerar imagens de maior definição, pois pode conter 256 intensidades diferentes. A Figura 4 é um exemplo de imagem monocromática.

Apesar da maior nitidez, a aplicação deste tipo de imagem em projetos de visão computacional pode ocasionar erros. Em análises de detalhes muito específicos é possível que hajam erros nos arredondamentos das conversões das cores por terem escala de intensidades reduzidas.

Figura 4. A noite estrelada de Vincent Van Gogh em imagem monocromática.
Fonte: Van Gogh, V. (1889). A Noite Estrelada, sob decomposição.

As imagens que garantem maior nitidez e qualidade são as de modelo multicanais (vermelho, verde e azul), nelas o valor (u) é representado por um vetor de três posições, cada um dos índices deste vetor representam um dos canais de cor e são compostos por valores no intervalo de [0, 1, ... , u_{max}]. Em geral neste sistema de cores é possível atribuir 256 variações de cor para cada um dos canais, desta maneira é possível representar mais de 16 milhões de combinações de cores possíveis.

Os valores destes canais podem ser interpretados por meios computacionais e a sua agregação resulta na geração de uma imagem digital colorida e de multicanais. A Figura 5 exemplifica o processo de composição de uma imagem de modelo multicanal em que o quadro superior direito se refere ao canal vermelho, o canal inferior esquerdo ao canal verde e o quadro inferior direito ao canal azul.

Figura 5. Decomposição de canais de uma ilustração.
Fonte: Van Gogh, V. (1889). A Noite Estrelada, decomposição.

3.3. Python

De acordo com a Python Software Foundation (2015), o Python é uma linguagem de programação de alto nível, interpretada, orientada a objetos, funcional e que por sua tipagem dinâmica, pode ser aplicada em diversas áreas tais como, desenvolvimento de sistemas Web e de programação científica. Ela prioriza mais a legibilidade do código do que a velocidade ou expressividade.

A mesma se caracteriza como multiparadigmas, isto é, pode ser implementada com os conceitos da orientação a objetos, a utilização de programação imperativa, funcional ou procedural.

Python é uma linguagem multiplataformas, sendo suportada pela maioria dos sistemas operacionais, inclusive em sistemas embarcados. Por possuir uma abordagem de programação de alto nível oferece uma vasta quantidade de bibliotecas e oferece interfaces de programação de aplicação de outras linguagens, tais como: o C e C++, que expandem ainda mais as suas funcionalidades.

A escolha de Python para o desenvolvimento do programa desenvolvido neste trabalho, deu-se justamente por seu suporte a biblioteca mais conhecida e utilizada para o desenvolvimento de programas para as aplicações de visão computacional chamada OpenCV (*Open Source Computer Vision Library*), a mesma foi desenvolvida na linguagem C e oferece uma vasta quantidade de funções, métodos, filtros e outra funcionalidades que facilitam o desenvolvimento de programas de visão computacional.

3.4. OpenCV

De acordo com a página oficial da biblioteca OpenCV (2018) a mesma se trata de uma biblioteca de código aberto, implementada nas linguagens de programação C e C++ para o desenvolvimento de aplicações na área de visão computacional. Esta biblioteca disponibiliza mais de três mil algoritmos para trabalhos cujo objetivo são realizar análises de visão computacional, processamento de imagem e vídeo, estrutura de dados, álgebra linear, interfaces gráficas de usuário, realidade virtual, aumentada e mista.

Possui também algoritmos de filtros de imagem, calibração de câmeras, reconhecimento de objetos e movimentos e suporta o processamento de vídeos em tempo real. Possui licença BSD (*Berkeley Software Distribution*) sendo gratuita para uso acadêmico e comercial.

Por ser escrita nas linguagens C e C++, pode ser utilizada por programas implementados em linguagem Python, por meio das interfaces de programação oferecidas pela linguagem; por este motivo foi escolhida para o desenvolvimento do programa.

3.5. Filtro de escala de cinza (*Grayscale Filter)*

Segundo Klette (2014), as imagens digitais monocromáticas em escala de cinza são imagens cuja composição é baseada na variação de cores entre a cor preto como menor intensidade e branca como maior intensidade. A Figura 6 exemplifica como as imagens de escala de cinza não possuem cores quando comparadas a imagens digitais multicamadas.

Figura 6. Fotografia de um cogumelo *Amanita muscaria* em escala de cinza.
Fonte: Pixabay, 2017

O cálculo para conversão de uma imagem multicamadas em uma imagem de escala de cinza é apresentado na Equação 2, esta conversão é necessária pois é responsável por realizar a adequação da imagem pelo fato de que a identificação de cores não é uniforme com relação ao olho humano fato que também ocorre na aquisição de imagens digitais pois foram construídas para que se aproximem ao máximo possível da visualização do olho humano. Os humanos percebem o verde mais fortemente do que o vermelho e azul, isso é decorrente do ponto de vista da biologia evolutiva pois grande parte do mundo natural aparece em tons de verde e, portanto, os humanos desenvolveram maior sensibilidade à luz verde.

Segundo Poynton (2003), a conversão acontece por meio da normalização dos valores dos canais RGB (*Red, Green e Blue*) para o modelo YIQ, dentro deste modelo *Y* representa a luminância do píxel convertido e é um valor composto por *R* que descreve o valor (*u*) da camada de cor vermelha, *G* que descreve o valor (*u*) da camada de cor verde e *B* que descreve o valor (*u*) da camada de cor azul. Esta conversão segue os padrões de recomendação da norma ITU-R BT.601 da *International Telecommunication Union*, a qual prevê um padrão para codificação de sinais de vídeo digital, cujas constantes definem valores de conversão mais adequados para correção dos valores da escala de cinza para visualização do olho humano e são utilizados na maior parte dos programas de processamento de imagens digitais como o algoritmo padrão.

