

DIÓGENES LODI PINTO

Uso de algoritmos computacionais para análise de imagens e aplicações na produção animal: técnicas de *Machine Learning* para classificação dos escores de marmoreio da área de olho de lombo

Pirassununga

2021

DIÓGENES LODI PINTO

Uso de algoritmos computacionais para análise de imagens e aplicações na produção animal: técnicas de *Machine Learning* para classificação dos escores de marmoreio da área de olho de lombo

Dissertação apresentada ao Programa de Pós-Graduação em Nutrição e Produção Animal da Faculdade de Medicina Veterinária e Zootecnia da Universidade de São Paulo para a obtenção do título de Mestre em Ciências.

Departamento:

Nutrição e Produção Animal

Área de concentração:

Nutrição e Produção Animal

Orientador:

Prof. Dr. Ricardo Vieira Ventura

Pirassununga

2021

Autorizo a reprodução parcial ou total desta obra, para fins acadêmicos, desde que citada a fonte.

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO

(Biblioteca Virgínia Buff D'Ápice da Faculdade de Medicina Veterinária e Zootecnia da Universidade de São Paulo)

T. 4146
FMVZ

Pinto, Diógenes Lodi
Uso de algoritmos computacionais para análise de imagens e aplicações na produção animal: técnicas de *Machine Learning* para classificação dos escores de marmoreio da área de olho de lombo / Diógenes Lodi Pinto. – 2021.
69 f. : il.

Dissertação (Mestrado) – Universidade de São Paulo. Faculdade de Medicina Veterinária e Zootecnia. Departamento de Nutrição e Produção Animal, Pirassununga, 2021.

Programa de Pós-Graduação: Nutrição e Produção Animal.

Área de concentração: Nutrição e Produção Animal.

Orientador: Prof. Dr. Ricardo Vieira Ventura.

1. Aprendizado de máquina. 2. Escore de marmoreio. 3. AOL. 4. Python. 5. LBP. I. Título.

Certificado da Comissão de Ética



Comissão de Ética no Uso de Animais

Faculdade de Medicina Veterinária e Zootecnia
Universidade de São Paulo

São Paulo, 23 de setembro de 2021

CEUAX N 7596020320

(ID 001334)

Ilmo(a). Sr(a).

Responsável: Ricardo Vieira Ventura

Área: Nutrição E Produção Animal

Título da proposta: "Uso de algoritmos computacionais para análise de imagens e suas aplicações na produção animal".

CERTIFICADO (Relatório Parcial versão de 19/maio/2021)

A Comissão de Ética no Uso de Animais da Faculdade de Medicina Veterinária e Zootecnia da Universidade de São Paulo, no cumprimento das suas atribuições, analisou e **APROVOU** o Relatório Parcial (versão de 19/maio/2021) da proposta acima referenciada.

Resumo apresentado pelo pesquisador: "Estudante: Diógenes Lodi Pinto CEUAX Nº 7596020320 Qual o estágio do estudo no momento? O projeto intitulado "[Uso de algoritmos computacionais para análise de imagens e suas aplicações na produção animal]", CEUAX Nº 7596020320, mestrando Diógenes Lodi Pinto, aprovado no exame de qualificação em 27/11/2020, estuda de forma objetiva técnicas de Análises de Imagem e Visão Computacional com o intuito de solucionar problemas ligados à área de Produção Animal. Para tal finalidade, utiliza um banco de imagens e mensurações (N = 7.704) relacionados à superfície do músculo Longissimus Dorsi, extraídas a partir da 12ª e 13ª vértebras de animais taurinos cruzados, cedidos pela empresa canadense AgSights. Até o presente momento, um subgrupo de imagens (N=1.050) tiveram seus backgrounds tratados, regiões de interesse rotuladas manualmente (3 labels por imagem), realizadas técnicas de visão computacional para extração de atributos e planilha dessas principais características atreladas a área de olho de lombo. O passo final será utilizar técnicas de machine learning/deep learning para classificação e verificar possibilidade de segmentação automatizada de todo o processo. Conforme mencionado anteriormente, não existe manipulação ou uso direto de animais no presente projeto. Por quanto tempo mais o estudo se estenderá? De acordo com os avanços obtidos até o momento, estima-se que serão necessários aproximadamente 160 dias para a finalização do trabalho, sua escrita, defesa pública e submissão das conclusões para publicação. Resultados parciais ou totais apresentados em congresso? Em 2020, foi apresentado entre os dias 07 e 08 de dezembro, durante o I Simpósio Internacional da Pós-graduação (XIV Simpósio de Pós-Graduação e Pesquisa em Nutrição Animal-VNP-2020), o projeto intitulado "[Uso de segmentação instanciada para a detecção da área de olho de lombo no músculo longissimus dorsi em imagens sem tratamento]" conforme anexo. Resultados parciais ou totais já publicados? O projeto apresentando a essa CEUAX, ainda não teve dados parciais ou totais publicados. "

Comentário da CEUA: Projeto em andamento.

Prof. Dr. Marcelo Bahia Labruna
Coordenador da Comissão de Ética no Uso de Animais
Faculdade de Medicina Veterinária e Zootecnia da Universidade
de São Paulo

Camilla Mota Mendes
Vice-Coordenadora da Comissão de Ética no Uso de Animais
Faculdade de Medicina Veterinária e Zootecnia da Universidade
de São Paulo

FOLHA DE AVALIAÇÃO

Autor: PINTO, Diógenes Lodi

Título: Uso de algoritmos computacionais para análise de imagens e aplicações na produção animal: técnicas de *Machine Learning* para classificação dos escores de marmoreio da área de olho de lombo

Dissertação apresentada ao Programa de Pós-Graduação em Nutrição e Produção Animal da Faculdade de Medicina Veterinária e Zootecnia da Universidade de São Paulo para a obtenção do título de Mestre em Ciências.

Data: 17 / 12 / 2021

Banca Examinadora

Prof. Dr. Ricardo Vieira Ventura
Instituição: VNP – FMVZ – USP

Julgamento: _____

Profa. Dra. Mirele Daiana Poleti
Instituição: FZEA – FMVZ – USP

Julgamento: _____

Prof. Dr. Roberto Carneiro
Instituição: FCAV – FMVZ – UNESP

Julgamento: _____

DEDICATÓRIA

Dedico este trabalho à minha esposa, Maria Emília, exemplo de dedicação e comprometimento e a meus pais, Luiz e Neusa, que sempre me apoiaram e incentivaram meus estudos.

AGRADECIMENTOS

À AgSights, Jasper Munro e Gordon Voort pelo banco de dados concedidos para a pesquisa.

Ao VNP, programa de Nutrição e Produção Animal da FMVZ da USP.

Aos amigos de laboratório (BioMA): Lucas Andrietta, Wecksley Souza, Alana Selli, Lígia Mesquita e Fernando Bussiman.

As amigas de trabalho (Sesi): Natália Rodrigues e Mariane Gallo.

Ao meu primo: Mateus Habermann

Ao meu sogro e cunhado: José Zanca e Mario Zanca

Aos Doutores: Dan Tulpan, Júlio Balieiro, Pollyana Garbossa, Guilherme Pugliesi, Mirele Poleti, Roberto Carvalheiro e Anderson Alves.

Em especial, ao meu orientador, Ricardo Ventura, por todo ensinamento e amizade construída ao longo dos últimos três anos.

RESUMO

PINTO, D. L. **Uso de algoritmos computacionais para análise de imagens e aplicações na produção animal:** técnicas de *Machine Learning* para classificação dos escores de marmoreio da área de olho de lombo. 69 f. Dissertação (Mestrado em Ciências) – Faculdade de Medicina Veterinária e Zootecnia, Universidade São Paulo, Pirassununga, 2021.

O presente estudo objetiva utilizar técnicas de Análises de Imagem e Visão Computacional (VC), com o intuito de solucionar problemas ligados à área de Produção Animal. Mais especificamente, investigou-se a possibilidade de automatização do escore de marmoreio via métodos de visão computacional aplicados em imagens .JPG. Tais imagens capturaram a região da área de olho de lombo (AOL), oriundas de animais taurinos canadenses. Estas análises foram conduzidas com o objetivo de auxiliar pequenas plantas frigoríficas durante o processo de classificação do escore de marmoreio. Dessa forma, buscou-se neste estudo a comparação de diferentes algoritmos de aprendizado de máquina tendo como entrada de dados diferentes tipos de atributos extraídos de imagens da AOL. Para tal finalidade, obtivemos acesso a um banco de imagens e mensurações (N = 7.704) relacionadas à superfície do músculo *Longissimus dorsi*, extraídas a partir da 12^a e 13^a vértebras. Primeiramente, conduzimos um estudo relacionado à determinação da porcentagem de gordura na AOL, via criação de algoritmo implementado em Python (bibliotecas Numpy e OpenCV). Os resultados mostraram que o excesso de brilho em algumas imagens levou o algoritmo a inflacionar em até 54% a GIM quando carcaças do padrão 'A' foram consideradas, e em até 800% para as classificações "Prime". Posteriormente, apresentamos uma análise piloto como tentativa de segmentação semântica, técnica que nos permite detectar objetos contidos numa mesma imagem por meio de máscaras de cores distintas. Resultados preliminares sugerem que a segmentação semântica possa ser utilizada como potencial ferramenta durante o processo de classificação automatizada das AOL, desde que o tratamento prévio das imagens seja efetivamente aplicado, principalmente em relação ao background, rotação e luminosidade. Dois métodos de Machine Learning foram investigados com o objetivo de classificar os escores de marmoreio automaticamente (*Decision Tree* - DT e *Random Forest* - RF). Com a intenção de extrair atributos baseados nos padrões e texturas das imagens, uma proposta de modificação do método LBP (*Local Binary*

Pattern) foi apresentada. Como resultado principal deste estudo, o *subset* formado pelas imagens das classes A e AAA (N = 400, extremos para o escore de marmoreio), apresentou os melhores resultados de acurácia (> 90%), para ambos os algoritmos. Aumentando-se o número de imagens para 582 amostras de cada classe, o algoritmo RF apresentou os melhores resultados (> 85%), quando comparado ao método DT (77%). Tal declínio pode estar associado com a utilização de imagens previamente classificadas erroneamente, via processo de avaliação subjetiva. Novo processo de avaliação, conduzido por seis avaliadores externos, encontra-se em fase de conclusão para repetição dos experimentos. Em conclusão, nosso trabalho obteve resultados classificatórios satisfatórios, além de apresentar uma nova metodologia para extração de atributos (LBP), anteriormente aplicado somente em áreas externas à produção animal. Métodos supervisionados requerem que a definição de *labels* seja conduzida com eficácia para a obtenção correta de padrões em processos classificatórios.

Palavras-chave: Aprendizado de máquina. Escore de marmoreio. AOL. Python. LBP.

ABSTRACT

PINTO, D. L. **Computer algorithms applied to image analysis and application on animal production:** Machine Learning approaches for the marbling score classification in beef cattle. 69 f. Dissertação (Mestrado em Ciências) – Faculdade de Medicina Veterinária e Zootecnia, Universidade São Paulo, Pirassununga, 2021.

The current study aims to use Image Analysis and Computer Vision approaches in order to solve problems related to Animal Production. We investigated the possibility of creating an automated pipeline to classify the marbling score via computer vision methods applied to .JPG files. Such images captured the ribeye area region collected from canadian beef cattle. These analyses were carried out with the purpose of helping small slaughterhouses on the process of marbling score classification. Two Machine Learning algorithms were compared having as input different attributes extracted from the Ribeye area images. Our database comprised images and measurements (N = 7.704) regarding the transversal cut between the 12# and 13# ribs of the Longissimus dorsi muscle. The first step of our study focused in the estimation of the fat percentage from the image samples via an algorithm implemented in Python using Numpy and OpenCV packages. The results showed that the brightness in some images inflated the marbling score by up to 54% in standard 'A' carcasses, and, an inflation up to 800% was observed in the "Prime" cuts. In the second step we evaluated the semantic segmentation as a pilot process. It allows the detection of different objects in an image based on previously labeled color masks. Preliminary results suggested that the semantic segmentation can be used as a potencial tool during the automated classification process of the ribeye area, as long as the pre-treatment of the images is effectively applied, mainly in relation to background, rotation, luminosity. Two methods of Machine Learning were investigated in order to classify image scores automatically (Decision Trees - DT and Random Forests - RF). In order to extract features based on patterns and textures, we proposed an extension of the LBP (Local Binary Pattern) method. The subset formed by the images of classes A and AAA scores (N = 400, extreme of marbling score) showed the best accuracy results (>90%) in both algorithms. After increasing the number of images to 582 samples each, the RF algorithm showed the best results (> 85%), when compared to the DT method (77%). The classification accuracy decrease may be associated with the use of erroneously

classified images due to the subjective evaluation process. A new classification was started by six external appraisers and the whole set of analysis will be repeated once it is completed. In conclusion, the results of the current study showed satisfactory classification results, as well as presented a new method for feature extraction (LBP), since it was applied so far on other unrelated areas. In addition, an accurate labeling process is needed to achieve better accuracy when applying supervised classification methods.

Keywords: Machine Learning. Marbling score. Ribeye area. Python. LBP

LISTA DE FIGURAS

Figura 1 - Cores primárias aditivas e subtrativas	16
Figura 2 - Cubo RGB e pirâmide hexagonal HSV	17
Figura 3 - Passos fundamentais do PDI.....	18
Figura 4 - Imagem semanticamente segmentada com áreas rotuladas.....	23
Figura 5 - Mask R-CNN results on the COCO test set	24
Figura 6 - Padrões de marmoreio canadense agrupado por categorias (A, AA, AAA e Prime).....	25
Figura 7 – “e + v Technology”	25
Figura 8 - USDA ribeye grids.....	26
Figura 9 - Two-level version (LBP) of the texture unit	29
Figura 10 - Padrões binários invariantes de rotação para uma vizinhança circular simétrica $LBP8, Rri$	29
Figura 11 - Vizinhança circularmente simétrica $LBP8, Rri$	30
Figura 12 - Diagrama descrevendo o Decision Tree	32
Figura 13 - Árvore de decisão aplicada a determinação das formas, cores e tamanho de um objeto baseadas na figura de Leslie Kaelbling.	33
Figura 14 - Árvore de Decisão da íris.....	34
Figura 15 - Floresta Aleatória combinando todas as saídas de todas as Árvores de Decisão	37
Figura 16 - Exemplo demonstrando a validação cruzada tripla.....	38
Figura 17- Determinação manual da AOL usando GIMP	42
Figura 18 - Steaks A / AAA e respectivas máscaras OpenCV	42
Figura 19 - Cartões criados para testar a acurácia do algoritmo.....	44
Figura 20 - Predição da AOL após 30 épocas (1 label).....	47
Figura 21 - Inserção de quadrados 9 cm ² através do Painted 3D.	48
Figura 22 - Determinação da área real da AOL para 100 amostra.....	49
Figura 23 - Máscara de cores sólidas criada a partir do script (Kentaro)	51
Figura 24 - Determinação das regiões de interesse (ROI)	52
Figura 25 - Divisão da AOL em quadrados 40 x 40 pixels	53
Figura 26 - Padrões de GIM para 30 amostras de janelas 40 x 40 pixels	53
Figura 27 - Cartões 10x10 pixels em diferentes padrões e tons de vermelho.....	54

Figura 28 - Determinando padrões uniformes possíveis para o LBP(8,1).....	56
Figura 29 - Diferentes padrões de steaks contidos no banco de dados.....	60
Figura 30 - Histograma de frequência do percentual de GIM presente em 1746 amostras por classes	63

LISTA DE TABELAS

Tabela 1 - 10 amostras do conjunto iris	35
Tabela 2 - Exemplos aplicados na função custo CART	35
Tabela 3 - Razão de pixels brancos e não brancos contidos na AOL (5 A e 5 Prime).....	44
Tabela 4 – Cálculo da entropia em AOL 10x10 pixels.....	55
Tabela 5 – Histograma de frequência para AOL 10x10 pixels e LBP(8,1)	56
Tabela 6 - Diferentes métricas LBP e matrizes quadradas de ordens distintas (200 A e 200 AAA)	58
Tabela 7 - Aplicação de diferentes métricas LBP e matrizes quadradas (200A e 200AAA).....	59
Tabela 8 - Aplicação do LBP(8,1) em matrizes quadradas (200/250A , 200/250AA e 200/250AAA)	60
Tabela 9 - Aplicação de diferentes métricas LBP e matrizes quadradas (582A, 582AA e 582 AAA)	61
Tabela 10 - Aplicação de diferentes métricas LBP e matrizes quadradas (582A e 582 AAA).....	62
Tabela 11 - Estatísticas descritivas para 1746 amostras (582A, 582AA e 582 AAA).....	63

SUMÁRIO

1	INTRODUÇÃO	15
1.1	VISÃO COMPUTACIONAL	15
1.2	APRENDIZADO DE MÁQUINA (MACHINE LEARNING)	18
1.3	APRENDIZADO PROFUNDO (DEEP LEARNING)	20
1.4	SEGMENTAÇÃO SEMÂNTICA E INSTANCIADA.....	22
1.5	PROBLEMAS RELACIONADOS À CARNE BOVINA.....	24
1.6	ALGORITMOS DE CLASSIFICAÇÃO	27
1.6.1	Padrões Locais Binários (Local Binary Pattern)	28
1.6.2	Árvores de Decisão (Decision Tree)	30
1.6.3	Florestas Aleatórias (Random Forest)	36
2	DESENVOLVIMENTO.....	39
2.1	PROBLEMÁTICA.....	39
2.2	MATERIAIS E MÉTODOS	40
2.2.1	Determinação de pixel brancos e vermelhos na AOL.....	41
2.2.2	Análise Piloto: nível de GIM capturado nos extremos de classificação para marmoreio (A and Prime)	43
2.2.3	Segmentação semântica manual.....	45
2.2.4	Padronização do background e inserção do objeto de referência	50
2.2.5	Extração de atributos via LBP	51
2.2.6	Algoritmos de Machine Learning (Decision Tree e Random Forest)	57
3	CONCLUSÃO.....	64
	REFERÊNCIAS.....	65

1 INTRODUÇÃO

1.1 VISÃO COMPUTACIONAL

Como o próprio nome sugere, Visão Computacional (VC) é uma área da Ciência da Computação, voltada ao estudo de métodos e algoritmos que fazem uso de câmeras e computadores com foco na extração de atributos visuais de diferentes objetos (SZELISKI, 2010). Para Ballard e Brown (1982), a Visão Computacional constrói as descrições explícitas e significativas de objetos físicos a partir de imagens. Essas descrições são pré-requisitos para reconhecer, manipular e pensar sobre objetos. Um sistema de visão por computador processa imagens adquiridas por meio de uma câmera eletrônica, que realiza o mesmo papel do sistema de visão humana em que o cérebro processa imagens derivadas do olho. O objetivo de ambos os sistemas é interpretar dados espaciais, que são indicados por mais de uma dimensão (NIXON; AGUADO, 2020).

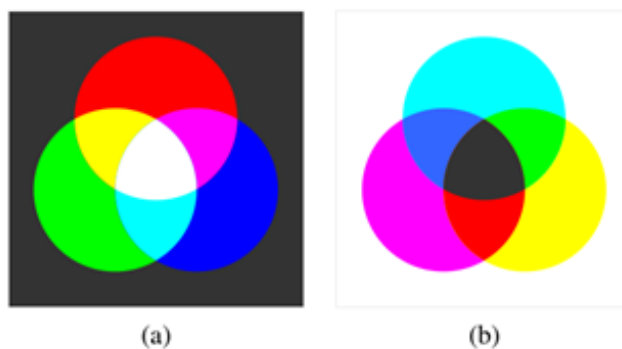
Segundo Gonzalez e Woods (2008), uma imagem pode ser definida como uma função bidimensional $f(x,y)$, onde x e y são coordenadas (plano) espaciais, e a função f representa a intensidade (ou nível de cinza) da imagem naquele ponto. Chamamos de imagem digital, cenários onde x , y e intensidade, todos provenientes de f , são finitos e representam quantidades discretas. Podemos dizer em outras palavras, que uma imagem é uma fotografia (geralmente uma projeção 2D de uma cena 3D) capturada por um sensor. Sensores de imagem digital consistem em uma matriz 2D de elementos fotossensíveis, e cada elemento (pixel / ponto de imagem / ponto de amostragem) tem uma área fixa sobre a qual é fotossensível (DANSON-HOWE, 2014).

Conforme Szeliski (2010), ao atingir o sensor de imagem, a luz de diferentes partes do espectro é de alguma forma integrada aos valores discretos de cor vermelha, verde e azul que vemos em uma imagem digital. Imagens coloridas (multiespectrais) possuem múltiplos canais, enquanto as em escala de cinza (monocromáticas) possuem um único canal. As imagens em escala de cinza representam a luminância (Y) de cada ponto de um cenário, já as coloridas representam a luminância e cromaticidade (informações de cor) desse cenário. Em Danson-Howe (2014), vemos que através da equação (1.1) é possível converter informações RGB (*red*, *green*, *blue*) em escala de cinza

$$Y = 0.299R + 0.587G + 0.114B \quad (1.1)$$

A Figura 1 ilustra dois cenários relacionados com cores primárias aditivas ou subtrativas. Em (a) temos três cores primárias aditivas, aquelas criadas adicionando-se luz colorida ao preto, que misturadas geram as cores ciano (azul e verde), magenta (vermelho e azul), amarelo (vermelho e verde) e branco (todas as três), já em (b) cores primárias subtrativas, criadas adicionando-se filtros à cor branca, que misturadas geram as cores azul (ciano e magenta), vermelho (magenta e amarelo), verde (ciano e amarelo) e preto (as três juntas).

Figura 1 - Cores primárias aditivas e subtrativas

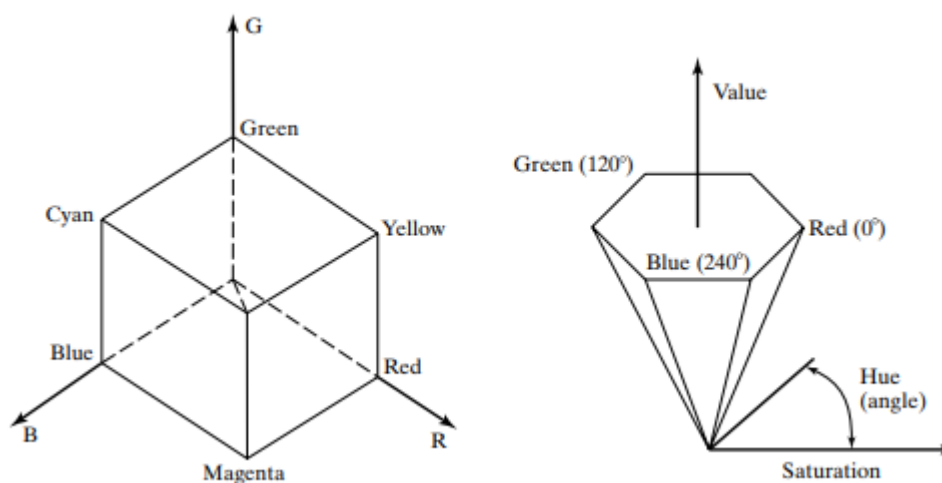


Fonte: Szeliski (2010).

Para Forsyth e Ponce (2012), as coordenadas de uma cor de um espaço linear nem sempre podem representar todas as propriedades necessárias para uma aplicação prática. Vemos na Figura 2 que às vezes se torna necessário trabalhar com o espaço HSV (*hue*, *saturation*, *value*), obtido através da planificação do cubo RGB em um hexágono. Dessa forma, este objeto se tornará a base da pirâmide se for considerada a luminosidade (*value*) como sendo sua altura, a matiz (*hue*) o ângulo em torno de um eixo neutro, eixo representado pela altura da pirâmide, e a saturação (*saturation*) a distância do eixo até as faces da pirâmide. Na prática, é a matiz que permite uma cor variar ao passar do vermelho ao verde, e a saturação concede a cor variar do vermelho para o rosa. O brilho (*brightness*), às vezes chamado de luminosidade (*lightness*) ou valor (*value*), possibilita a variação do preto ao branco. Os modelos HLS (*hue*, *luminense*, *saturation*) são frequentemente usados em visão computacional, pois além de separar luminância e crominância, divide ainda a

chrominância em matiz e saturação, permitindo-nos falar de azul escuro ou vermelho claro, por exemplo. (DANSON-HOWE, 2014).

Figura 2 - Cubo RGB e pirâmide hexagonal HSV



Fonte: Forsyth e Ponce (2012).

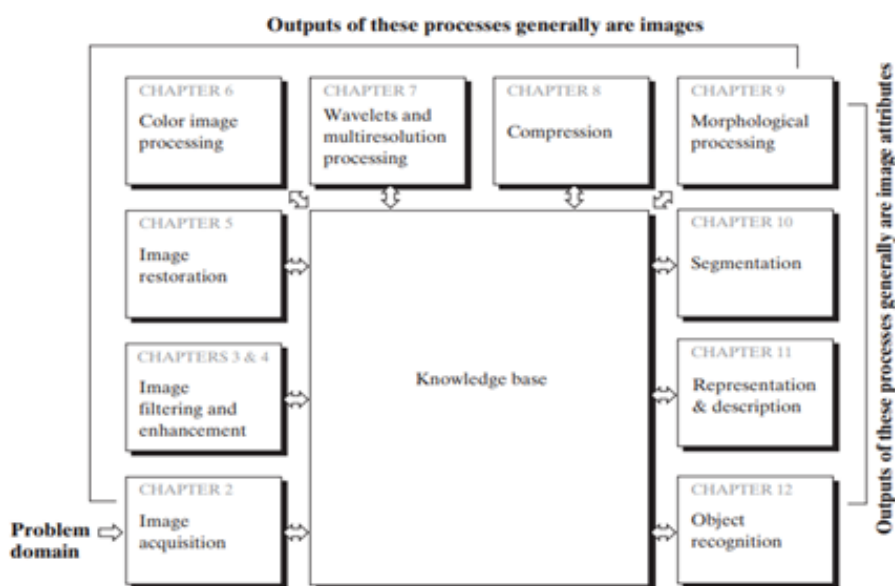
Uma vez que uma imagem é formada por meio da interação de elementos de uma cena 3D, iluminação, câmeras e sensores óticos, um estágio anterior às aplicações de Visão Computacional (VC) é o uso de processamento de imagem, objetivando convertê-la em uma forma mais adequada para análise posterior (SZELISKI, 2010). Para que um algoritmo de VC obtenha êxito na extração de atributos de um determinado objeto, é necessário por vezes, que a imagem desse objeto seja previamente ajustada por algum algoritmo de Processamento Digital de Imagens (PDI) (GONZALEZ; WOODS, 2008).

Em síntese, o Processamento Digital de Imagem (PDI) está relacionado à captura, tratamento e conhecimento dos dados capturados, por meio de imagens estáticas ou em movimento. De maneira mais detalhada, a Figura 3 mostra os caminhos desde a captura da imagem até a identificação do objeto em si. Vale ressaltar que esse processo não é necessariamente sequencial, e que etapas podem ser suprimidas.

Retornando a extração de atributos, um desafio é encontrar contornos oclusivos da imagem, pois são esses que fornecem a indicação da forma de um objeto. Existem algumas abordagens para a extração de atributos que combinam diversas características, locais (arestas e cantos) ou globais (cor), e ainda

agrupamento por estrutura ou pela forma e aparência (NIXON; AGUADO, 2020). Para Forsyth e Ponce (2012) um objeto é separado de seu *background* por um contorno oclusivo, significando que ao se desenhar uma trajetória na imagem que cruze o contorno oclusivo, teremos de um lado os pixels pertencentes ao objeto e do outro os pixels pertencentes ao *background*.

Figura 3 - Passos fundamentais do PDI



Fonte: Gonzalez e Woods (2008)

Em relação ao contorno de oclusão e sua importância na separação entre o interior do objeto e o *background* da imagem, é possível que os pontos pertencentes às fronteiras (*boundaries*) sejam suscetíveis a interpretações errôneas devido às mudanças acentuadas de contraste de uma imagem, principalmente em relação a variação de brilho que ocorrem nas bordas dessas imagens. Segundo Danson-Howe (2014), regularmente, entre os elementos de uma imagem, há uma pequena área que não é fotossensível, podendo assim, mesmo que improvável, ocasionar perda de informações dos objetos nessas regiões fronteiriças.

1.2 APRENDIZADO DE MÁQUINA (MACHINE LEARNING)

Para Kubat (2017), às vezes somos incapazes de definir certos objetos ou conceitos com precisão, e queremos transmiti-los para a máquina através de exemplos, que por sua vez, deve ser capaz de converter esses exemplos em

conhecimento. Uma coleção de atributos pode ser vista como um desses exemplos, pois desejamos que o sistema de aprendizado de máquina descubra padrões destes dados que foram medidos quantitativamente a partir de algum objeto ou evento. Tipicamente, representamos esses exemplos através como um vetor n-dimensional, onde cada coordenada desse vetor é um atributo (GOODFELLOW; BENGIO; COURVILLE, 2017). Murphy (2012) define o Aprendizado de Máquina (*Machine Learning*) como um conjunto de métodos capazes de detectar padrões, de maneira automática, em determinada quantidade de dados e, em seguida, utilizar esses padrões aprendidos para a previsão de dados futuros.

As vantagens referentes ao uso do Aprendizado de Máquina estão associadas com a resolução de problemas cujas soluções exigem diversas etapas de realizações manuais, ou diversas listas de regras, tipicamente em problemas complexos que não possuem método de resolução trivial ou ainda incompreensão do problema para uma grande quantidade de dados (GÉRON, 2019). Para Goodfellow, Bengio e Courville (2017), as tarefas de aprendizado de máquina são descritas em termos de como o sistema processa um exemplo e tenta ser capaz de emular objetos ou conceitos com acurácia convincente.

Podemos classificar os sistemas de aprendizado de máquina em quatro categorias: aprendizado supervisionado, o qual aprende através de pares de entrada/saída, onde o humano atua como um professor que supervisiona as saídas desejadas pelo modelo na forma de rótulos (*labels*); aprendizado não supervisionado, que trabalha apenas com valores de entrada conhecidos, porém sem previsões de saída; aprendizado semi-supervisionado, que trabalha com uma mistura entre dados rotulados e não rotulados sendo a maioria não rotulados; e por último, aprendizado por reforço, cujo sistema autorregula seu treinamento, premiando práticas exitosas contra si mesmo (MÜLLER; GUIDO, 2017; GÉRON, 2019). Conforme Raschka e Mirjalili (2017), esse agente autorregulador cria uma função de recompensa (*reward function*), capaz de mensurar a sua ação, sem necessidade de rotular ou atribuir um valor inteiro, tendo como objetivo a maximização dessa recompensa.

O uso dessas técnicas de aprendizado de máquina visa minimizar o erro para um determinado conjunto de treinamento. No entanto, para um conjunto de treinamento de baixa qualidade, que pode ser influenciado pela imprecisão dos exemplos rotulados, o modelo pode ser afetado por problemas de sobreajuste (*overfitting*), que normalmente estão associados com generalização e classificação

errônea (AWAD; KHANNA, 2015). Podemos dizer que nesses casos, os dados do conjunto de treinamento obtiveram bons ou ótimos resultados e a performance do modelo obteve resultado oposto com o conjunto dos dados de teste.

Outra classe de problemas, também alvo de resolução via técnicas de visão computacional, é a classificação. Um classificador executa um tipo de tarefa na qual o computador é solicitado a especificar em qual categoria determinada entrada de dados pertence. Um exemplo de classificação é o reconhecimento de objetos, cuja entrada é uma imagem, e a saída um código numérico que permite identificar o objeto nessa imagem (GOODFELLOW; BENGIO; COURVILLE, 2017). Alguns métodos de classificação de imagens são capazes de identificar objetos que sejam únicos e dominantes numa imagem (FORSYTH; PONCE; 2012). De maneira geral, um classificador é desenvolvido por meio de um conjunto de dados rotulados (método supervisionado), que são posteriormente utilizados para treinamento deste possível classificador, durante a classificação de novas imagens, a partir da análise dos atributos (semelhantes ou dessemelhantes) entre classes de imagens.

1.3 APRENDIZADO PROFUNDO (DEEP LEARNING)

O aprendizado de máquina convencional geralmente usa métodos de otimização e gradiente descendente para aprender modelos parametrizados. Exemplos de tais modelos incluem regressão linear, máquinas de vetores de suporte, regressão logística, redução de dimensionalidade e fatoração de matriz. De maneira simples, podemos dizer que as Redes Neurais (Neural Networks) são um tipo de aprendizado de máquina que tenta imitar o cérebro humano. Uma Rede Neural Artificial (*Artificial Neural Networks - ANNs*) tenta imitar o cérebro humano no seu nível mais básico, ou seja, o do neurônio. Para Braga, Ludermir e Carvalho (2007), o que torna as ANNs ferramentas eficazes na resolução de problemas, é sua capacidade de aprender por meio de exemplos e generalizar a informação aprendida, além disso, elas podem extrair dados explícitos e implícitos ao problema. Um neurônio artificial tem uma estrutura semelhante à de um neurônio humano e compreende as seguintes seções: *input layer*, *summation layer*, *activations layer* e *output layer* (SINGH; MANURE, 2020).

Dentro do aprendizado de máquinas, se destaca o subconjunto do aprendizado profundo (*Deep Learning*), considerado uma das maiores histórias de sucesso do

aprendizado de máquinas e visão computacional (ROSEBROCK, 2017). Para Patterson e Gibson (2017), uma definição aceita para *Deep Learning* é que esta consiste em rede neural com mais de duas camadas, mas fazem uma ressalva em relação ao significado do conceito de profundo (*deep*), comparado ao número de camadas, e apresentam um contexto histórico acerca do assunto. Podemos destacar o uso de redes neurais no campo da Bioinformática através dos exemplos citados por Braga, Ludermir e Carvalho (2007, p.201), são eles: “análise de expressão gênica, previsão de estrutura de proteínas, alinhamento de sequências, reconhecimento de genes, localização de proteínas e anotação automática”. Além desses, a de se mencionar ainda o seu uso em reconhecimento de imagens, cheiros, sabores, sons e classificações de ordem financeiras e logísticas, destacando aqui, vários exemplos que emergem dentro da agropecuária 4.0.

A vantagem do *Deep Learning* em comparação à abordagem tradicional de ANNs, reside no fato de ser capaz de extrair atributos mais robustos, o que torna o *Deep Learning* uma opção mais atrativa diante de problemas complexos (GOODFELLOW; BENGIO; COURVILLE, 2017). Patterson e Gibson (2017) descrevem alguns aspectos que levaram a evolução das redes neurais, são eles: maior quantidade de neurônios em relação às redes anteriores, formas mais complexas de conectar camadas/neurônios em uma rede neural, maior poder computacional disponível para treinamento e extração automática de atributos. Por extração de atributos, ele se refere ao processo da rede decidir quais características de um conjunto de dados podem ser usados como indicadores para rotular esses dados de maneira confiável.

As Redes Neurais Convolucionais (*Convolutional Neural Networks* ou *ConvNets*) são comumente usadas na classificação de imagens e detecção de objetos dentro da área de visão computacional. Esta metodologia conecta múltiplos *inputs* em um único *output*, em oposição aos modelos contendo vários *inputs* e *outputs*. Visualmente, as *ConvNets* codificam regiões da imagem, muito semelhante com a forma a qual humanos visualizam uma imagem de alta resolução e tamanho, deslizando-se os olhos sobre ela. Posteriormente, cria-se uma visão compacta dos dados. O trabalho envolvendo tais agrupamentos de regiões compactas agregam maior precisão e velocidade computacional, uma vez que menos conexões são envolvidas (BALLARD, 2018).