Equação 2. Equação para conversão de uma imagem multicamadas RGB para o padrão YIQ

$$Y = 0{,}299 * R + 0{,}587 * G + 0{,}114 * B$$

A finalidade da utilização deste tipo de filtro de imagem, é seu refinamento, eliminando detalhes desnecessários e a preparação para a aplicação do processo de limiarização da imagem, que irá resultar em uma imagem binária que facilita a identificação dos animais presentes na imagem a ser processada.

3.6. Filtro de suavização

A suavização é uma das operações mais comuns nas aplicações de processamento de imagem. Sua finalidade é salientar determinadas características de uma imagem e também reduzir os ruídos gerados durante o processo de aquisição, seja por uma limitação de hardware que realizou a captura, seja pela excessiva compressão a fim de reduzir o tamanho em memória ocupado ou ainda por problemas nos processos de quantização e digitalização.

Segundo Jesus e Costa (2015), existem diferentes métodos de suavização de imagens tais como: a normalização, o filtro Gaussiano, o filtro mediano e bilateral.

O método Gaussiano se baseia na distribuição estatística de valores que se agrupam em torno de uma média conhecida como distribuição Gaussiana ou distribuição

normal e sendo aplicado neste tipo de filtro para identificar regiões de imagem que tenham informações semelhantes e desta maneira reduz a quantidade de informações no agrupamento da imagem causando desta forma uma redução da quantidade de informações redundantes da figura.

A Figura 7 mostra uma representação gráfica da distribuição Gaussiana, conhecida informalmente como curva de sino.

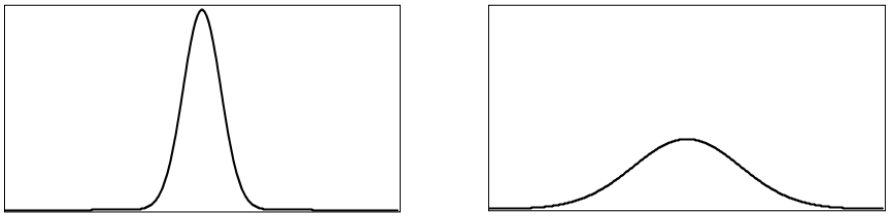

Figura 7. Características de uma curva Gaussiana. Fonte: elaboração própria.

De acordo com Shiffman (2012), esta curva é gerada por meio de uma função matemática que define a probabilidade de que um valor qualquer ocorra em função de uma média (μ) e um desvio padrão (σ).

A equação da curva gaussiana de distribuição normal padrão é representada pela Equação 3 onde z representa a distribuição e determina o valor de x.

Equação 3. Função matemática para a geração de uma curva Gaussiana.

$$z = \frac{x - \mu}{\sigma}$$

Na Figura 7 a curva da imagem do lado esquerdo possui um desvio padrão menor e por conta disso a curva possui uma menor faixa de distribuição, já na imagem do lado direito o desvio padrão é maior e, portanto, a distribuição dos valores na curva a tornam menos acentuada.

No contexto de processamento de imagens, a suavização Gaussiana é normalmente utilizada para reduzir o ruído de uma imagem e reduzir os detalhes.

O efeito visual desta técnica gera um desfoque da imagem processada e quanto menor for o valor do desvio padrão, mais agrupados em torno da média os valores serão e consequentemente mais desfocada uma imagem será. O desfoque é o termo usado para definir os pontos ao qual convergem ou divergem um feixe de ondas eletromagnéticas, sendo os feixes de luz utilizados para a composição das imagens digitais.

A Figura 8 representa a aplicação de um desfoque com base nos valores de pixel em uma fotografia de um gavião, a aplicação do filtro de suavização de 10px possui menor atenuação dos detalhes e a de valor 30px possuí maior atenuação do efeito de desfoque por possuir um maior nível da aplicação do filtro.

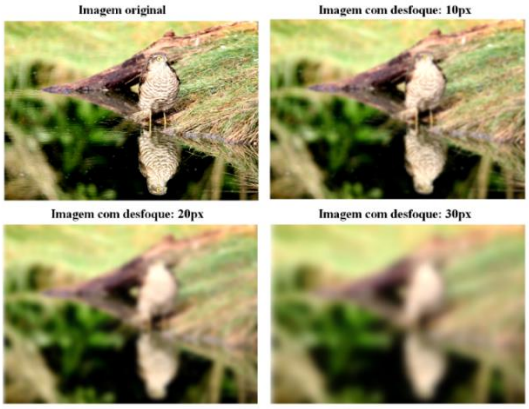

Figura 8. Fotografia de um Falcão-do-Tanoeiro sob diferentes níveis de filtro gaussiano. Fonte: Max Pixel, 2016

3.7. Limiarização de imagens

Segundo Klette (2014), o processo de limiarização de imagens digitais (do inglês, *thresholding*) é um método de segmentação cuja aplicação faz a troca de cada um dos pixels de uma imagem por uma cor, desde que, os valores de intensidade dos mesmos, sejam menores que o valor de uma constante de limite; desta maneira um pixel cuja cor seja de baixa intensidade, pode ser relacionado à cor preta e um de alta intensidade seja relacionado à cor branca.