Os diferentes filtros, utilizados durante os estágios de convolução, são responsáveis pelo aprendizado relacionado às diferentes características pertencentes aos dados de entrada de cada camada, e fica a cargo das funções de ativação (*activation function*) gerarem as saídas dessas camadas. Considerando-se como dados da camada de entrada dessa rede uma imagem RGB, podemos imaginar essa imagem como uma matriz bidimensional constituída de três canais de cores sobrepostas em cada um dos seus pixels. O objetivo da *ConvNets* é reduzir essa imagem, através de um mapa de características (*feature map*) a fim de facilitar o processamento computacional.

A convolução pode ser definida pelo produto escalar entre todos os pesos contidos nos filtros (*kernel*), pequenas matrizes que percorrem (*convolve*) a imagem de entrada, resultando em um mapa de característica. O conjunto desses mapas de características é conhecido como camada de convolução (*convolutional layers*) e é usado para mapear as ativações da próxima camada (cujas ativações retêm relações com a camada anterior), preservando as principais características da imagem até se determinar, na última camada, a matriz de probabilidades e então classificar essa imagem em determinadas classes.

1.4 SEGMENTAÇÃO SEMÂNTICA E INSTANCIADA

Em relação ao processo que antecede a classificação de imagens, é interessante ressaltar a necessidade prévia de se aplicar técnicas para detecção e isolamento dos objetos (*object detection*) a serem classificados (contidos em uma imagem). Para descrever a localização desses objetos, geralmente são usadas caixas delimitadoras (*bounding boxes*) que consistem em um retângulo determinado pela região compreendida entre as coordenadas x e y (superior esquerda e inferior direita do retângulo) que contém o objeto de interesse. Os algoritmos de detecção de objetos costumam usar diversas caixas delimitadoras, com diferentes tamanhos e proporções, a fim de prever qual dessas contém o alvo com maior precisão. Elas são chamadas de caixas âncoras (*anchor boxes*) e são usadas durante o treinamento para a detecção de objetos, contendo dois *labels* cada, o da categoria do objeto contido, e o referente ao seu deslocamento em relação a caixa delimitadora da verdade terrestre (*ground-truth bounding box*). Quando muitas caixas são geradas para um mesmo alvo, utiliza-se o índice de Jaccard para a mensuração da similaridade entre duas caixas,

determinada pela razão entre as áreas de intersecção e união, provenientes de duas caixas delimitadoras. Aplica-se então o algoritmo de supressão não-máxima (Non Maximum Suppression) para remover caixas delimitadoras com predições similares, simplificando assim os resultados (ZANGH et al., 2021).

Além do próprio método de detecção de objetos, destacam-se a segmentação semântica e segmentação instanciada (ou segmentação de instâncias). Esta última concerne a detecção de objetos e a localização dos diferentes objetos de uma mesma imagem (identificá-los dentro de caixas delimitadoras). A segmentação semântica se encarrega de tentar segmentar imagens em regiões contendo diferentes categorias semânticas, regiões estas capazes de rotular (*Dog*, *Cat* e *Background*) e predizer objetos com bordas ao nível do pixel, conforme Figura 4.

Figura 4 - Imagem semanticamente segmentada com áreas rotuladas



Fonte: Zangh et al (2021)

Para Zangh et al. (2021), a segmentação instanciada é chamada de detecção e segmentação simultâneas, pois ao contrário da semântica, ela não distingue apenas a semântica, mas sim duas instâncias de um objeto. Caso essa imagem contivesse dois cachorros e um gato, por exemplo, cada um dos três estariam representados (após a finalização da segmentação) com cores distintas em relação aos pixels, enquanto a outra deixaria ambos os cachorros com a mesma cor e o gato diferenciado.

Para He et al. (2017) a segmentação instanciada é um desafio, porque além de detectar corretamente os objetos de uma imagem, requer segmentar precisamente cada instância, como vemos na Figura 5. Nessas imagens, o fundo continua intacto, mas caso fosse necessário segmentá-lo, existe ainda a segmentação panorâmica, que conseguiria segmentar todos os objetos contidos na imagem, além de diferenciá-los por instâncias.

Figura 5 - Mask R-CNN results on the COCO test set












Fonte: He et al. (2017)

1.5 PROBLEMAS RELACIONADOS À CARNE BOVINA

A Agência Canadense de Classificação da Carne - CBGA (*Canadian Beef Grading Agency*) é gerida por um conselho administrativo composto por representantes da indústria, criadores de gado, fabricantes de embalagens, e varejistas, cujo objetivo é garantir uma avaliação imparcial do valor da carne bovina, de bisonho e vitela deste país, conforme descrito em seu *website* oficial. Dentro de todas as possíveis avaliações dos cortes cárneos que possam ser requeridas, o foco deste trabalho se dará na classificação do escore de marmoreio (*marbling score*), baseada na quantidade, tamanho e distribuição de gordura intramuscular (GIM) depositadas na área de olho de lombo (AOL), localizada entre a décima segunda e décima terceira costela do músculo *Longissimus*.

A Figura 6 mostra os padrões de GIM representativos das quatro principais classes canadenses comparadas às norte-americanas, porém o foco do trabalho é apenas no escore canadense: Canada A, AA, AAA e Prime. Essas classes podem ainda ser subdivididas em 13 avaliações distintas, se considerarmos fatores relacionados à maturidade (idade), conformação (musculatura), avaliação de marmoreio (cor da carne, textura, marmoreio) e gordura (cor, textura e cobertura), sendo estas características atreladas à maciez, suculência e o sabor que podem afetar na aceitabilidade e escolha dos consumidores.

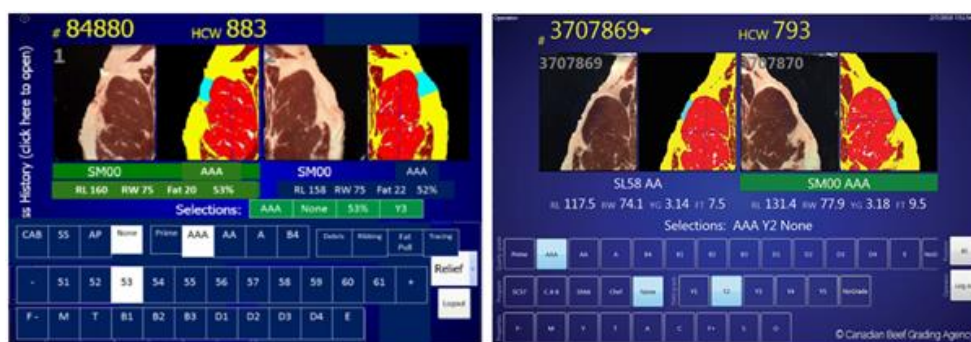
Figura 6 - Padrões de marmoreio canadense agrupado por categorias (A, AA, AAA e Prime)

CANADA	MARBLING SCORE		USA
Canada Prime	Abundant		USDA Prime
	Moderately Abundant		
	Slightly Abundant'		
Canada AAA	Moderate		USDA Choice
	Modest		
	Small'		
Canada AA	Slight'		USDA Select
Canada A	Trace		USDA Standard
	Practically Devoid		

Fonte: GBGA (2020)

Já existem no mercado diferentes instrumentos que auxiliam os auditores na classificação de cada corte, fornecendo a pontuação de marmoreio e sua classe de rendimento, através da análise de visual computacional da área do olho de lombo baseada no cálculo de pixels vermelhos e brancos contidos dentro do músculo. Um dos sistemas de visão computacional aprovados pela CFIA (*Canadian Food Inspection Agency*) é o “e + v Technology” (Figura 7) introduzido em 1999. Essa ferramenta, por si só, não dispensa o avaliador certificado pela CBGA, pois este pode substituir o escore atribuído pela ferramenta em cortes ditos não precisos.

Figura 7 – “e + v Technology”

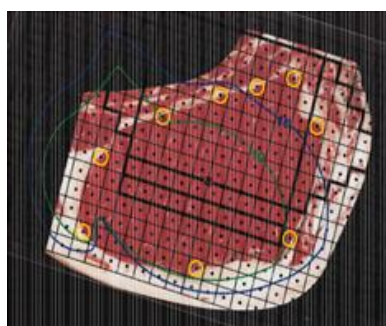


Fonte: CBGA (1999/2020)

A Figura 8 ilustra um método manual disponibilizado pela USDA para avaliação aproximada da área de olho da carcaça bovina. O método consiste em sobrepor à

carcaça, independentemente da orientação adotada, uma grade contendo pequenos quadrados de 0,1 in² (polegada quadrada). Uma ressalva ao se utilizar esse processo, está na padronização da sobreposição da grade, pois diferentes eixos podem gerar divergências na mensuração de uma mesma área. Assim, é possível notar que há tempos a indústria frigorífica vem demandando análises instrumentais que possam substituir classificações subjetivas da linha de produção e que possam ser capazes de prever características de qualidade das carcaças.

Figura 8 - USDA ribeye grids



Fonte: USDA (2011)

Utilizando-se de amostras de carne suína e bovina, Barbon et al. (2017), criaram um método que segmenta imagens pré-processadas, analisa os pixels de GIM e realiza uma aprendizagem baseada no algoritmo *k-NN* (*k-Nearest Neighbours*), capaz de avaliar o nível de marmoreio no músculo *Longissimus thoracis*, removidos da penúltima e última costela. O sistema criado pelos pesquisadores obteve acurácias iguais a 81,59% (bovinos) e 76,14% (suínos), para 335 amostras suínas e 45 bovinas da raça Nelore. Lee, Yonn e Choi (2019) demonstraram uma forte correlação entre GIM e AOL ($r = 0,80$ e $0,88$, respectivamente) em carcaças do músculo *Longissimus thoracis* de novilhos Hanwoo. Foram utilizadas na pesquisa imagens digitais posteriormente convertidas em imagens binárias. Depois da binarização, determinou-se o número de manchas de marmoreio na AOL, e o método Soxhlet foi aplicado a fim de se determinar o percentual de GIM.

Meunier et al. (2021) desenvolveram um método para análise de imagens utilizando-se do software *ImageJ*, programa para processamento de imagens (contemplando também as oriundas de celular smartphone/android), e obtiveram uma acurácia equivalente a 90% durante a predição de GIM em amostras de carcaças bovinas de diferentes raças, obtidas entre a 5^a e 6^a vértebras. Foi utilizado para fins

de comparação de eficiência, os resultados obtidos pelo método Soxhlet, além de três abordagens de ML durante o desenvolvimento desse método: *Sparse Partial Least Squares* (SPLS), *Random Forest* (RF) e *Multiple Linear Regression* (MLR) combinados com *Cluster of Variables* (ClustOfVar).

Iglesia et al. (2020) realizaram diferentes tipos de análises não invasivas para classificação de carcaças bovinas da raça *Retinta* (*Bos taurus*) e compararam seu método com as tradicionais avaliações feitas por seis peritos de um frigorífico espanhol. Para extrair as bordas de toda carcaça do animal, e dividi-la em 18 regiões de interesse, foram utilizados o algoritmo HED (*Holistically-Nested Edge Detection*), baseado em CNN, além de métodos de aprendizado profundo supervisionado (*Deeply Supervised networks*). Os pesquisadores preferiram esse método quando comparado ao algoritmo *Canny*, utilizado para detecção de bordas. Uma técnica para reconhecimento facial foi incorporada ao trabalho, *Landmark Detection*, objetivando identificar pontos de referências para divisão da carcaça. Para treinamento e avaliação do método de classificação, os autores usaram o algoritmo de aprendizado supervisionado *Random Forest Regression-Voting*. Os resultados demonstram que o sistema criado por eles classificou as carcaças com alta precisão de até 10 vezes mais peças que os peritos.

Liu et al. (2018) compararam os métodos de regressão *stepwise* e *SVM* para predição de GIM em lombos de suínos. No modelo *stepwise*, atributos relacionados a cores foram extraídos e executados via método *bootstrap* e gráfico *scree plot* para auxiliar na Análise dos Componentes Principais (*Principal Component Analysis*), reduzindo para 3, de um total de 18 atributos possíveis, durante as análises posteriores. Já com a técnica de aprendizado de máquina *SVM regression*, todos os preditores foram utilizados no treinamento de dados, ou seja, a totalidade dos atributos de cores, 18 no total. Os resultados demonstraram que o uso de sistema de visão computacional é promissor, visto que o modelo de regressão *stepwise* previu corretamente o percentual de GIM em 64% das tentativas, enquanto o modelo *SVM* apresentou maior precisão, chegando a prever corretamente 75% das vezes.

1.6 ALGORITMOS DE CLASSIFICAÇÃO

O desafio da classificação está diretamente associado com a escolha dos melhores atributos de uma imagem, pois cada problema tem propriedades especiais.

Para Rosebrock (2017), a classificação de imagens (*image classification*) consiste em atribuir um rótulo (*label*) a uma imagem de um conjunto predefinido de categorias. Em um pensamento matemático associado aos estudos das funções, podemos dizer que a entrada de dados é uma imagem e a saída da função retorna um rótulo que irá categorizar a imagem. Segundo Kubat (2017), um classificador que apresente um bom comportamento no conjunto de treinamento não garante alto desempenho no futuro, e por este motivo devemos perguntar qual dos classificadores a ser escolhido possui a probabilidade de obter a melhor pontuação em exemplos futuros.

Em suas pesquisas por atributos significativos, Haralick, Shanmugam e Dinstein (1973) destacam três elementos padrões usados na interpretação humana de fotografias coloridas, sendo eles: atributos espectrais, responsáveis pela descrição das variações tonais em várias bandas do espectro visível e/ou porções infravermelhas de um espectro eletromagnético; atributos texturais, referentes às distribuições espaciais das variações tonais dentro de uma banda; e atributos contextuais, que possuem informações advindas dos blocos de dados pictóricos em torno da área a ser analisada. Para estes autores, o conceito de *tone* está relacionado às variações dos tons de cinza, enquanto *texture* se refere à distribuição estatística dos tons de cinza.

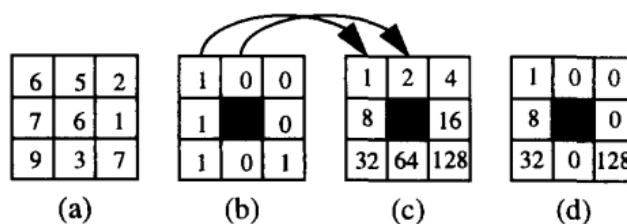
1.6.1 Padrões Locais Binários (Local Binary Pattern)

Segundo Nixon e Aguado (2020) a descrição da textura pelo LBP (*Local Binary Pattern*) é relativamente recente e logo ganhou visibilidade da comunidade científica devido aos seus atrativos recursos de desempenho, desde sua aparição em Ojala, Pietikäinen e Harwood (1996) e mais visibilidade ainda em Ojala, Pietikäinen e Mäenpää (2002). O método é baseado no reconhecimento de que certos padrões binários locais, denominados uniformes, representam propriedades fundamentais da textura local de uma imagem e que seu histograma de frequência é comprovadamente um atributo muito poderoso (OJALA; PIETIKÄINEN; MÄENPÄÄ, 2002). Suas aplicações têm se mostrado efetivas nas áreas de reconhecimento e detecção de faces, análise de expressões faciais e classificações demográficas, entre outras tarefas (HUANG et al., 2011).

A Figura 9 ilustra o método LBP proposto por Ojala, Pietikäinen e Harwood (1996) para descrever padrões locais binários puros, aplicado a uma região 3x3,

comparando-se o pixel central de seus 8 pixels vizinhos equidistantes uma unidade em escala de cinza [Fig 9(a)], e nos casos em que o pixel vizinho possua valor numérico maior ou igual ao pixel central, este armazenará o resultado 1, caso contrário, armazenará o resultado 0 [Fig 9(b)]. Após isso, essa sequência binária é multiplicada de maneira ponderada pelos pesos advindos das potências da base 2 [Fig 9(c)]. Por fim, a somatória dos 8 pixels não nulos resulta no número 169 desta unidade de textura [Fig 9(d)].

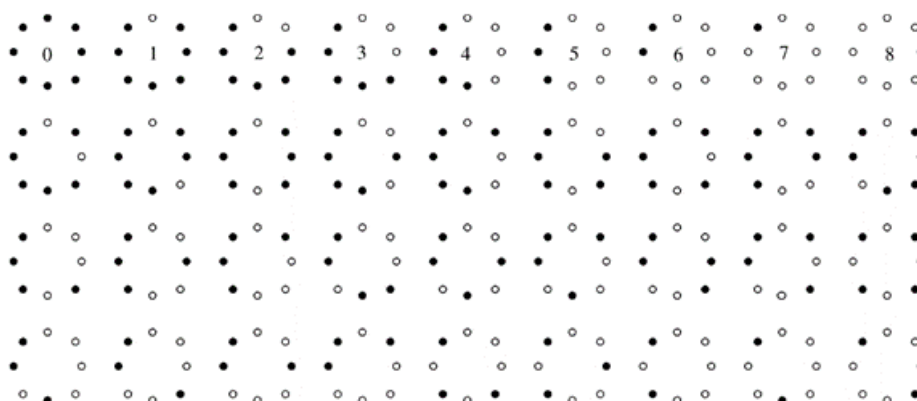
Figura 9 - Two-level version (LBP) of the texture unit



Fonte: Ojala et al. (1996)

A fim de alcançar uma melhor habilidade discriminativa, Ojala, Pietikäinen e Mäenpää (2002) perceberam que alguns padrões básicos dominavam a capacidade de descrever texturas. Os padrões binários uniformes são caracterizados por possuírem no máximo duas transições 0/1. Na Figura 10 temos nove padrões uniformes numerados de 0 a 8, os demais são considerados não-uniformes. É utilizado a terminologia $LBP_{8,R}^{riu^2}$ para indicar padrões binários invariantes de rotação uniformes. O padrão #0 representa o limite para um ponto brilhante, enquanto #8 representa um ponto escuro local (NIXON; AGUADO, 2020).