Para que este processo seja aplicado, necessariamente a imagem de origem deve estar em escala de cinza. A Figura 9 mostra o resultado do processo de limiarização.

Figura 9. Fotografia de uma foca após o processo de limiarização com valor limite de escala 127. Fonte: Pixabay, 2017

A biblioteca do OpenCV oferece diferentes processos de limiarização de imagens. A limiarização simples é aplicada com quatro diferentes parâmetros.

O primeiro parâmetro da função é a imagem em escala de cinza que será processada, os segundo e terceiro são os valores de mínimo e máximo para classificação do pixel e o quarto parâmetro é a definição do método de limiarização cujo resultado pode ser, binário ou binário invertido.

Outro método disponível é a limiarização adaptativa, sua vantagem com relação ao processo simples é a adaptação para imagens cujas condições de iluminação são destacadas em áreas distintas da imagem. Neste método a aplicação de filtros adaptativos é mais indicada, pois o processamento da limiarização é realizado em pequenas frações e unidos posteriormente para a geração da imagem.

É possível observar na Figura 10, que a aplicação da limiarização simples em imagens em que não existe um bom contraste entre fundo e objetos para segmentação, geram resultados indesejados; já a aplicação de limiarização adaptativa tornam os resultados mais eficazes para a segmentação dos objetos da imagem.

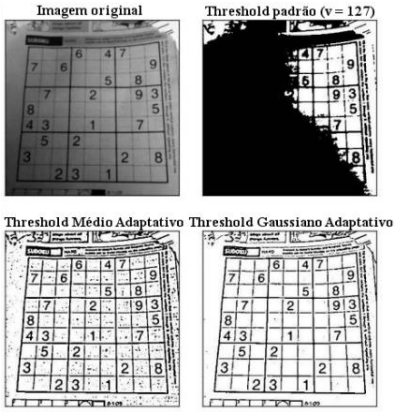

Figura 10. Imagem comparativa dos métodos de Limiarização disponíveis na biblioteca OpenCV. Fonte: Docs OpenCV, 2017

3.8. Transformações morfológicas

A fim de refinar os dados gerados pela limiarização das imagens processadas é necessária a aplicação de transformações morfológicas que são responsáveis por operar na limpeza das formas encontradas no processo de limiarização.

O processo se fundamenta na comparação entre o valor de cada pixel de uma imagem binária com relação a seus vizinhos e baseado no formato dos objetos a serem analisados, define se um pixel deve ou não existir na imagem pós-processada.

Existem seis tipos de operações morfológicas disponíveis para utilização na biblioteca OpenCV: erosão, dilatação, abertura, oclusão e os algoritmos de *Tophat* e *Blackhat*.

De acordo com a definição de Fisher et Al. (1996), a erosão enquanto operação morfológica busca remover a maior quantidade possível de ruídos existentes nas imagens binárias por meio da comparação entre os pixels vizinhos. Para definir a região dos vizinhos analisada é necessário configurar uma propriedade chamada *kernel*; esta propriedade tem o papel de definir o espaço, em pixels, que será percorrido por toda a imagem e que servirá como base para a verificação dos pixels vizinhos.

O processo se baseia na verificação do pixel origem e compara com seus vizinhos, caso possua valores iguais o pixel origem recebe o valor de seus vizinhos, caso contrário o mesmo é mantido com o mesmo valor de origem. A Figura 11 exemplifica o processo de eliminação de um pixel de ruído em uma imagem binária com utilização de *kernel* = 1.

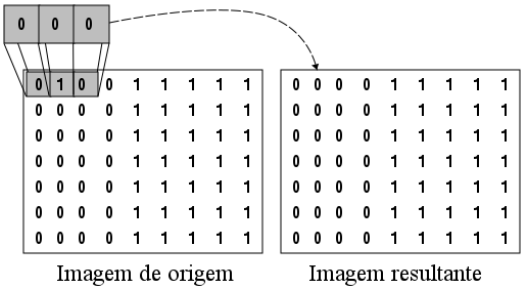

Figura 11. Processo de eliminação de pixel de ruído em uma imagem binária. Fonte: elaboração própria.

É possível observar na Figura 12 o processo de erosão aplicado na fotografia de uma mão e verificar o resultado da remoção dos ruídos existentes após o processamento da transformação morfológica.

Neste caso o objeto presente na imagem consegue uma melhor definição de forma, o que facilita em processos de detecção de objetos, já que os pixels de ruído não serão mais considerados em eventuais contagens.

Figura 12. Fotografia de uma mão após aplicação da operação de erosão.
Fonte: Fisher et Al., 1996

3.9. Detectores de objetos binários

Após o processo de aplicação dos operadores morfológicos, o resultado é uma imagem binária que possui grupos de objetos binários denominados Blobs (*Binary Large Object*) e se refere a um grupo de pixels conectados em uma imagem binária. Entretanto, somente saber que existem grupos de objetos binários em uma imagem não indica quantidades exatas, pois podem existir grupos de forma semelhante, conectados entre si e que poderiam ser considerados um grupo único, caso não sejam aplicados filtros de detecção.

Um exemplo disso, no programa desenvolvido para o presente trabalho, é a possibilidade de os animais estarem encostados lado a lado e que os filtros de processamento de imagem acabam considerando dois animais muito próximos, como um animal apenas. A Figura 13 exemplifica este tipo de problema.

Figura 13. Propriedades de algoritmos de detecção por convexidade e inércia.
Fonte: Shutterstock (Paulo Vilela), 2017

O algoritmo de detecção por cor é utilizado para seleção de pixels de uma determinada cor no padrão RGB. No exemplo mostrado, ao realizar o processo de binarização, o resultado obtido é uma imagem monocromática de duas cores (preto e branco), portanto a aplicação dos filtros deve buscar na imagem pixels de cor branca.

Já a detecção por convexidade é aplicada para eliminar os erros gerados por grupos de pixels ligados por uma pequena parte.

Especificamente, no exemplo dado o mesmo serve para eliminar os erros gerados por animais muito próximos uns dos outros. Na detecção por convexidade os parâmetros passados definem valores mínimos e máximos de convexidade de um grupo de pixels e faz a separação destes objetos pelo seu formato.

A aplicação do algoritmo de detecção por inércia, se caracteriza por agrupar os pixels que estejam entre valores mínimos e máximos de alongamento de um objeto. A Figura 14 evidencia as propriedades de um filtro de inércia e de convexidade.

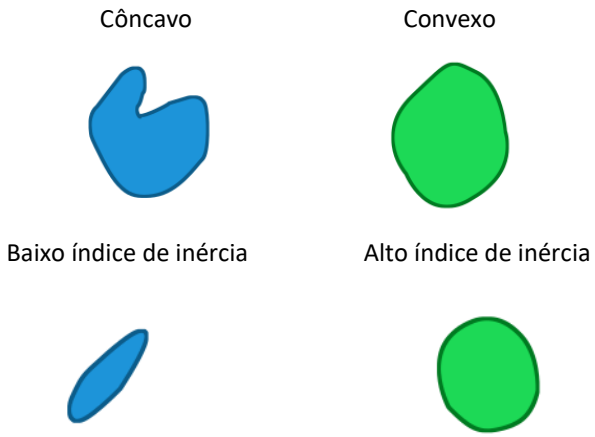

Figura 14. Propriedades de algoritmos de detecção por convexidade e inércia. Fonte: elaboração própria.

4. Metodologia

Inicialmente foi realizada a coleta de artigos e referências bibliográficas para encontrar na literatura acadêmica as possíveis soluções para a contagem de objetos por meio de imagens estáticas, considerando os trabalhos levantados e já citados anteriormente a tese de mestrado de Balan (2003) possuía objetivos em comum como a contagem de população de animais com base nas características extraídas de imagens, em particular uma das soluções apresentadas pelo mesmo em sua tese que se baseava nos métodos de detecção de bordas e serviu de referência para buscar as ferramentas que deveriam ser aplicadas para este método.

O método de detecção de bordas descrito por Balan (2003) se baseia na localização de regiões da imagem onde a variação dos tons de cinza ocorrem de maneira abrupta. Estas regiões são determinadas descontinuidades, podendo ocorrer na forma de pontos isolados, linhas, segmentos ou curvas, constituindo assim por meio delas os contornos, formas ou bordas dos objetos contidos na imagem.

Levando em consideração que as imagens utilizadas apresentam características que sugerem fortemente a utilização de métodos de segmentação tais como o contraste elevando entre os objetos de análise (animais) e o fundo das imagens (pasto ou lama), o método de detecção de bordas aliado aos métodos de limiarização se mostraram uma solução adequada a aplicação.

O próximo passo foi a obtenção das imagens aéreas utilizadas para a análise estatística dos resultados obtidos com a aplicação do programa desenvolvido, as mesmas foram adquiridas por meio de pesquisas em bancos públicos de imagens com foco nas

imagens aéreas de alta resolução e com seus direitos de uso liberados para reutilização não comercial.

Em sequência foi realizada a implementação de um programa na linguagem de programação Python cujo objetivo era identificar os objetos e realizar a posterior contagem. Segue descrito abaixo os passos necessários para realizar a contagem de gado implementada pelo programa desenvolvido:

1. Realizar a aplicação do filtro de escala de cinza à imagem a ser analisada;
2. Efetuar a aplicação de filtros de suavização para remover ruídos e imperfeições na imagem;
3. Aplicar a limiarização para definir os contornos dos objetos a serem detectados;
4. Realizar transformações morfológicas para refinar os resultados e evitar problemas comuns ao processamento de imagens;
5. Utilizar detectores de objetos binários para efetivamente realizar a contagem dos objetos de estudo na imagem.

5. Resultados obtidos

A execução do programa, elaborado neste trabalho foi realizada em quatro imagens. O resultado é apresentado com marcações vermelhas que apontam os objetos captados. Em um terminal de console é mostrada uma mensagem com a quantidade de objetos detectados na figura processada.

Para fazer os testes do programa foram coletadas imagens aéreas de gado em confinamento e em campo aberto. As imagens coletadas foram selecionadas para que oferecessem uma delimitação bem precisa de seus elementos constitutivos (animais, terreno, características da imagem) a fim de reforçar os elementos básicos para o funcionamento adequado do programa desenvolvido. Todas as imagens coletadas estão disponíveis para acesso público na plataforma do Google Drive[2] mantida pelo autor.

Com as imagens coletadas foram realizados testes, aplicando-se diferentes parâmetros de limite de limiarização, diferentes níveis de regiões de kernel a serem percorridas e valores distintos de convexidade e concavidade para os objetos identificados em cada imagem, todos parametrizados por interface de linha de comando passados como argumento para a execução do programa.