Figura 10 - Padrões binários invariantes de rotação para uma vizinhança circular simétrica $LBP_{8,R}^{riu^2}$



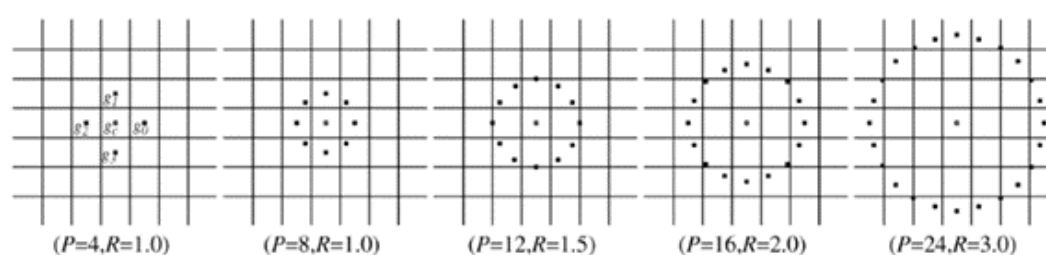
Fonte: Ojala, Pietikäinen e Mäenpää (2002)

Para Huang et al. (2011) as mais importantes propriedades do LBP são sua tolerância a respeito das mudanças de iluminação monotônicas e sua simplicidade computacional. Ahonen, Hadid e Pietikäinen (2006) utilizaram histogramas LBP concatenados num único histograma de atributos espacialmente melhorados representando eficientemente a imagem facial, cuja área da face foi dividida em pequenas regiões para aplicação do LBP, pois a ideia por trás era que as imagens da face podem ser compostas de micro padrões invariantes com respeito às transformações em tons de cinza.

Para determinar os vizinhos circularmente simétricos $g_p (p = 0, 1, \dots, P - 1)$ em um sistema de coordenadas cartesianas, é necessário primeiramente transformar esses pontos em coordenadas polares $(-R \cdot \sin(2\pi p/P), R \cdot \cos(2\pi p/P))$, onde R representa o raio, em pixels, e P a quantidade de elementos pertencentes a vizinhança do pixel central g_c , para um conjunto (P, R) em escala de cinza (Figura 11). Segundo Ojala, Pietikäinen e Mäenpää (2002), os valores de cinza dos vizinhos que não caem exatamente no centro do pixel, são estimados por interpolação.

Para este estudo, focaremos no uso dos métodos LBP para fins de extração de atributos, como fonte de entrada de dados para dois classificadores distintos, as Árvores de Decisão (*Decision Tree*) e Florestas Aleatórias (*Random Forest*).

Figura 11 - Vizinhança circularmente simétrica $LBP_{8,R}^i$



Fonte: Ojala, Pietikäinen e Mäenpää (2002)

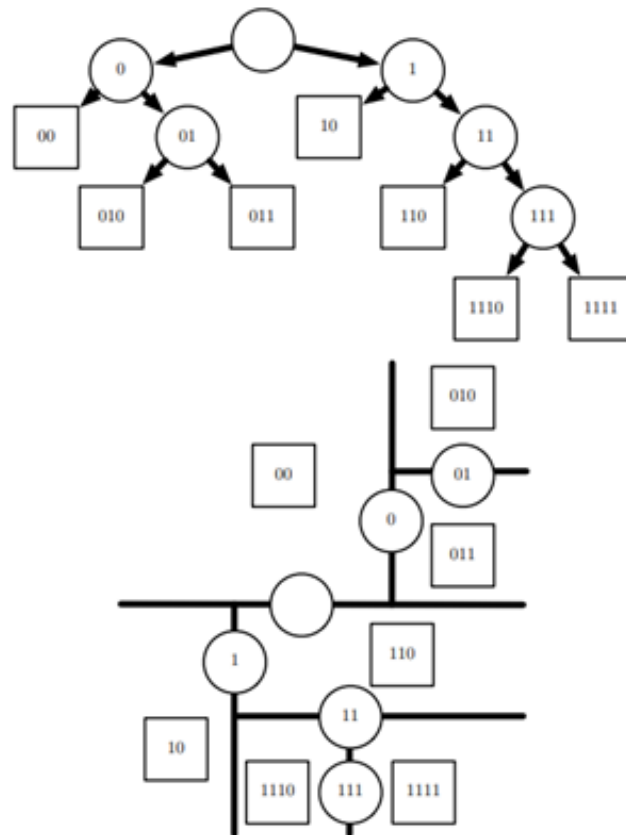
1.6.2 Árvores de Decisão (Decision Tree)

O algoritmo de árvores de classificação e regressão (CART) é um método de classificação com foco na construção de uma árvore de decisão que utiliza como ponto chave o conceito de índice de impureza (Gini index) como critério principal de divisão. CART pode ser representado como uma árvore binária construída via divisão dos nós em outros dois nós filhos de forma repetitiva (KUZILEK et al., 2014). Conhecidas como

Árvores de Decisão, *Classification and Regression Trees* (CART), elas são descritas através do particionamento recursivo do espaço de entradas e definindo-se um modelo local, em cada uma das regiões resultantes. Podemos representá-las por uma árvore, com uma folha por região (MURPHY, 2012). Nas Árvores de Classificação (*Classification Tree*) o objetivo é classificar valores discretos, enquanto nas Árvores de Regressão (*Regression Tree*) o objetivo é prever valores contínuos. A diferença entre ambas está no fato que uma prevê uma classe para cada nó, enquanto a outra prevê um valor.

Para Shalev-Shwartz e Ben-David (2014), uma árvore de decisão é um preditor do tipo, $h: X \rightarrow Y$, que prevê o label associado a uma instância x , ao percorrer de um nó da raiz (*root node*) de uma árvore em direção a folha (*leaf*). Em uma classificação binária, tem-se $Y = \{0, 1\}$, porém as árvores binárias podem ser aplicadas em outros problemas de predição. Normalmente, a divisão é baseada em um dos atributos de x ou num conjunto predefinido de regras. Uma das vantagens de usar Árvores de Decisão reside no fato de não gastar tempo demasiado na preparação dos dados (GÉRON, 2019).

A Figura 12 ilustra duas implementações de uma Árvore de Decisão. Primeiramente, no topo, vemos um nó raiz (profundidade 0) que escolhe enviar um dos possíveis exemplos de entrada para o nó filho (*child node*) esquerdo da raiz, ou para o nó filho direito da raiz (profundidade 1). As escolhas se sucedem para os nós filhos direito ou esquerdo (profundidade 2), até que o momento que se encerram as escolhas, chegando ao nó das folhas (*leaf nodes*). No exemplo, os nós internos são desenhados como círculos, e os nós das folhas como quadrados, sendo estes não geradores de novos nós filhos. Na sequência (segunda parte da figura localizada na parte inferior), temos uma possível divisão no plano 2-D, com cada nó interno desenhado ao longo de uma linha divisória, usada para categorizar o exemplo e os nós das folhas, desenhados no centro das regiões que os recebem. Cada folha requer pelo menos um exemplo de treinamento. Quando uma área é considerada “pura”, ela não poderá mais ser dividida (GOODFELLOW; BENGIO; COURVILLE, 2017; GÉRON, 2019).

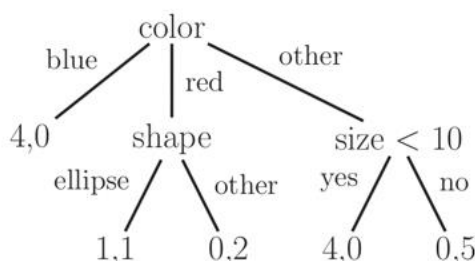
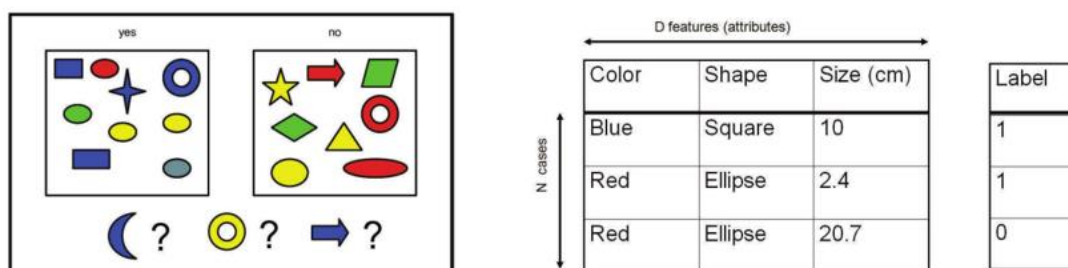
Figura 12 - Diagrama descrevendo o *Decision Tree*

Fonte: Goodfellow, Bengio e Courville (1998).

Murphy (2012) apresenta um novo exemplo de árvore de decisões cujo objetivo é checar a forma, cor e tamanho de um objeto, e assim classificá-lo (Figura 13). Inicialmente a cor do objeto é checada quanto a coloração, e caso a cor azul seja identificada, segue-se para o ramo esquerdo (terminando o subprocesso na folha intitulada “4,0”). Até este ponto avaliado, 4 exemplos positivos e 0 negativos correspondentes ao critério azul. Para essa folha, pode-se prever que $p(y = 1|x) = 4/4$ se x for azul. Se o objeto for vermelho, então sua forma é verificada, pois nesse caso, a forma representa uma condição (nó interno), e na sequência ela caminha para duas folhas. Caso ela seja uma elipse vermelha, então será encaminhada para a folha intitulada “1,1”, caso contrário, para a folha “0,2”. Suas previsões são $p(y = 1|x) = 1/2$ e $p(y = 1|x) = 0/2$, respectivamente. É possível verificar que há uma elipse vermelha na classe “yes” e outra na classe “no”, por isso temos como resultado o valor “1,1”. De forma semelhante, enquanto temos dois objetos vermelhos na classe “no” que não são elipses (flecha e coroa circular), obtivemos assim o valor “0,2”. Para situações nas quais o objeto não seja azul ou vermelho, então o tamanho será

checado. Com exceção da folha “1,1”, todas as demais são consideradas puras, pois seus exemplos pertencem a uma classe ou outra.

Figura 13 - Árvore de decisão aplicada a determinação das formas, cores e tamanho de um objeto baseadas na figura de Leslie Kaelbling.



Fonte: Adaptado de Murphy, 2012.

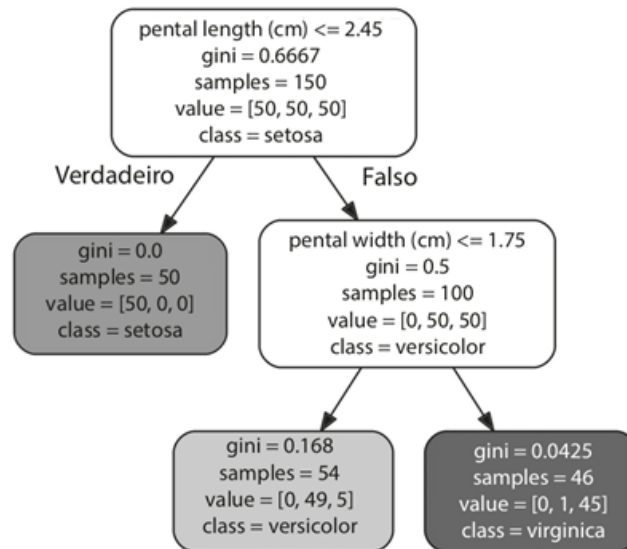
O algoritmo Árvore de Decisões conta ainda com alguns atributos que podem ser úteis durante o treinamento em um determinado nó, quando usado por meio da biblioteca *Scikit-Learn/Python*, que através do algoritmo CART, irá gerar apenas árvores binárias, ou seja, os nós sem folhas sempre terão filhos, respondendo às perguntas com sim/não. Conforme apresentado na Figura 14, o atributo *Samples* indicará a quantidade de instâncias de treinamento daquele nó, *Value* indicará a distribuição dessas instâncias por classes e, por fim, o *coeficiente de Gini* indicará o grau de impureza: 0 quando um nó for puro, ou seja, todas as instâncias de treinamento pertencem a uma mesma classe. (GÉRON, 2019).

A equação (1.2) mostra o cálculo do *coeficiente de gini*, G_i , do i -ésimo nó. Como exemplo, vemos na Figura 14, que o nó esquerdo (profundidade 1) é puro, pois seu coeficiente de gini é igual a $1 - (50/50)^2 - (0/50)^2 - (0/50)^2 = 0$. Por padrão, a medida do coeficiente é aplicada pelo *Scikit-Learning* para determinar o grau de impureza, porém é possível modificar o critério para a opção de entropia.

$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2 \quad (1.2)$$

onde $p_{i,k}$ é a média das instâncias da classe k entre as instâncias de treinamento de um nó i .

Figura 14 - Árvore de Decisão da íris



Fonte: Géron (2019).

Géron (2019) fornece a equação (1.3) para o custo que o algoritmo CART tenta minimizar durante o treinamento das árvores de decisão. Inicialmente, o algoritmo divide o conjunto de treinamento em dois subconjuntos utilizando um atributo k e um limitante t_k , e na sequência ele escolhe o par (k, t_k) que produz os subconjuntos mais puros. A divisão continua recursivamente até que se atinja a profundidade máxima (max_depth) ou não consiga minimizar a impureza.

$$J(k, t_k) = (1/m) \cdot m_{esquerda} \cdot G_{esquerda} + (1/m) \cdot m_{direita} \cdot G_{direita} \quad (1.3)$$

onde :

$G_{esquerda/direita}$ mede a impureza do subconjunto esquerdo/direito

$m_{esquerda/direita}$ é o número de instâncias no subconjunto esquerdo/direito

A Tabela 1 foi alterada para efeitos comparativos do custo do CART para diferentes valores de (k, t_k) contidos no conjunto de dados *Iris* (SHIN, 2014), tendo como exemplo o nó da raiz (profundidade 0), cujo atributo k é *petal length* (cm) e t_k variando e m igual a 10, pela facilidade dos cálculos.

Tabela 1 - 10 amostras do conjunto iris

ID	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Target
1	5,1	3,5	1,4	0,2	Iris-setosa
2	4,9	3	1,4	0,2	Iris-setosa
3	4,7	3,2	1,3	0,2	Iris-setosa
51	7	3,2	4,7	1,4	Iris-versicolor
52	6,4	3,2	4,5	1,5	Iris-versicolor
53	6,9	3,1	4,9	1,5	Iris-versicolor
101	6,3	3,3	6	2,5	Iris-virginica
102	5,8	2,7	5,1	1,9	Iris-virginica
103	7,1	3	5,9	2,1	Iris-virginica
104	6,3	2,9	5,6	1,8	Iris-virginica

Fonte: Adaptado de Shin, 2014.

Na tabela 2, é possível observar como o custo CART é calculado. Ao tomar como exemplo, um valor para t_k igual a 4.7 (linha 4), os IDs 1, 2, 3, 51 e 52 possuem valor menor ou igual a 4.7. Logo, haverá 5 elementos de um total de 10, que satisfazem essa restrição, ficando 5 instâncias à esquerda, e outras 5 à direita. Após substituir esses valores na função custo CART, obtemos o valor 0.85, sendo descartado por não apresentar o valor que minimize seu objetivo. O exemplo é ilustrativo e não reflete o CART da Figura 14. Géron (2019) comenta que encontrar uma árvore ideal se torna um problema praticamente intratável, mesmo para conjuntos de treinamento muito pequenos, e que é aceitável se contentar com uma solução razoavelmente boa.

Tabela 2 - Exemplos aplicados na função custo CART

t_k	m_{esq}	m_{dir}	$(1/m) \cdot m_{esquerda} \cdot G_{esquerda} + (1/m) \cdot m_{direita} \cdot G_{direita}$	$J(k, t_k)$
1	0	10	$(1/10)[0 \cdot (1 - 0^2 - 0^2 - 0^2) + 10 \cdot (1 - (3/10)^2 - (3/10)^2 - (4/10)^2)]$	0,66
1,3	1	9	$(1/10)[1 \cdot (1 - (1/10)^2 - 0^2 - 0^2) + 9 \cdot (1 - (2/10)^2 - (3/10)^2 - (4/10)^2)]$	0,738
1,4	3	7	$(1/10)[3 \cdot (1 - (3/10)^2 - 0^2 - 0^2) + 7 \cdot (1 - 0^2 - (3/10)^2 - (4/10)^2)]$	0,798
4,7	5	5	$(1/10)[5 \cdot (1 - (3/10)^2 - (2/10)^2 - 0^2) + 5 \cdot (1 - 0^2 - (1/10)^2 - (4/10)^2)]$	0,85
4,9	6	4	$(1/10)[6 \cdot (1 - (3/10)^2 - (3/10)^2 - 0^2) + 4 \cdot (1 - 0^2 - 0^2 - (4/10)^2)]$	0,828
5,1	7	3	$(1/10)[7 \cdot (1 - (3/10)^2 - (3/10)^2 - (1/10)^2) + 3 \cdot (1 - 0^2 - 0^2 - (3/10)^2)]$	0,84
5,6	8	2	$(1/10)[8 \cdot (1 - (3/10)^2 - (3/10)^2 - (2/10)^2) + 2 \cdot (1 - 0^2 - 0^2 - (2/10)^2)]$	0,816
5,9	9	1	$(1/10)[9 \cdot (1 - (3/10)^2 - (3/10)^2 - (3/10)^2) + 1 \cdot (1 - 0^2 - 0^2 - (1/10)^2)]$	0,756
6	10	0	$(1/10)[10 \cdot (1 - (3/10)^2 - (3/10)^2 - (4/10)^2) + 0 \cdot (1 - 0^2 - 0^2 - 0^2)]$	0,66

Fonte: Adaptado de Shin, 2014.

Para evitar sobreajuste (*overfitting*) é possível parar o crescimento da árvore caso a diminuição do erro não seja suficiente para justificar subárvores extras. Normalmente, deixa-se a árvore crescer completamente para a realização de uma poda (*pruning*) futura (MURPHY, 2012). Por meio do Scikit-Learning (biblioteca para aprendizado de máquina implementada em Python), pode-se restringir a profundidade máxima da árvore via alteração do hiperparâmetro *max_depth*, reduzindo-se assim, o risco de sobreajuste. Normalmente, a poda é realizada de baixo para cima, onde cada nó pode ser substituído por uma de suas subárvores ou por uma folha, com base em algum limite ou estimativas estatísticas (SHALEV-SHWARTS, BEN-DAVID, 2014).

Para Shalev-Shwartz e Ben-David (2014), a estrutura geral para o crescimento de uma árvore se inicia com uma única folha (raiz), a qual atribui-se um *label* de acordo com a maioria dos votos entre todos os *labels* possíveis do conjunto de treinamento. Define-se uma medida de ganho que quantifica a melhoria na divisão dessa folha, e a divide (ou não) escolhendo-se o ganho máximo como objetivo. Em resumo, uma árvore de decisão classifica um vetor de atributos obtendo uma votação majoritária entre os exemplos rotulados que alcançaram uma mesma folha (FORSYTH; PONCE, 2012).

1.6.3 Florestas Aleatórias (Random Forest)

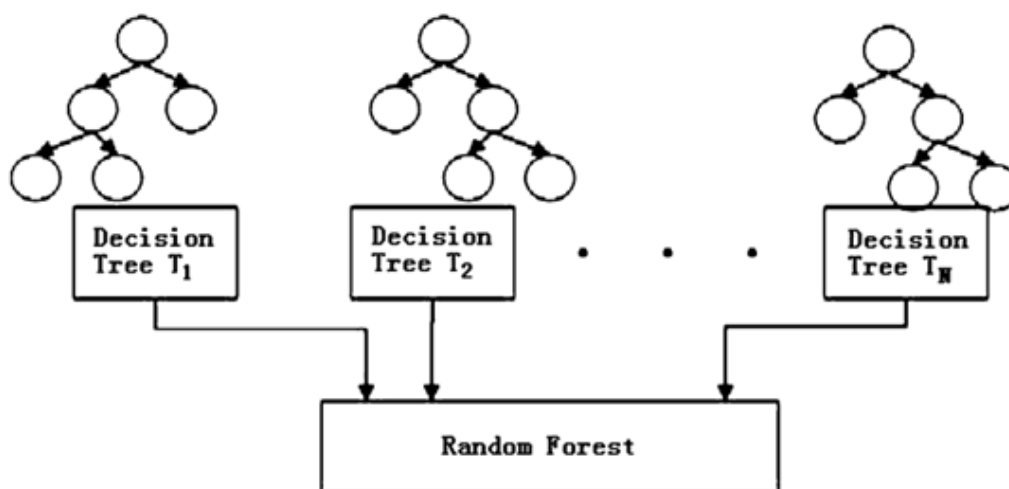
Em algumas situações torna-se necessário agregar às previsões um conjunto de previsores (*Ensemble*), sejam classificadores ou regressores, para fins de se obter uma previsão superior ao método tradicional preditivo (individual). Essa técnica é denominada *Ensemble Learning*. Um *Ensemble* de Árvores de Decisão, treinadas cada uma em subconjuntos aleatórios diferentes do conjunto de treinamento, é chamado de Florestas Aleatórias (GÉRON, 2019). Para Li e Jain (2011), uma Floresta Aleatória é um método de combinação de classificadores que consiste na união de N árvores binárias, onde cada uma dessas árvores é considerada como um classificador fraco. A estrutura da Floresta Aleatória combina todos esses classificadores fracos em um classificador robusto, sendo seu *output* computado pela votação de cada árvore binária.

Por meio da Figura 15, ilustramos N árvores de decisão, treinadas através de abordagens aleatórias, e para cada uma das árvores de decisão, T_n , as amostras são selecionadas aleatoriamente de um conjunto de todos os treinamentos possíveis.

Depois do treinamento das N árvores, a decisão final combina todos os outputs, T_1, T_2, \dots, T_n , considerando a média de todos (LI; JAIN, 2011).

Os métodos *Ensemble* apresentam melhora quando existe a possibilidade de se utilizar previsores independentes, uma vez que isso aumenta a chance de cometerem diferentes tipos de erros, melhorando a acurácia do *Ensemble*. Após reunir as previsões de cada classificador distinto, pode-se usar o classificador denominado *hard voting*, caso o objetivo for prever a classe que obteve a maioria dos votos. Assim, torna-se possível determinar a probabilidade de cada classe e caso seja desejável escolher a classe com maior probabilidade entre todos os classificadores, tem-se um classificador *soft voting* (GÉRON, 2019).

Figura 15 - Floresta Aleatória combinando todas as saídas de todas as Árvores de Decisão



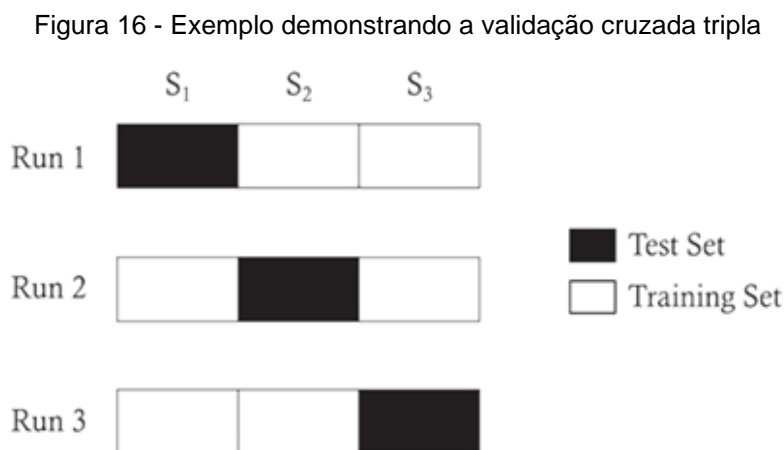
Fonte: Li e Jain (2011).

Para Murphy (2012), uma maneira de reduzir a variância de uma estimativa se dá pela determinação da média de várias estimativas. Esta técnica é chamada *bagging*, que significa “*bootstrap aggregating*”. Florestas aleatórias melhoram a performance por meio de empacotamento (*bagging*), selecionando-se aleatoriamente um subconjunto de variáveis de entrada para cada etapa recursiva do processo de treinamento (FORSYTH; PONCE, 2012). Caso a escolha seja a de se utilizar o mesmo algoritmo de treinamento, mas decidindo por treiná-los em subconjuntos aleatórios e distintos, duas opções podem ser usadas: *bagging* (amostragem com reposição), ou *pasting* (amostragem sem reposição). Usando-se a biblioteca *Scikit-Learn*, é possível escolher entre *bagging* (*bootstrap=True*) ou *pasting* (*bootstrap=False*) dentro da

biblioteca *sklearn.ensemble*, após importação do pacote *BaggingClassifier* (GÉRON, 2019).

James et al. (2015) declaram que aproximadamente dois terços das instâncias, em média, são aproveitadas pelas árvores (*bagged*) durante um treinamento, e que um terço dessas observações, não são usadas neste passo. Tais instâncias são caracterizadas como “fora do saco”, ou instâncias *oob* (*out-of-bag*). Pode-se utilizar esse conjunto de observações *oob* para estimar o erro de um previsor. Para Géron (2019), pode-se avaliar o Ensemble utilizando-se a média das avaliações *oob* de cada previsor, uma vez que o previsor não utiliza essas instâncias durante seu treinamento, evitando-se assim o uso de validação cruzada (*cross validation*).

A validação cruzada é um método que busca avaliar um modelo, utilizando-se de todas as instâncias rotuladas durante as fases de treinamento e teste. A Figura 16 ilustra um conjunto de dados rotulado e dividido aleatoriamente em três subconjuntos de igual tamanho. Em cada iteração da avaliação, computa-se as taxas de erro dos subconjuntos testes (preto), com base nos dois subconjuntos usados para treinamento (branco). Ao final, a taxa geral de erros dos testes é calculada somando-se os erros individuais em cada teste, e dividindo-os pelo total de iterações (TAN et al., 2019).



Fonte: Tan et al. (2019).

Ao se usar Florestas Aleatórias, espera-se como uma das de suas vantagens principais, a facilidade na determinação de quais atributos serão importantes para o problema. O algoritmo pesquisa o melhor atributo dentro de um subconjunto aleatório de atributos, ao invés de buscar a melhor característica ao dividir um nó, gerando

assim uma grande diversidade de árvores, trocando um alto viés por uma baixa variância (GÉRON, 2019).

2 DESENVOLVIMENTO

2.1 PROBLEMÁTICA

Considerando o contexto da indústria de carne bovina, a classificação de carcaças desempenha papel fundamental na medida em que promove uma ferramenta avaliadora de características intrínsecas à carne, em busca do fornecimento de um produto desejado pelo consumidor final no momento da compra. Há tempos a indústria frigorífica demanda por análises instrumentais que possam substituir classificações subjetivas da linha de produção e serem capazes de prever características de qualidade das carcaças. A avaliação de escore de marmoreio é um dos critérios que compõem as classificações de carcaças, por interferir diretamente na decisão de consumo (CHENG et al., 2015). As classificações de carcaças, desempenhadas de maneira visual por técnicos, são particularmente susceptíveis a erros pelo fato de haver grande quantidade de material a ser analisado. Em alguns casos, usam-se até mesmo técnicas destrutivas na avaliação de carcaças. Além disso, a extensão dos depósitos de gordura intramuscular nas carcaças é um fator que agrava tal problema (BARBON et al., 2017; LEE; YOON; CHOI, 2019).

Em frigoríficos de pequeno porte, quando possível, cada carcaça recebe um escore relativo a uma escala pré-definida, levando em conta atributos como marmoreio, maciez, *Warner-Bratzler Shear Force* (WBSF) (força de cisalhamento), entre outras características. A qualidade da carne e os atributos de palatabilidade são caracterizados principalmente pela maciez (textura), suculência e sabor (AASLYNG, 2002; HOSSEINPOUR; ILKHCHI; AGHBASHLO, 2019), propriedades estas, que ajudam os consumidores a realizar uma boa tomada de decisão na compra dos cortes de sua preferência. Pesquisas na área sugerem que o nível de marmoreio e o WBSF influenciam tanto a probabilidade que os consumidores comprarão o produto, bem quanto o preço que estão dispostos a pagar por bifes relacionados à área do olho-de-lombo (PLATTER et al., 2005).

Os frigoríficos de pequeno-médio porte, como se dá no caso da indústria parceira desta investigação, diferentemente das grandes empresas multinacionais,

têm em sua rotina a possibilidade de coleta de fenótipos não comumente coletados em plantas de grande porte, devido à baixa escala e condições de coleta. Tal fato possibilita um melhor entendimento do produto que está sob comercialização, possibilitando a atribuição de maior valor agregado a cada corte, uma vez que lucros obtidos por elevados montantes são se fazem possíveis nesta situação. Apesar desses gastos com coleta de novos fenótipos serem onerosos aos frigoríficos de pequeno porte, o pagamento diferenciado e possibilidade de adequação quanto ao destino de cada corte permite que um melhor valor seja agregado no preço final do produto.

O presente estudo visa determinar, através de algoritmos/técnicas de Visão Computacional e linguagem de programação Python, o escore de marmoreio da área de olho de lombo, via corte transversal do músculo *Longissimus dorsi* (contra-filé), de bovinos sem composição racial previamente estabelecida.

Em uma abordagem inicial, buscou-se a adoção de algoritmos de aprendizado profundo para fins de segmentação automática da AOL. Em virtude de problemas identificados nas imagens originalmente cedidas, como falta de padronização do background, variação dos objetos de referência (régua de cores e tamanhos distintos), iluminação e brilho. A escolha das Redes Neurais Convolucionais se mostrou promissora, porém nos despertou a necessidade de obtermos melhor entendimento de como os padrões de GIM eram interpretados pelos algoritmos. Assim, os objetivos iniciais deste estudo foram desviados para uma nova abordagem onde avaliamos em primeiro momento um conjunto de algoritmos de aprendizado de máquina tradicional, aplicados a imagens que sofreram adequações quanto aos problemas acima citados. Os capítulos seguintes mostrarão etapas como os testes iniciais aplicados ao conjunto de imagens, a escolha do método para extração de atributos da AOL, e por fim, uma comparação entre os algoritmos de Machine Learning, Árvores de Decisão e Floresta Aleatória, utilizados nos processos de preditivo de classificação.

2.2 MATERIAIS E MÉTODOS

Sete mil, setecentos e quatro imagens (7.704), foram cedidas pela empresa canadense AgSights, obtidas via fotografia da superfície do músculo *Longissimus dorsi* (entre a 12^a e 13^a vértebras) de animais taurinos (*Bos taurus*) sem composição

racial previamente estabelecida. Fenótipos adicionais, relacionados à qualidade de carne, tais como área de olho de lombo (AOL, cm²), espessura de gordura subcutânea (EGS, mm), maciez, classificação para o escore de marmoreio e gordura intramuscular (GIM, escore) também foram compartilhadas pela indústria parceira. Informações complementares tais como identificação da carcaça, data do abate, sexo do animal, peso e idade ao abate também foram incorporadas ao conjunto de dados analisados.

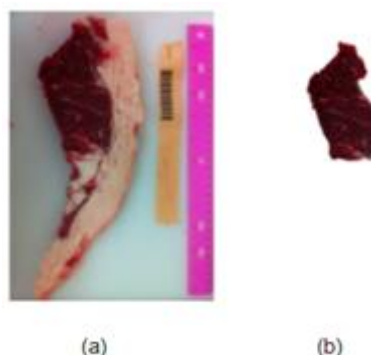
As imagens não possuíam tratamento prévio (todas em formato .JPG), resolução 800 x 598 pixels, referentes às quantidades de linhas e colunas em pixels, respectivamente. Vale ressaltar que esse banco de imagens não possuía a finalidade inicial de determinação da área de olho de lombo (AOL) automatizada ou quaisquer eventos associados com a determinação computadorizada do escore de marmoreio. Dessa forma, a partir do presente estudo, novas imagens serão e estão sendo capturadas pela indústria em questão, seguindo recomendações que possibilitem novo uso das mesmas para a resolução dos problemas aqui tratados.

De um total de 7.704 imagens iniciais, 1.807 foram descartadas por não apresentarem escore de marmoreio ou informações sobre a AOL, além de situações diversas tais como nomes duplicados em imagens distintas ou quaisquer discrepâncias adicionais caracterizadas nas respectivas colunas do banco de dados. Ao final dessa primeira seleção, permanecemos com 5897 imagens, apresentando a seguinte distribuição para as classes de marmoreio: 1471 A, 2395 AA, 2009 AAA e 22 Prime.

2.2.1 Determinação de pixel brancos e vermelhos na AOL

Em uma análise inicial, buscando-se conhecer melhor a base de dados e dar início ao processo de implementação das metodologias de VC, o uso de máscaras para determinação de pixels vermelhos (carne) e branco (GIM) contidos na AOL foi desenvolvido por meio algoritmos implementados em Python 3.6.7, fazendo-se uso das bibliotecas Numpy 1.18.1 e OpenCV 3.4.1. Tais ferramentas possuem código aberto e uma grande comunidade de usuários, fator determinante para a escolha destas. Para isso, foram criados dois subconjuntos contendo 5 imagens para cada extremo de classificação de marmoreio. Tais imagens foram então obtidas aleatoriamente dentro dos conjuntos A e Prime.

Figura 17- Determinação manual da AOL usando GIMP

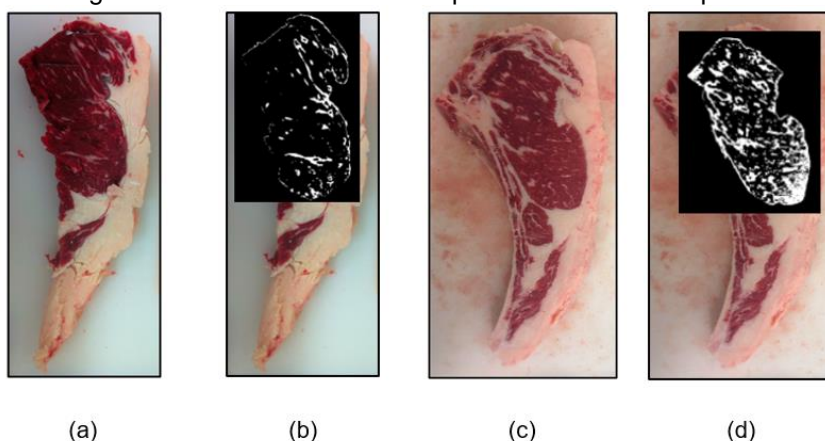


Fonte: autoria própria.

Na Figura 18 vemos um *steak* acompanhado do recorte manual de sua AOL utilizando-se o programa de manipulação de imagens do GNU, GIMP 2.10.20, salvo sem alteração de formato, 800x598 pixels. Em “a” vemos a imagem sem aplicação de nenhum filtro, e em “b” o recorte da sua respectiva AOL após inserção de canal alfa.

Via scripts Python e biblioteca de programação OpenCV (*Open Source Computer Vision*), códigos para o redimensionamento e formação de filtros de cor HSV (*Hue, Saturation and Value*) foram ajustados para a detecção de diferentes tons de cores da imagem. Tal sequência de comandos permitiu a criação de máscaras que auxiliassem na mensuração de pixels brancos e vermelhos contidos na AOL, conforme Figura 18. A definição de tais grupos vermelho e branco possuem a finalidade de determinar o percentual de gordura na AOL.