A Tabela 1 contém uma compilação dos resultados e apresenta a quantidade de animais detectados em cada uma das fotografias testadas (na ordem do artigo), a quantidade de animais de cores claras, a quantidade original e a taxa de acerto em porcentagem com base na quantidade de detecções e a quantidade original de animais.

Balan (2003) propõe como critério de qualidade dos resultados os aspectos de qualidade dos resultados obtidos e critérios de convergência. O primeiro diz respeito à capacidade do método empregado identificar todas, ou a maioria das regiões ou objetos relevantes na cena. O segundo consiste na análise dos desvios de falsos positivos e negativos e verdadeiros positivos e negativos gerados pelo diferente nível de escala de cinza específico de cada imagem.

[2] Todas as imagens coletadas estão disponíveis em pasta de arquivos do Google Drive sob a URL de acesso público: (https://drive.google.com/drive/folders/0B_VFW80S7UPQb0twZUpYTE92VEE?usp=sharing).

Tabela 1. Compilação de resultados das execuções com base nas fotografias testadas.

Fotografia	Quantidade de detecções	Quantidade de animais claros	Quantidade original de animais	Taxa de acerto em relação a animais brancos	Taxa de acerto
1. Gado em campo aberto, distantes uns dos outros e com bom contraste com o fundo. (Figura 15)	19	19	19	100,0%	100,0%
2. Gado em campo aberto, distantes uns dos outros e com animais de cores distintas. (Figura 16)	7	7	8	100%	87,5%
3. Gado em campo aberto e com animais próximos uns dos outros e com bom contraste de fundo. (Figura 17)	22	23	23	95,7%	95,7%
4. Gado em ambiente de confinamento e com grande quantidade de animais, animais de cores distintas e de cor semelhante ao terreno. (Figura 18)	49	53	141	92,4%	34,8%
5. Gado em campo aberto, com animais próximos uns aos outros e com bom contraste de fundo. (Figura 19)	41	48	48	85,4%	85,4%
6. Gado em campo aberto, animais separados, com bom contraste de fundo e com animais de cores escuras.	3	3	5	100%	60%
7. Gado em campo aberto, com contraste ruim entre animais e fundo e com animais de cores escuras.	10	12	17	83.3%	58.9%
8. Gado em campo aberto, com bom contraste de imagem e com diversos animais próximos uns aos outros.	14	20	21	70%	66,6%

Foi possível concluir com os resultados apurados, que não existe uma parametrização global que consiga detectar plenamente todos os animais por meio do processamento de imagens, ainda mais quando os mesmos não são separados por cor.

Para os animais brancos, o algoritmo apresenta uma menor taxa de erro. Para os de cor escura, dada a necessidade de aplicar filtros distintos para identificação de cada

coloração, há uma dificuldade maior, pois quando existem animais marrons ou malhados no mesmo lote, o processo de limiarização não consegue distinguir os pixels de animais, cujas cores são escuras, o que faz com que sejam considerados como parte do fundo da imagem. Consequentemente não são adicionados à contagem final dos animais.

A Figura 15 apresenta o processamento de uma imagem contendo animais em um pasto com uma boa distância entre eles e um contraste relativamente grande entre objeto e fundo. Nesta imagem o algoritmo conseguiu detectar todos os animais presentes sem apresentar erros.

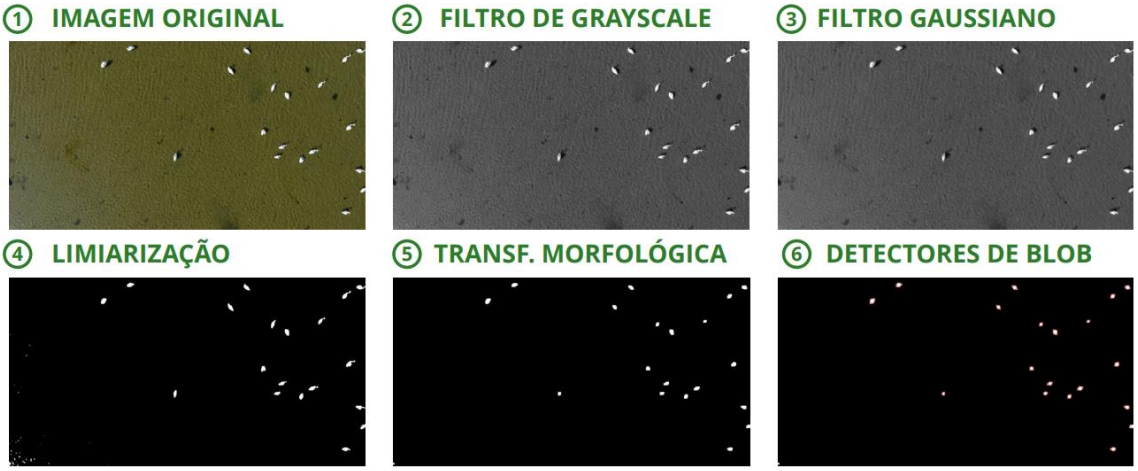

Figura 15. Resultado do algoritmo em uma imagem com animais brancos.
Fonte: Shutterstock (Mukola), 2017

A Figura 16 exemplifica o resultado do algoritmo, em casos onde existe a presença de animais de cores diversas. Nesta figura existe um animal de cor preta, o qual após a aplicação do filtro de limiarização é considerado como parte do fundo da imagem e consequentemente é descartado na contagem final.

Nesta execução não foi necessária a aplicação de parâmetros diferentes do padrão e o resultado apesar da eliminação do animal de cor negra foi considerado bom tendo em consideração o verdadeiro negativo de somente um animal, representando uma identificação de 87,5% dos animais presentes na imagem.