Figura 18 - Steaks A / AAA e respectivas máscaras OpenCV



Fonte: autoria própria

Para melhor entendimento sobre o uso das bibliotecas OpenCV e faixa de espectro característicos de diferentes cores, um algoritmo foi desenvolvido para

retorno dos valores mínimos e máximos da tupla RGB. Como passo posterior, foi criada uma estrutura condicional para checagem de pixels inseridos nessa faixa de min-max, gerando como retorno o valor 1 (um), cor branca, e caso contrário produziria o valor 0 (zero). Uma máscara binária é projetada informando-se quais pixels de uma determinada região de interesse pertencem à faixa pretendida. Como visto na Figura 18, o fundo da imagem possui cor branca, tonalidade esta que deveria estar representada dentro da faixa de min-max anteriormente mencionada durante o processo de desenvolvimento das máscaras. Diante de impasses relacionados com a falta de padrão do background contido nas milhares de imagens, decidimos adotar o recorte manual da AOL. Assim, trechos de cada imagem que contemplavam a AOL foram recortados manualmente para uso posterior dentro desta primeira fase investigatória.

2.2.2 Análise Piloto: nível de GIM capturado nos extremos de classificação para marmoreio (A and Prime)

A Tabela 3 ilustra os diferentes testes comparativos realizados na determinação do cálculo de gordura intramuscular (pixels brancos) contidos na AOL, entre 10 amostras divididas em dois grupos extremos de classificação (5 *steaks* do grupo A e 5 do grupo Prime). Devido ao fato de não haver dados laboratoriais sobre o nível de gordura presente em cada amostra, conduzimos uma análise preliminar para entendimento dos percentuais de GIM contidos entre os dez exemplares anteriormente mencionados. É possível observar que as amostras AOL_3, AOL_6 e AOL_7, pertencentes ao grupo A, apresentaram valores superiores ao mínimo observado pelas AOL_2 e AOL_5, representantes dos valores mínimos de GIM encontrados dentre as amostras do grupo Prime. Tal fato indica possíveis discrepâncias e a necessidade de maior exploração de tal conjunto de dados. Dentro de cada subgrupo, observamos uma enorme variação de resultados, uma vez que a diferença entre o maior e menor valor foi superior a 800% para o grupo A. De forma semelhante, no grupo Prime, essa diferença foi acima dos 400%, caracterizando também uma alta superestimação da GIM para este grupo.

Algumas hipóteses para tamanha diferença foram levantadas, entre elas: luminosidade do local, posição da câmera, resolução da imagem, excesso de brilho na carne devido ao flash automático da câmera e configurações internas do celular ao

registrar essas fotos. Em alguns exemplares, as dilacerações na carne foram interpretadas como gordura, pois a profundidade do corte afetou a quantidade de luz incidente nas regiões seccionadas. Mesmo com resultados não fidedignos, verificou-se a possibilidade de trabalhar com a biblioteca OpenCV em imagens sem tratamento prévio, mas o grande desafio persistia na automatização da seleção da AOL.

Tabela 3 - Razão de pixels brancos e não brancos contidos na AOL (5 A e 5 Prime)

Amostras	Escore	Pixels Vermelhos	Pixels Brancos	Carne contida na AOL (%)	GIM contida na AOL (%)
AOL_1	Prime	46508	5006	90,3	9,7
AOL_2	Prime	41173	2445	94,4	5,6
AOL_3	A	42031	3999	91,3	8,7
AOL_4	A	53645	508	99,1	0,9
AOL_5	Prime	50163	2999	94,4	5,6
AOL_6	A	44554	3070	93,6	6,4
AOL_7	A	45348	4261	91,4	8,6
AOL_8	Prime	41959	3306	92,7	7,3
AOL_9	A	53978	1936	96,5	3,5
AOL_10	Prime	21136	10034	67,8	32,2

Fonte: autoria própria

Para melhor entendimento sobre a acurácia dos algoritmos previamente averiguados, cartões de teste foram criados, conhecendo-se o número exato de pixels vermelhos e brancos, o que possibilitou determinar a capacidade computacional do método. A Figura 19 apresenta diferentes cartões criados utilizando-se tabelas do Microsoft Word, cujas células da tabela foram consideradas unidades de pixel. Os cartões foram compostos por 99% de pixels vermelhos e 1% de brancos.

Figura 19 - Cartões criados para testar a acurácia do algoritmo



Fonte: autoria própria.

Nas cores com tons mais escuros de vermelho, o algoritmo obteve valor resultante de pixels vermelhos próximos a 98,7%, ou seja, um erro inferior a 0,5%. Como o algoritmo foi utilizado apenas em carnes bovinas de cores reais próximas aos cartões mais escuros, o erro foi considerado insignificante e prosseguimos utilizando o mesmo algoritmo.

Após os primeiros estudos e obtenção de resultados preliminares que indicaram possíveis entraves causados pela falta de padronização das imagens, projetamos um simples protótipo mimetizando um pequeno estúdio fotográfico, de forma que nos permitisse entender/controlar a luminosidade incidente nos cortes, o que possibilitou exercer melhor controle sobre a influência dos fatores externos que estariam impactando diretamente os valores de GIM obtidos por nosso algoritmo.

Apesar de não podermos neste estudo, repetir todo o processo de obtenção de novas imagens utilizando o protótipo acima citado, resultados encorajadores foram obtidos usando este pequeno protótipo, após análise de um conjunto pequeno de amostras. De forma simplificada e observada de maneira geral, notamos que os valores médios para GIM (computados dentro da AOL e obtidos após controle de brilho para amostras contidas dentro da caixa) apresentaram valores menos inflados e mais próximos dos relatados na literatura. Vale ressaltar que tais afirmações não podem ser utilizadas ainda na prática, uma vez que investigações pormenorizadas sobre este ponto ainda devem ser conduzidas.

2.2.3 Segmentação semântica manual

A próxima etapa de nosso projeto consistiu na retomada do processo de análise das imagens anteriormente capturadas e cedidas para o uso de nossa investigação. Optamos pela verificação manual de milhares de imagens, seguida por um processo de segmentação semântica manual. Tal etapa, apesar dos enormes esforços exigidos, possibilitou a criação de um conjunto de dados que permitirá o desenvolvimento de um mecanismo autônomo de classificação futuro.

Para o desenvolvimento do estudo, baseado em Redes Neurais Convolucionais (*ConvNets*), fez-se necessário inserir manualmente o contorno das AOL através de um processo de anotação de labels, cujos pixels pertencentes ao contorno (região poligonal fechada) são armazenados no formato de coordenadas cartesianas. Diferentes *softwares* destinados à adição de *labels* foram analisados

bibliograficamente e posteriormente investigados quanto a facilidade de uso massivo. A maioria dessas ferramentas compartilharam grandes similaridades operacionais na marcação desses *labels*, porém um fator determinante para a escolha do *software* a ser utilizado foi o nível de detalhamento dos *outputs* gerados por estes, os quais deveriam possuir similaridades com o *input* necessário para o método escolhido no treinamento das *ConvNets* (E.g.: Mask R-CNN).

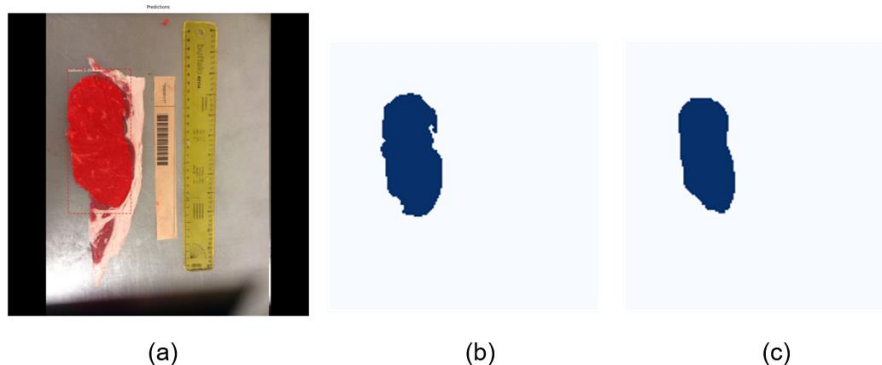
O programa VIA (VGG Image Annotator) pode ser caracterizado como um *open source project* desenvolvido pelo *Visual Geometry Group* da Universidade de Oxford. Esta ferramenta foi criada para obtenção de anotações poligonais de regiões de interesse e descrevê-las através de coordenadas dos *pixels* da referida imagem, no formato CSV (*Comma Separates Values*) ou JSON (*JavaScript Object Notation*). A extensão *.json* se assemelha ao arquivo *.xml*, porém com algumas vantagens relacionadas com otimização durante o armazenamento, segundo alguns especialistas. O VGG permite contornar regiões de interesse poligonal fechada de diversas imagens de um conjunto de dados, e retorná-las em um único arquivo de formato *json*.

Tais dados exportados pelo VGG foram obtidos de acordo com um planejamento de que serviriam posteriormente como entrada de dados para a Rede Neural Convolutiva. Com o intuito de obtermos mais clareza quanto ao uso da metodologia de segmentação semântica aplicada a dados brutos sem tratamento, avaliamos o uso do modelo Mask RCNN (JAY, 2018; ABDERRAZZAQ, 2021) aplicado a um conjunto contendo 624 imagens (447 treino / 177 teste), rotuladas manualmente via VGG, (ainda sem tratamento prévio de problemas relacionados a existência de outros objetos na figura ou despadronização do *background*). Nosso objetivo era o de segmentar apenas uma região de interesse (AOL), adotando 30 épocas de treino. Como forma de avaliação, foram avaliadas as qualidades de segmentação das 177 máscaras preditivas contidas no conjunto de testes.

A Figura 20 ilustra um dos resultados obtidos para uma das imagens escolhida aleatoriamente. Temos em “a” uma máscara vermelha, dentro de sua respectiva caixa delimitadora (*bounding box*), apresentando a probabilidade da rede neural estar correta na identificação do objeto em estudo. Na porção seguinte da figura, observamos em “b” um recorte da AOL feita manualmente pelo autor, cujas coordenadas foram armazenadas em arquivos de extensão *.json*. Na última parte da figura (c) podemos visualizar a predição da AOL determinada pela rede, idêntica à

máscara detectada na figura “a”, porém de cores sólidas e sem imagem original ao fundo.

Figura 20 - Predição da AOL após 30 épocas (1 *label*)



Fonte: autoria própria.

Apesar das caixas delimitadoras identificarem corretamente as regiões de interesse, nem todas demonstraram resultados satisfatórios em relação ao que era previsto para o banco de teste. Com base em um cenário de avaliação onde erros no contorno da AOL não eram permitidos, 26% das imagens apresentaram segmentação semântica próxima da definição manual, e em 72% das imagens foram identificadas mínimas regiões com falhas de contorno. Em 2% das amostras, duas regiões de interesse foram detectadas.

Devido ao severo critério adotado para considerar como errônea eram tais predições da AOL, várias segmentações deixaram de ser consideradas satisfatórias, indicando em uma primeira análise dos resultados que as *ConvNets* não seriam eficazes. Entretanto, considerando-se um ponto de corte mais permissivo, onde erros de até 5% na determinação da área não indiquem erros de classificação, uma queda acentuada na porcentagem de erros pode ser notada e outra possibilidade para o baixo desempenho é de que o conjunto de dados não tenha sido suficiente para o estudo. Problemas como definição do score de marmoreio são mais permissivos quanto aos erros narrados, em comparação com a pura determinação da área a ser identificada.

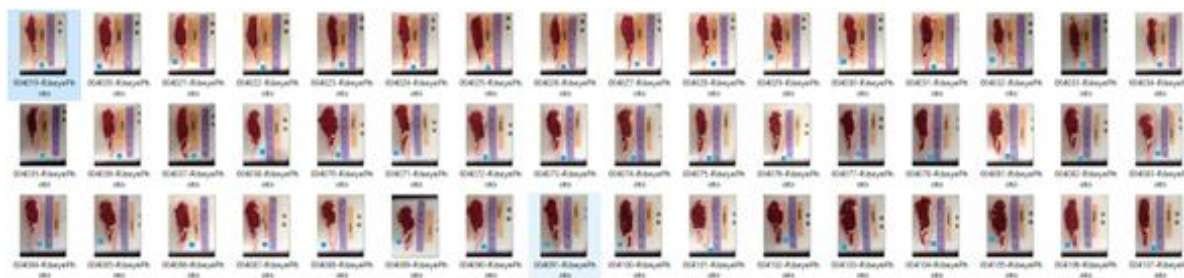
Com o intuito de desenvolvimento de uma metodologia que considerasse níveis mínimos de predição da área, optamos por manter tal critério severo para análise da área e optamos pelo foco de nossas atividades na construção de um banco de dados robusto, que posteriormente serviria para novas tentativas de segmentação. Tal

afirmação se baseia na hipótese de que tais resultados precários tenham sido motivados pelo baixo número de imagens avaliadas, ainda somado aos problemas narrados como *background* e múltiplos objetos identificados no conjunto do tipo “.JPEG”.

Em uma das tentativas para se explicar tais resultados, observou-se que uma parcela das imagens do banco de treino continha *steaks* localizados em posições distintas dentro de uma dada área de avaliação (vertical e horizontal), o que ocasionou em um intervalo de pontos indesejados (se tratando do domínio das coordenadas cartesianas). Dessa forma, uma imagem com dimensões 800x598 pixels, rotacionada em 90 graus, passava a apresentar dimensões equivalentes a 598x800 pixels. Para que não fosse necessário excluir as imagens divergentes do banco de dados, uma alternativa viável foi rotacionar todas as fotos em sua pasta de origem e salvá-las numa mesma direção. Adotou-se como padrão a posição vertical, com espessura de gordura posicionada à direita. Após concluir esse processo em ambos os conjuntos, novos rótulos foram adicionados, sem necessidade de se refazer o todo o conjunto de passos durante a anotação de cada imagem problemática.

Conforme mencionado na introdução do capítulo, antes de continuarmos os estudos aplicando-se métodos de aprendizado profunda, uma nova avaliação das ferramentas disponíveis para anotação detalhada das imagens. Com auxílio do *software* Painted 3D para Windows, quadrados de lado 3 cm, proporcionais a cada uma das régulas inseridas na imagem, foram inseridos manualmente em um conjunto de 1050 novas imagens, conforme ilustrado na Figura 21. Tais objetos foram inseridos para fins de construção de um objeto de referência de tamanho conhecido em cada uma das imagens. Este procedimento auxiliará em diversas etapas posteriores, entre elas durante o cálculo da AOL.

Figura 21 – Inserção de quadrados 9 cm² através do Painted 3D.

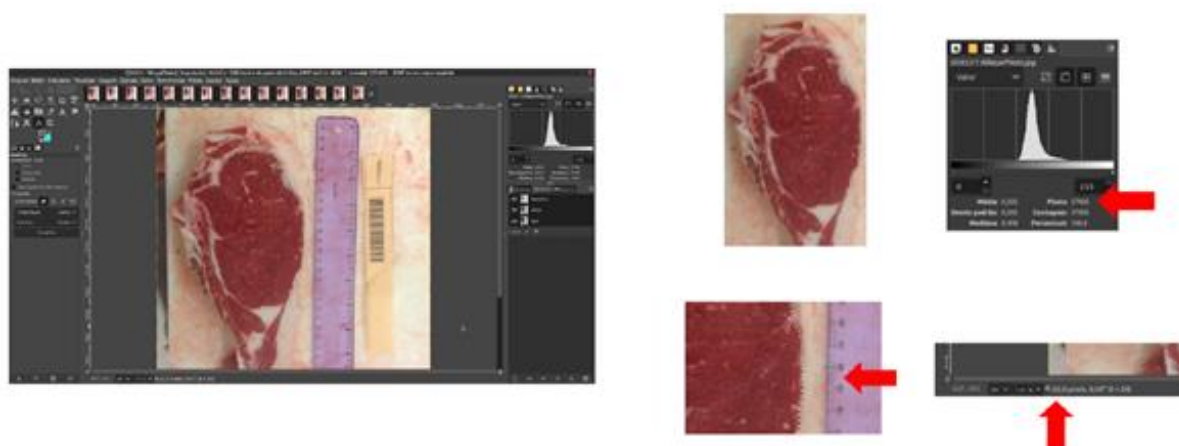


Fonte: autoria própria.

Como a proposta consiste na verificação de diferentes algoritmos de Machine Learning aplicados à classificação do escore de marmoreio e determinação da AOL, uma das necessidades foi verificar se haveria correlação entre a área de olho de lombo obtida através de um software de manipulação de imagens (GIMP 2.10.12), e o valor da área da AOL informada pela empresa canadense. Como passo preliminar, buscamos validar se a área estimada da AOL (manualmente com auxílio de um objeto de referência inserido) estaria altamente correlacionada com a área real da AOL.

Uma amostra contendo 100 imagens da classe Canada A foi escolhida de maneira aleatória, determinando-se em cada uma, a quantidade de pixels pertencentes à AOL, independentemente da posição do observador em relação à carne. Para esse cálculo foi criada uma função diretamente proporcional entre o número (pixel/mm) contidos na régua ao lado do *steak* e à quantidade de pixels pertencentes a AOL. Sabendo-se ambos os valores, a área real estimada (mm^2) da AOL foi determinada da seguinte maneira: primeiramente, determinou-se a quantidade de pixels por milímetro de régua; elevou-se esse número ao quadrado, encontrando-se assim a quantidade de pixels/ mm^2 ; por fim dividiu-se a quantidade total de pixels contidos na AOL pelos pixels/ mm^2 obtendo a área real estimada (mm^2), conforme Figura 22.

Figura 22 - Determinação da área real da AOL para 100 amostra



Fonte: autoria própria.

Para realizar o cálculo de correlação entre a área fornecida pela indústria e área estimada pelo autor, utilizou-se o *software* Minitab 18 durante os cálculos estatísticos. As 100 amostras apresentaram alta correlação positiva, com coeficiente de correlação de Pearson igual a 0,980 e valor-p praticamente nulo. Logo, a hipótese

de que as áreas fornecidas pela indústria foram devidamente coletadas, evitando assim, a contagem de todas as áreas de interesse manualmente foi aceita e estas serão usadas de parâmetro quando o modelo final de determinação da AOL for necessário.

2.2.4 Padronização do background e inserção do objeto de referência

Na busca por parcerias com outros pesquisadores, foi sugerido modificar o *background* das imagens já recebidas, simulando que as futuras imagens advindas do frigorífico sejam padronizadas, sem necessidade da dependência de uma régua ao lado dos *steaks*, e para essas possam deixar de existir, um objeto de referência com tamanho fixo deverá ser colocado em determinado local da cena. Optamos pela escolha de um fundo que se destacasse do objeto alvo (*Steak*), ao mesmo tempo que também fosse facilmente diferenciada do objeto de referência (Quadrado de 9 cm²).

Tomando-se um *steak* majoritariamente composto por cores semelhantes ao vermelho e branco, (representativas da carne e GIM, respectivamente) e considerando-os numa escala de cinza (0 a 255 pixels), podemos observar as seguintes faixas de valores (em pixels) para os componentes da carne, gordura, fundo verde e objeto de referência: i) 40-120; ii) 120-255; iii) constante e igual a 150; iv) constante e igual a 0. No caso das cores constantes, o processo de seleção de tais elementos torna-se facilitado, o que possibilita a melhor distinção entre o objeto alvo e seu background. Dessa forma, evidencia-se os contornos de oclusão.

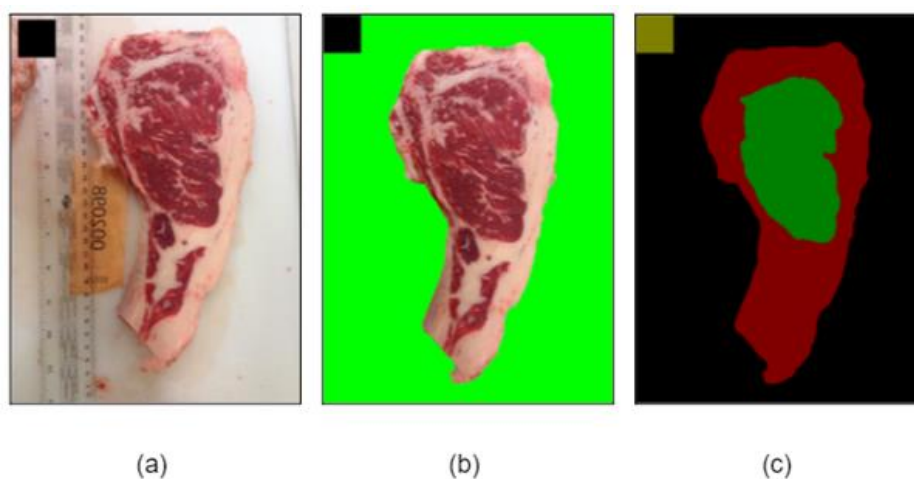
A escolha do *background* composto pela cor verde foi baseada nos resultados reportados pela indústria cinematográfica, onde existem estúdios para efeitos especiais baseados na utilização do tecido *chroma key*. Dessa maneira, as 1050 imagens escolhidas na etapa anterior, foram descartadas e refeitas selecionando toda a carcaça separando-a do *background*, pois isso permitiria uma possibilidade futura em criar outros processos para identificação de sub-regiões de interesse pertencentes ao *steak*, descartando interpretações equivocadas ou más definições de bordas, uma vez que algumas tábuas de corte continham sangue proveniente das próprias peças.

Em seu repositório GitHub, Kentaro (2016) apresenta uma aplicação do software Labelme para geração de máscaras de cores sólidas em distintas regiões rotuladas a partir dos arquivos JSON, algo bem parecido com os resultados da segmentação apresentada anteriormente. Semelhante ao proposto por Kentaro

(2016), desenvolvemos um código, inicialmente implementada para uso em outras áreas externas à produção animal, para o presente estudo.

Conforme Figura 23, três cenários contendo criação de máscaras são apresentados: (a): imagem original após inserção do quadrado de 9 cm² sobreposto a um trecho da régua; (b): *steak* e objeto de referência isolado do background original, posteriormente inseridos em um novo fundo verde; finalmente em (c): máscaras de cores sólidas geradas a partir dos labels via arquivos JSON (*steak*, AOL e objeto de referência).

Figura 23 - Máscara de cores sólidas criada a partir do script (Kentaro)



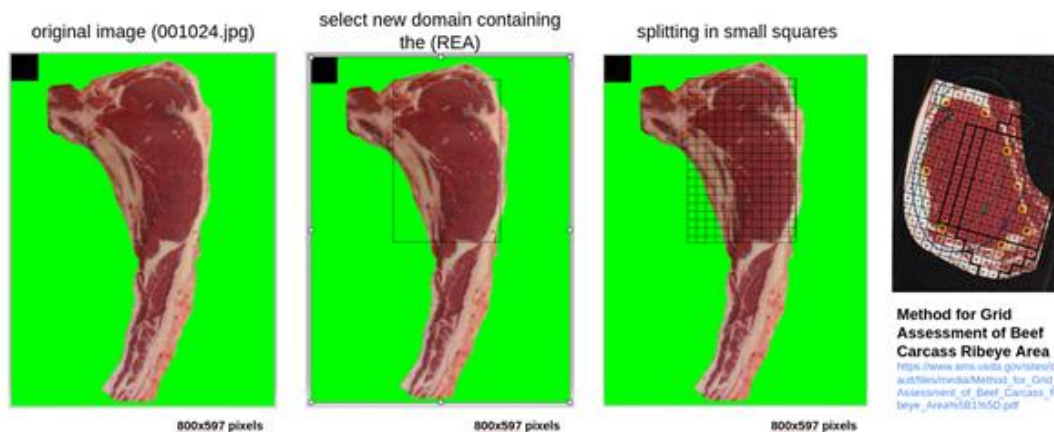
Fonte: autoria própria.

Após padronizar as imagens em relação ao background, e gerar suas respectivas máscaras de cores sólidas, um script *Python* foi confeccionado de forma a essas duas informações como entrada de dados. (via comandos pertencentes às bibliotecas *Numpy* e *OpenCV*). Para fins de classificação, apenas os pixels pertencentes a AOL permaneceram distintos de zero, enquanto os demais componentes da máscara de cores sólidas se tornam zero. Isso é possível devido a existência de cores sólidas, uma vez que seus pixels em escala de cinza são conhecidos e distintos dentro de um intervalo fechado. Sendo assim, se o objetivo for determinar a quantidade de pixels contida no quadrado, pode-se criar uma linha de comando que conte apenas pixels pertencentes ao intervalo de cores amarelas, por exemplo.

2.2.5 Extração de atributos via LBP

A Figura 24 ilustra o processo de transformação da imagem, desde sua composição inicial já caracterizada com a padronização do background, até a obtenção das regiões de interesse (AOL sendo avaliada como foco principal para a determinação do escore de marmoreio). Vale ressaltar que a imagem apresentando distintas cores sólidas, passou por um processo de transformação em novas matrizes de incidência (0/1), enquanto background (fundo verde) foi convertido em uma matriz contendo escalas de cinza. O algoritmo percorre simultaneamente as imagens e realiza uma correspondência cartesiana entre seus pixels, sendo que toda vez que se depara com um zero, automaticamente zera o elemento da imagem paralela, deixando diferente de zero apenas a região de interesse, a AOL. Posteriormente, divide-se a matriz resultante em pequenos quadrados deslizantes contendo apenas elementos não nulos. Como mencionado no levantamento bibliográfico, a USDA realiza um método classificatório (manual) para determinação dos escores de marmoreio e obtenção da AOL em carcaças bovinas, também baseado na construção de grades ao longo da imagem.

Figura 24 - Determinação das regiões de interesse (ROI)

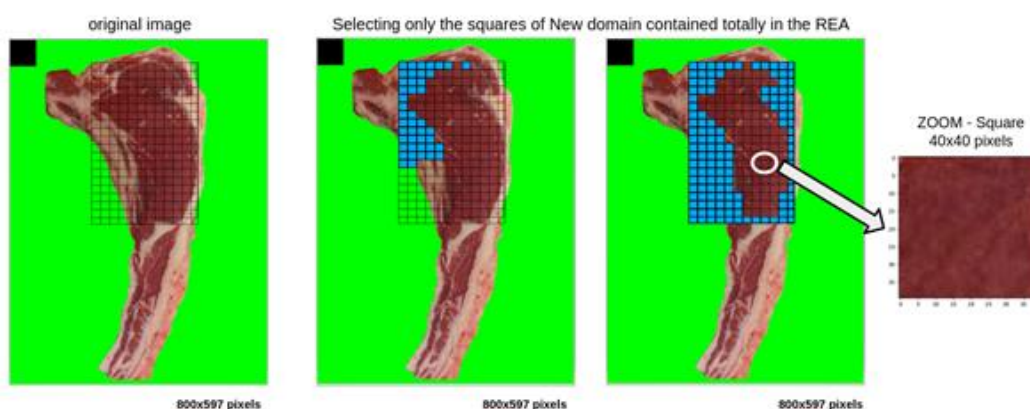


Fonte: autoria própria

Os quadrados não contidos totalmente na AOL foram representados na Figura 25 pela cor azul. A ordem das matrizes quadradas representantes desses quadrados não possui um tamanho ideal conhecido, sendo assim, nossa proposta foi a de se investigar diferentes tamanhos de janelas a fim de confrontar melhor relação entre a extração de atributos para os diferentes métodos de Machine Learning estudados. A definição do tamanho ideal de janelas se dará por meio dos valores de acurácia durante a classificação.

Uma vez detectada a região da AOL, iniciamos o processo de extração de atributos dentro da área de interesse. Tais atributos serão utilizados como entrada de dados durante o processo de treinamento supervisionado pelos algoritmos de aprendizado de máquina. Diferentes alternativas metodológicas para extração de *features* (atributos) foram investigadas.

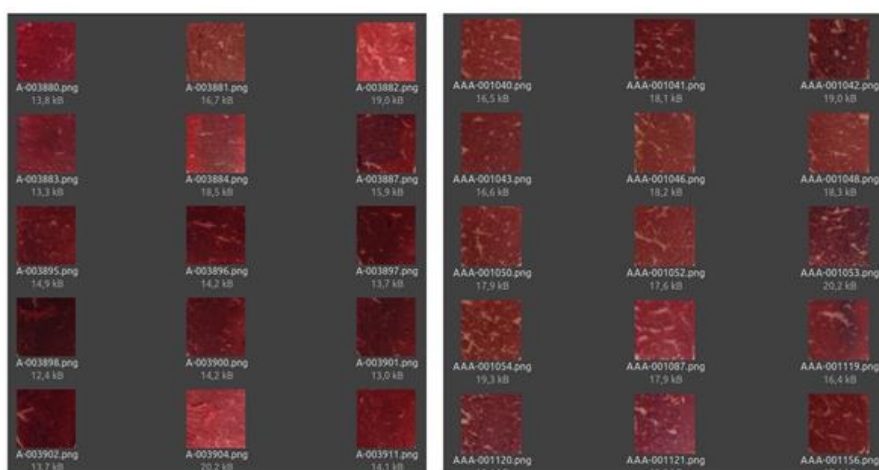
Figura 25 - Divisão da AOL em quadrados 40 x 40 pixels



Fonte: autoria própria.

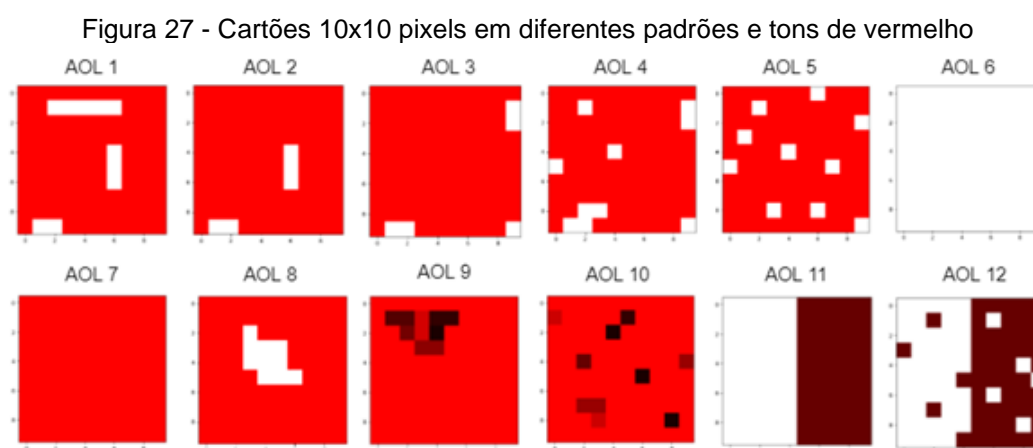
Descreveremos a seguir os passos utilizados para escolha de um conjunto de dados inicial para comparação das metodologias e posterior uso dos métodos de entropia e LBP. Trinta (30) amostras foram selecionadas, 15 de cada grupo extremo, A e AAA, objetivando separar diferentes padrões reconhecidos pelas *features* dentro da AOL. Algumas amostras contidas na Figura 26 possuíam cores e brilhos diferentes, sendo também inseridas nesta etapa exploratória.

Figura 26 - Padrões de GIM para 30 amostras de janelas 40 x 40 pixels



Fonte: autoria própria.

A primeira tentativa de extração de *features* baseou-se no cálculo de entropia entre os pixels vermelho e branco, onde analisamos a possibilidade de se determinar a mudança entre pontos, traços ou estrias de gordura contidas nessas pequenas janelas. Conforme procedimento anterior, cartões foram criados contendo 100 pixels cada, com diferentes traços de pixels brancos, diferentes tons de vermelho, e quantidade conhecida de pixels de cada uma dessas cores (Figura 27). Uma vez que a entropia utiliza em seu cálculo probabilidades entre dois elementos, classes binárias foram criadas para representação dos agrupamentos de pixels vermelho e brancos.



Fonte: autoria própria.

É possível verificar na Tabela 4 que as figuras “AOL 8” e “AOL 10” retornaram valores praticamente iguais, não ocorrendo a distinção correta para mudanças de tons entre os pixels vizinhos, em situações onde as diferenças eram drásticas ou não. Dessa forma, para situações reais onde fibras da carne podem estar mais destacadas pela cor dos sulcos ao invés de estrias mais proeminentes, o cálculo da entropia teria praticamente o mesmo resultado. As imagens AOL 1, 4, 5 e 8 apresentaram a mesma entropia, independentemente da posição em que os traços estavam, algo que impossibilitou a ideia inicial de usar somente o cálculo de entropia, já que ele não conseguiria detectar pontos, traços ou estrias dispersas ao longo dessas pequenas regiões da AOL. Nesses casos, o cálculo de entropia considerou apenas o percentual de pixels brancos ou vermelhos, sem diferenciar sua maneira de organização, homogênea ou heterogênea, como observado nas AOL 11 e 12.

Posteriormente às tentativas de entropia, após diversa busca na literatura por algoritmos que fornecessem destaque para alterações locais da imagem, optamos por investigar o uso do método Local Binary Pattern (LBP), motivado pelo seu uso de

sucesso em outras áreas de atuação que também necessitavam de melhor entendimento sobre regiões específicas da imagem. Análises descritivas foram conduzidas de forma a entendermos como os atributos visuais eram distribuídos dentro das diferentes categorias de marmoreio (A, AA, AAA e Prime). Paralelamente, iniciamos o teste de tal algoritmo comparando-se diferentes opções de dimensionalidade relacionadas com as janelas deslizantes, utilizadas pelo LBP.

Tabela 4 – Cálculo da entropia em AOL 10x10 pixels

Imagem	Pixels em tons de vermelho	Pixels em tons de branco	Entropia
AOL 1	90	10	0,1368
AOL 2	95	5	0,0703
AOL 3	95	5	0,0703
AOL 4	90	10	0,1368
AOL 5	90	10	0,1368
AOL 6	0	100	0
AOL 7	100	0	0
AOL 8	90	10	0,1368
AOL 9	100	0	0,0664
AOL 10	100	0	0,1517
AOL 11	50	50	0,5
AOL 12	50	50	0,5

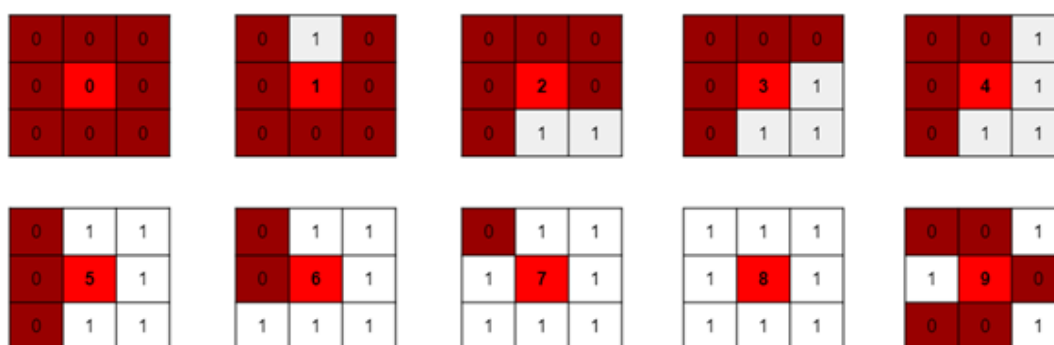
Fonte: autoria própria.

Tomando as AOL de 1 a 10, contidas na Figura 28, aplicamos a métrica LBP (8,1) a cada um dos 100 pixels pertencentes a imagem e na sequência um histograma de frequência foi construído para que pudéssemos observar alguma regularidade nos padrões estabelecidos pelo algoritmo. Para isso, foi utilizado parte do código disponibilizado por Rosebrock (2015), pesquisador que propõe a utilização do *Local Binary Pattern* (LBP) via biblioteca *Skimage*. Ao invés de se obter as saídas do algoritmo nas bases decimais, utilizamos apenas o valor central do pixel anterior a essa transformação.