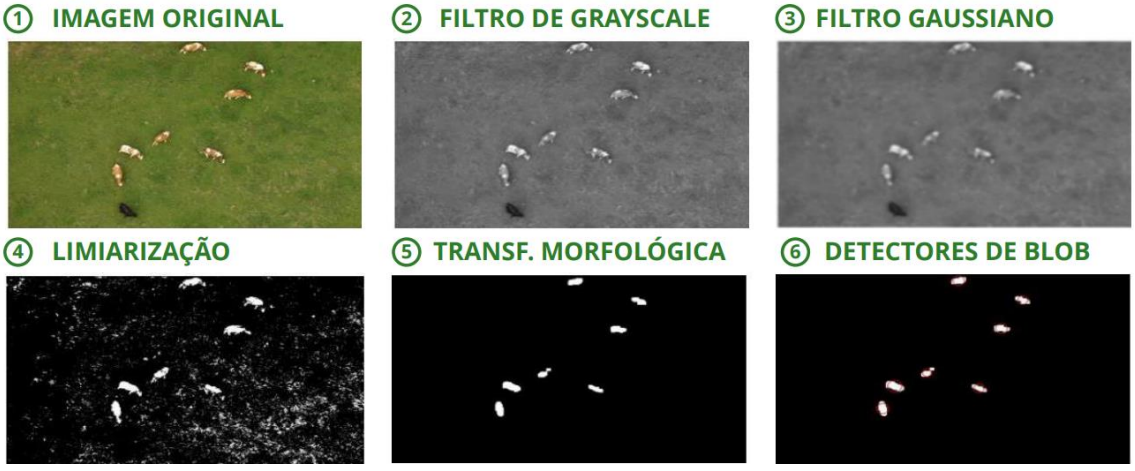

Figura 16. Resultado do algoritmo em uma imagem com animais de múltiplas cores.
Fonte: Pixabay, 2017

A Figura 17 mostra o resultado do algoritmo em casos em que, no momento de retirada da fotografia, os animais se encontravam muito próximos. Nestes casos o processamento do algoritmo necessita de algumas modificações nos filtros de transformação morfológica, mais especificamente na diminuição do kernel de análise para que os animais que se posicionam próximos uns aos outros sejam identificados na figura processada.

O resultado do algoritmo apresentou uma boa quantidade de identificações perdendo a identificação de somente um animal que estava próximo de outro e que no processo de transformação morfológica acabou se tornando um único objeto.

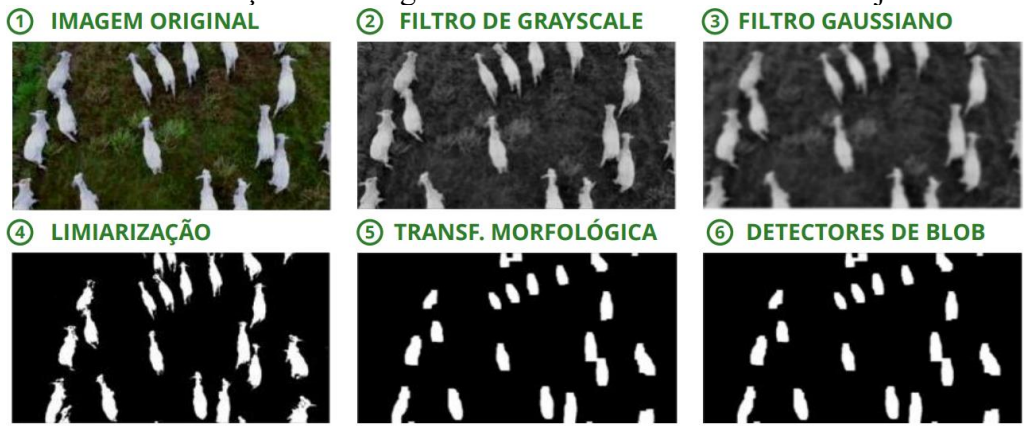

Figura 17. Resultado do algoritmo em imagens com animais muito próximos.
Fonte: Shutterstock (Paulo Vilela), 2017

A Figura 18 apresenta os resultados do algoritmo após a execução em uma fotografia que analisa a existência de 141 animais de diferentes cores em confinamento.

Nela existe uma grande variação de coloração dos animais (pretos, brancos, marrons e malhados), assim foi necessária a mudança dos parâmetros padrões do programa sendo modificados os limites de limiarização superior e inferior, o tamanho do kernel de análise e nos valores de convexidade e concavidade.

O resultado não foi satisfatório, dado que na fotografia original existiam 141 animais em confinamento, dos quais 53 possuíam colorações mais claras. Com a execução do programa, se analisássemos apenas os animais de cor clara, foi possível detectar uma quantidade de 49 animais dentre os 53, o que equivale a 92,4%, porém se considerados o total de animais, o resultado fica bastante prejudicado sendo rebaixado a 34,8% de animais detectados.

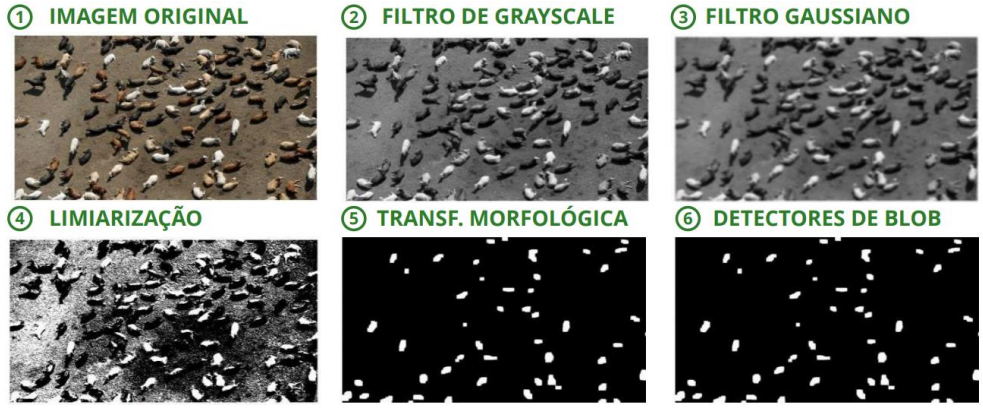

Figura 18. Resultado da execução do programa em uma imagem com animais de múltiplas cores, próximos uns dos outros e em terreno de cor similar a cor de alguns animais.
Fonte: Lucy Nicholson/Reuters, 2015

A Figura 19 apresenta os resultados do estudo em uma fotografia de animais em campo aberto, estando muito próximos uns dos outros e sendo em maioria de coloração branca.

O resultado da execução foi satisfatório tendo uma taxa de acerto de 85,4%. O problema encontrado neste exemplo é a baixa resolução da imagem que afetou nos resultados finais.

A existência de bezerros que possuem tamanho inferior ao de um animal adulto também causou distorções; resultou que fossem considerados como parte do fundo por não possuírem um tamanho relevante para serem considerados após a aplicação dos filtros morfológicos padrões. Alguns deles, além de serem bezerros, possuíam coloração semelhante à do terreno e, portanto, foram desconsiderados na contagem final.

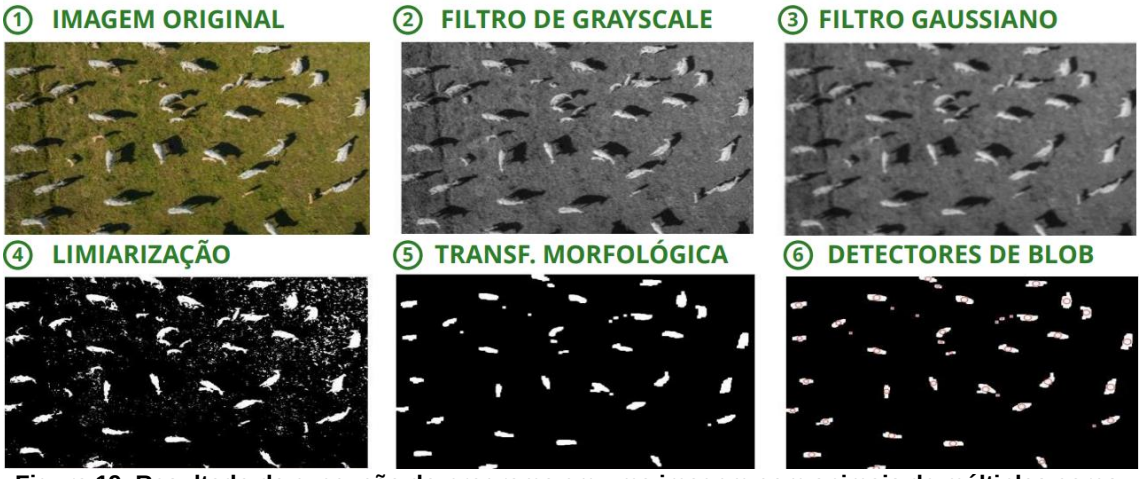

Figura 19. Resultado da execução do programa em uma imagem com animais de múltiplas cores, próximos uns dos outros e em terreno de cor similar a cor de alguns animais.
Fonte: Atlantide Phototravel/Corbis 2014

O resultado da taxa de acerto considerada na tabela é com base na quantidade de detecções pela quantidade de animais originais na fotografia, este valor é convertido diretamente a uma porcentagem que resulta na última coluna da tabela. A quantidade de animais claros serve apenas de referência para um segundo índice que possa ser considerado no aproveitamento do programa com relação a detecção exclusivamente de animais de cores claras.

É possível observar pelos resultados compilados na tabela que o algoritmo possui uma boa taxa de acerto na maior parte dos casos, sendo afetado somente quando existem animais de cores distintas da branca na fotografia.

Como os filtros escolhidos utilizam por base a cor branca, os animais de cores escuras tais como os malhados, marrons e pretos acabam sendo desconsiderados na contagem.

O resultado de significância menor, 34,8% de taxa de acerto, ocorreu numa fotografia em que os animais se encontravam em um ambiente de confinamento e por conta de o terreno possuir cor similar à de alguns dos animais, por isso a taxa de acerto foi baixa. Entretanto, quando o resultado é feito em comparação a quantidade de animais de cores claras na mesma fotografia, a taxa de acerto sobe para 92,4%.

6. Conclusões e trabalhos futuros

Com os resultados obtidos pelo programa desenvolvido, foi observado que é possível utilizar algoritmos de visão computacional para detectar dentro de imagens estáticas a quantidade de objetos presentes. Entretanto, existem restrições com relação às cores dos animais que podem ser eliminadas da contagem quando os filtros aplicados são direcionados a uma determinada característica.

O programa desenvolvido é efetivo na contagem de animais de cores brancas atingindo uma considerável taxa de sucesso na identificação do gado presente nas fotos analisadas.

Foi observado também que as limitações do programa são decorrentes de duas características inerentes às imagens coletadas:

1. Primeiramente, quando existem animais de cores escuras os filtros aplicados pelo algoritmo acabam por considerar estes como parte do fundo da imagem e logo após a segmentação são desconsiderados da contagem final;
2. Segundo problema recorrente é quando existem animais muito próximos uns dos outros e de diferentes tamanhos, desta maneira a segmentação não consegue distinguir ambos e os considera um animal somente.

E por fim, o terceiro fator negativo encontrado, foi a semelhança de características dos animais de cor marrom com a cor do chão (composto principalmente de terra) em ambientes de confinamento.

Uma maneira de mitigar os erros apresentados no algoritmo seria a utilização fotografias registradas com câmeras térmicas. Neste tipo de imagem, os animais que apresentam temperatura superior à da superfície terrestre tomam forma melhor em suas silhuetas, assim com a aplicação de filtros de segmentação adequados, o resultado do algoritmo poderia ser melhorado consideravelmente.

A Figura 20 mostra animais num campo, por meio de uma fotografia térmica.

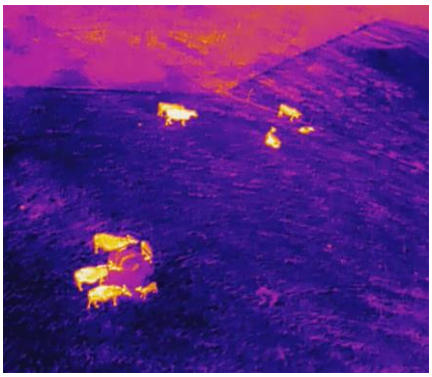

Figura 20. Fotografia térmica de animais de um gado em pasto aberto.
Fonte: Jeremiah Karpowicz, 2017

Outra forma de refinar ainda mais os resultados, seria a utilização de vídeos para a análise, neste caso é possível fazer comparações entre os quadros de imagem do vídeo, resultando em análises de maior fidelidade ao propósito de contagem dos animais de uma fazenda.

Uma futura melhoria a ser trabalhada para o aperfeiçoamento e automatização de parametrizações, seria a aplicação de algoritmos de inteligência artificial e aprendizagem de máquina, estes seriam aplicados para identificar os melhores parâmetros a serem considerados para cada tipo de imagem, dadas as características dos animais, nesta hipótese, os animais de cores escuras (marrom, preto e malhado) poderiam ser mais facilmente detectados e isso, consequentemente, poderia ampliaria sobremaneira a quantidade de animais de variadas características durante o processo de contagem, ampliando assim ainda mais os resultados apurados, justificando-se assim, a aplicação do algoritmo de visão computacional na contagem de animais por meio de processamento de imagens, conforme detalhados no presente artigo.

7. Referências

IBGE (2016). Indicadores IBGE - Contas nacionais trimestrais: Indicadores de volume e valores correntes. p. 20–21.

FAOSTAT – Food and Agriculture Organization of United States: Banco de dados. Disponível em meio eletrônico. Acesso em novembro de 2017.

Amaral, G. F., Carvalho, F. A. A., Capanema, L. X. L., e Carvalho, C. A. D. (2012). Panorama da pecuária sustentável. BNDES. Setorial n. 36, Set. 2012, p. 249-288.

Guedes B. S. A. (2017). Reconhecimento de Gestos usando Redes Neurais Convolucionadas. Universidade de Brasilia.

Chamoso P., Raveane W., Parra V., González A. (2014). UAVs Applied to the Counting and Monitoring of Animals. Advances in Intelligent Systems and Computing. 291. 10.1007/978-3-319-07596-9_8.

Jesus E., Costa R. (2015). A utilização de Filtros Gaussianos na Análise de Imagens Digitais. Brazilian Society of Applied and Computational Mathematics. Vol. 3, N. 1, 2015, Natal-RN.

J. Howse, (2014). Training detectors and recognizers in Python and OpenCV. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Munich, 2014, pp. 1-2. 10.1109/ISMAR.2014.6948516

Bhumika G., Ashish C. & N., and Umang G. (2017). Study on Object Detection using Open CV - Python. International Journal of Computer Applications 162(8):17-21.

Balan, A. G. R. (2003). Técnicas de segmentaçao de imagens aéreas para contagem de populaçao de aves (Doctoral dissertation, Universidade de São Paulo).

Klette, R. (2014). Concise computer vision – An Introduction into Theory and Algorithms – UTICS (Undergraduate Topics in Computer Science). Editora Springer.

Leighton, R.B., N.H. Horowitz, A.G. Herriman, A.T. Young, B.A. Smith, M.E. Davies, and C.B. Leovy (1969). Mariner 6 television pictures: First report. Science, n.165, p. 684–690.

Poynton, C. (2003). Digital Video and HDTV Algorithms and Interfaces. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA. ISBN: 1558607927.

Python Software Foundation (2015). Python Brochure. Vol. 1, 2nd Edition, p. 3

Gonzalez, R. C., & Woods, R. E. 1. (2008). Digital image processing (3rd ed.). Upper Saddle River, N.J.: Prentice Hall.

Shiffman,D. (2012). The Nature of Code. ITP – Interactive Media Arts (IMA). p. 11-14.

Fisher, R., Perkins, S., Walker, A., & Wolfart, E. (1996). Hypermedia image processing reference. England: John Wiley & Sons Ltd. 118-130.