Para determinar padrões locais uniformes (LBP), é necessário o entendimento sobre a escolha do número de vizinhos e o raio (em pixels), parâmetros exigidos pelo método. Neste método existem g_p elementos que formam o perímetro do quadrado de centro g_c da matriz de ordem quadrada. No geral, se um pixel pertencente à vizinhança for maior ou igual ao pixel central, substituiremos seu valor por 1. Caso seja menor, a substituição é feita por 0. Por fim, substituímos o pixel central pelo somatório dos pesos 0 e 1, representantes dos vizinhos equidistantes do centro, determinados pelo raio.

A Figura 28 apresenta nove (matrizes de 0 a 8) padrões uniformes para a métrica (8,1), lembrando que eles ocorrem apenas quando há no máximo duas ligações 0/1 entre os elementos pertencentes à vizinhança. Nos nove (matrizes de 0 a 8) casos em que o padrão uniforme ocorre, os pixels pertencentes à vizinhança tiveram seus valores 0 e 1 somados e atribuídos ao pixel central. As exceções ocorrem nos padrões não uniformes, os quais terão valor uma unidade a mais que a quantidade de vizinhos, ou seja, $P + 1$, representado pela matriz de centro 9, a qual possui 6 ligações 0/1, ultrapassando assim, as duas ligações máximas permitidas.

Figura 28 - Determinando padrões uniformes possíveis para o LBP(8,1)



Fonte: autoria própria

Uma vez entendido como o LBP (8,1) irá agir nas AOL da Figura 28, aplicamos o algoritmo e armazenamos as informações em um histograma de frequência, apresentado na Tabela 5.

Tabela 5 – Histograma de frequência para AOL 10x10 pixels e LBP(8,1)

Classes	AOL 1	AOL 2	AOL 3	AOL 4	AOL 5	AOL 6	AOL 7	AOL 8	AOL 9	AOL 10
0	0	0	1	4	10	0	0	0	0	0
1	6	4	4	4	0	0	0	2	1	1
2	0	0	0	0	0	0	0	0	0	0
3	4	4	3	3	3	4	4	7	6	8
4	0	0	0	0	0	0	0	0	0	0
5	30	30	28	27	29	32	32	34	28	38
6	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	1	8	17
8	56	61	64	60	58	64	64	54	47	22
9	4	1	0	2	0	0	0	2	10	13

Fonte: autoria própria.

Em relação aos histogramas da Tabela 5, um possível problema pode ser avistado nas colunas 6 e 7, pois apesar de possuírem características distintas,

totalidade vermelha e branca, os pixels vizinhos a um pixel pertencente a essas imagens possuem a mesma cor, logo o método atribuirá aos 4 vértices do quadrado a classe 3, aos 32 pixels pertencentes aos lados classe 5 e aos demais 64 pixels pertencentes ao interior classe 8. Esses dois exemplos são praticamente improváveis de acontecer, pois na prática, uma janela desse tamanho não teria somente pixels da mesma cor se tratando da carne. Logo, desconsiderando esses dois extremos, as demais imagens não geraram histogramas idênticos e pode-se observar um dado satisfatório entre os histogramas das AOL 5 e 8, demonstrando, portanto, que tal método permite distinguir a GIM acumulada em um trecho (AOL 8), assim como para os casos onde a GIM encontra-se espalhada de maneira homogênea (AOL 5). Por estes motivos, o LBP foi investigado mais a fundo e escolhido para seleção de atributos.

Para o desenvolvimento desse projeto, independentemente da ordem das janelas quadradas, o método LBP percorrerá cada um de seus elementos e substituirá o pixel central pelo somatório dos vizinhos, ou seja, não haverá transformação em base decimal, como os demais pesquisadores o fazem. Usaremos um histograma de frequências começando em 0 até $P+1$. Um máximo de 10 classes é esperado para o LBP(8,1), 18 para o LBP(16,2) e 26 considerando-se os parâmetros LBP(24,3). A AOL foi subsequentemente dividida submatrizes de tamanhos distintos e o LBP aplicado em cada uma dessas matrizes.

2.2.6 Algoritmos de Machine Learning (Decision Tree e Random Forest)

Após finalização do processo de *labels* e anotação de um total 1050 imagens, 200 amostras de cada um dos grupos de marmoreio A, AA e AAA foram escolhidas (assumindo-se que baixos e altos valores de GIM fossem representados para os respectivos extremos de classificação) para início dos testes envolvendo a extração de atributos via LBP, posteriormente treinados nos diferentes algoritmos de machine learning: *Decision Tree* (DT) e *Random Forest* (RF). Para tal etapa, nenhuma alteração de hiperparâmetros foi conduzida, a não ser os mesmos contidos na biblioteca *Sklearn*.

Diferentes configurações para o LBP foram testadas utilizando-se de três tamanhos distintos de submatrizes de ordem quadrada para fins de se percorrer AOL adjacientemente (sem *overlap*). Baseando-se na mesma estratégia, um aumento

gradativo no tamanho amostral foi aplicado aos mesmos cenários anteriormente descritos. Uma sequência de testes foi pensada da seguinte forma. Entre as 1050 imagens disponíveis até aquele momento, separar 200 em cada grupo de escores que melhor representassem cada uma das classificações, ou seja, pouca GIM, mediana GIM e abundância GIM dentro dos grupos A, AA e AAA, e depois ir aumentando as amostras gradualmente e verificando as acurácias.

Na literatura, não há um tamanho definido a respeito das matrizes quadradas a serem usadas durante o método LBP. Alguns autores preferem reduzir a imagem inicial em pequenas matrizes, de tamanho 7×7 , para posterior aplicação do método. Em nossos testes iniciais, tal procedimento se mostrou ineficaz. Por este motivo, decidimos manter a imagem original e percorrê-la em pequenas janelas de formato quadrado. Como apresentado na Tabela 6, diversos tamanhos de janelas foram investigados ao se aplicar distintas configurações para o algoritmo LBP. A alteração de tais parâmetros promoveu resultados de acurácia muito próximos para vários cenários testados, apresentando diferenças consideráveis quanto ao tempo computacional exigido para a varredura de cada janela alvo.

Tabela 6 - Diferentes métricas LBP e matrizes quadradas de ordens distintas (200 A e 200 AAA)

Método	Acurácia Decision Tree (%)	Acurácia Random Forest (%)
LBP(8,1) – ordem 3	86,66	90,83
LBP(8,1) – ordem 4	88,33	92,5
LBP(8,1) – ordem 5	84,16	89,16
LBP(8,1) – ordem 6	83,33	90,83
LBP(8,1) – ordem 7	82,5	90
LBP(8,1) – ordem 8	83,33	90
LBP(8,1) – ordem 9	83,33	90,83
LBP(8,1) – ordem 10	85	90,83
LBP(16,2) – ordem 3	83,33	90,83
LBP(16,2) – ordem 4	80	88,33
LBP(16,2) – ordem 5	83,33	89,16
LBP(16,2) – ordem 6	80	91,66
LBP(16,2) – ordem 7	78,33	87,5
LBP(16,2) – ordem 8	78,33	88,33
LBP(16,2) – ordem 9	82,5	91,66
LBP(16,2) – ordem 10	84,16	90
LBP(24,3) – ordem 3	70,83	71,66
LBP(24,3) – ordem 4	80,83	87,5
LBP(24,3) – ordem 5	80,83	86,66
LBP(24,3) – ordem 6	85	87,5
LBP(24,3) – ordem 7	74,16	88,33
LBP(24,3) – ordem 8	77,5	89,16
LBP(24,3) – ordem 9	84,16	89,16
LBP(24,3) – ordem 10	79,16	90

Fonte: autoria própria.

A tabela 7 mostra os melhores resultados para 400 imagens, 200 representantes de cada um dos extremos de classificação, pouca GIM ou abundância de GIM, divididas em dois conjuntos, treino e teste, na proporção de 70 para 30, respectivamente. Utilizando o algoritmo de Aprendizado de Máquina, *Decision Tree* (DT) e Random Forest (RF), sem alteração de hiperparâmetros, a não ser os mesmos contidos na biblioteca *Sklearn*, diferentes métricas para o LBP e três tamanhos distintos de submatrizes de ordem quadrada para percorrerem a AOL adjacientemente, sem *overlap*.

Tabela 7 - Aplicação de diferentes métricas LBP e matrizes quadradas (200A e 200AAA)

Método	Acurácia	
	Decision Tree (%)	Random Forest (%)
LBP(8, 1) - ordem 20	83,33	90,83
LBP(8, 1) - ordem 30	86,67	90
LBP(8, 1) - ordem 40	90,83	91,67
LBP(16, 2) - ordem 20	80	88,33
LBP(16, 2) - ordem 30	84,17	88,33
LBP(16, 2) - ordem 40	79,16	88,33
LBP(24, 3) - ordem 20	82,5	87,5
LBP(24, 3) - ordem 30	80,83	81,67
LBP(24, 3) - ordem 40	75	82,5

Fonte: autoria própria.

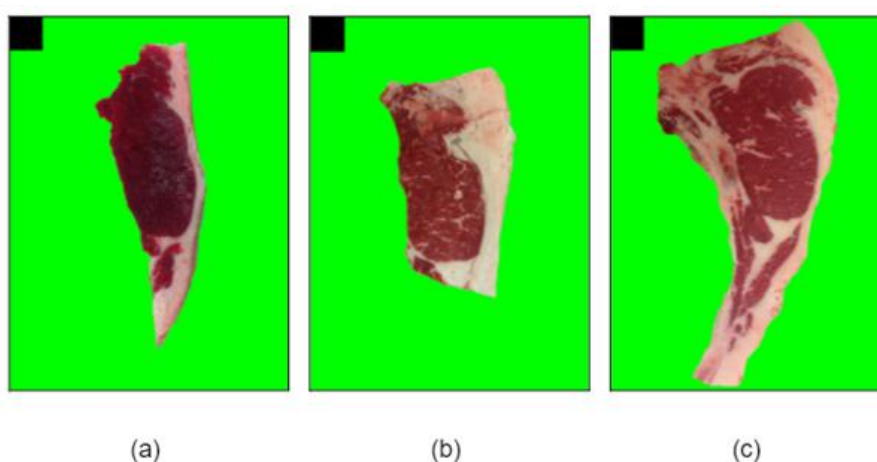
Os melhores resultados classificatórios foram alcançados considerando-se um total de 400 imagens (200 amostras de cada um dos extremos de classificação foram representadas deste conjunto de dados). Os conjuntos de treinamento e teste foram formados com base na proporção 70:30, respectivamente. Para este conjunto de 400 imagens, 200 A e 200 AAA, obtivemos acurácia máxima de 90,83% via algoritmo DT e 91,67% quando o método RF foi utilizado. Em ambos os cenários foram adotados os parâmetros LBP (8,1) e matrizes quadradas de ordem 40. De maneira prática, podemos traduzir que de um total de 120 imagens avaliadas no conjunto de teste, o algoritmo DT fez a predição correta de 108 delas, classificando erroneamente 12 imagens. Seguindo o mesmo raciocínio, o algoritmo RF classificou corretamente 110 das 120 imagens testadas. Uma vez que a métrica (8, 1) sobressaiu entre os nove exemplos anteriores, os testes subsequentes foram feitos adotando-se os mesmos parâmetros, modificando-se apenas a ordem das matrizes e expandindo o total de imagens avaliadas para 600 e 750 no total (onde 200 ou 250 imagens de cada grupo foram amostradas na Tabela 8).

Tabela 8 - Aplicação do LBP(8,1) em matrizes quadradas (200/250A , 200/250AA e 200/250AAA)

Método	(200A / 200 AA / 200 AAA)		(250A / 250 AA / 250 AAA)	
	Acurácia DT (%)	Acurácia RF (%)	Acurácia DT (%)	Acurácia RF (%)
LBP(8, 1) - ordem 20	53,88	59,44	49,77	49,77
LBP(8, 1) - ordem 30	57,77	56,66	48	49,77
LBP(8, 1) - ordem 40	53,88	58,88	46,66	49,77

Fonte: autoria própria

Após inserção das classes medianas de GIM (AA), uma visível redução na acurácia foi observada para ambos os métodos, se comparados ao teste anterior contendo apenas os grupos A e AAA: 57,77% (DT) e 59,44(RF) para (N = 600) e 49,77% (DT/RF) para (N = 750). Os melhores resultados ocorreram em janelas quadradas de ordens distintas do teste com duas classes, antes o lado do quadrado era 40 e agora 30 e 20. Está evidente queda na acurácia, à medida que um terceiro grupo foi inserido, talvez esteja relacionada ao fato de tais imagens possuírem padrões que se confundem, ora com o grupo A, ora com grupo AAA. Tais valores podem ser facilmente identificados via histogramas de frequência (Gráfico não apresentado neste documento). Adicionalmente, uma hipótese adicional está relacionada à subjetividade da avaliação humana. Outro possível problema está relacionado com imagens onde a AOL se encontrava incompleta, como pode ser visto na Figura 29.

Figura 29 - Diferentes padrões de *steaks* contidos no banco de dados

Fonte: autoria própria.

Para que um treinamento de Machine Learning seja eficaz, deve-se garantir conjuntos balanceados, tanto no treinamento quanto validação, ou seja, devemos possuir o mesmo número de elementos nos três conjuntos (A, AA e AAA). De um total

de 5.897 imagens anteriormente disponíveis, apenas 2.446 destas se enquadraram nos padrões estabelecidos no nosso estudo (582A, 955 AA e 909 AAA). Devido ao desbalanceamento de tais classes, os testes finais foram conduzidos com base em 1.746 imagens, onde cada grupo de marmoreio contemplou um total de 582 imagens.

Em busca de se avaliar o impacto do desbalanceamento das amostras, mas contemplando a totalidade das amostras anotadas, prosseguimos com nova análise onde o treinamento englobou todo o conjunto de dados disponível. Novos resultados para tais comparações são apresentados na Tabela 9. Os mesmos foram conduzidos mantendo-se as mesmas métricas do LBP, diferentes ordens para as matrizes quadradas, mesmos algoritmos de Machine Learning e proporção 70:30 para a formação dos conjuntos de treinamento e teste (sem alteração de hiperparâmetros).

Tabela 9 - Aplicação de diferentes métricas LBP e matrizes quadradas (582A, 582AA e 582 AAA)

Método	Acurácia	
	Decision Tree (%)	Random Forest (%)
LBP(8, 1) - ordem 20	48,28	58,77
LBP(8, 1) - ordem 30	45,99	60,68
LBP(8, 1) - ordem 40	47,32	57,44
LBP(16, 2) - ordem 20	51,71	63,16
LBP(16, 2) - ordem 30	50,38	62,02
LBP(16, 2) - ordem 40	50,38	61,45
LBP(24, 3) - ordem 20	50,38	61,45
LBP(24, 3) - ordem 30	50,95	63,54
LBP(24, 3) - ordem 40	49,23	61,64

Fonte: autoria própria.

De forma semelhante ao apresentado na Tabela 8, os melhores resultados da Tabela 9 ficaram próximos a 60%, porém desta vez, os parâmetros LBP(8,1) não demonstraram ter o melhor desempenho, como relatado na Tabela 6. O fato de uma métrica ser melhor que outra em determinados testes justifica novos experimentos e devida atenção.

Um novo estudo foi realizado utilizando-se apenas de amostras retiradas das classes extremas, a fim de reenfatar o uso de LBP (8,1) aplicado a um conjunto ampliado de imagens (agora com 1164 imagens (582A e 582 AAA)), mantendo-se os mesmos critérios anteriores (Tabela 10). De maneira inesperada, a métrica (8, 1) não foi a melhor para os dois grupos extremos. Quando comparadas aos melhores resultados obtidos para os cenários envolvendo apenas 400 imagens (200 A e 200 AAA), a melhor métrica encontrada fez uso de LBP(16, 2). Os resultados de acurácia via método *Decision Tree* caíram aproximadamente 15%, quase duas vezes superior

aos obtidos via *Random Forest* (queda de 7%). Um possível aumento de acurácia era esperado após expandirmos o N amostral durante o processo de treinamento, porém tal fato não foi verificado. Novas investigações devem ser conduzidas para testar se imagens pertencentes aos grupos extremos também serão classificadas de maneira equivocada, de tal maneira que um exemplar A tenha recebido classificação AAA, ou vice-versa.

Tabela 10 - Aplicação de diferentes métricas LBP e matrizes quadradas (582A e 582 AAA)

Método	Acurácia Decision Tree (%)	Acurácia Random Forest (%)
LBP(8, 1) - ordem 20	71,14	83,42
LBP(8, 1) - ordem 30	69,71	80
LBP(8, 1) - ordem 40	71,14	79,42
LBP(16, 2) - ordem 20	77,14	85,42
LBP(16, 2) - ordem 30	73,71	85,42
LBP(16, 2) - ordem 40	77,42	85,14
LBP(24, 3) - ordem 20	72,85	85,14
LBP(24, 3) - ordem 30	72,57	84
LBP(24, 3) - ordem 40	76,28	84

Fonte: autoria própria

Analisando o percentual de acertos alcançados pelo método RF (85,42%) o classificador acertou o escore de 299 imagens de um total de 350. Resultados semelhantes foram encontrados por Barbon et al. (2017), após aplicação de um método baseado no algoritmo k-NN (81,59%), em um total de 45 amostras extraídas entre última e penúltima costela do músculo *Longissimus thoracis*, de bovinos da raça Nelore. Pang et al (2014) obtiveram acurácia máxima baseada no algoritmo K-fold cross validation (86,05%) na predição de escore de marmoreio via classificador SVM, combinando método de segmentação de imagem (Otsu), para determinação de GIM na AOL e criação de 7 *features*, num total de 123 imagens.

Verificando se as suposições feitas anteriormente estavam corretas, porém com um banco de dados ampliado, estatísticas descritas são mostradas na Tabela 11, colaborando no entendimento de alguns questionamentos levantados a respeito da influência do não tratamento das imagens coletadas. Potenciais entraves envolvem aspectos sobre a luminosidade incidente sobre os cortes, cujo excesso de brilho influencia na formação da tupla RGB, retomando coordenadas não representativas da realidade, ou seja, aquela vista aos olhos do avaliador humano. Isso pode ser observado através dos valores máximos em cada uma das classes, os quais se distanciam significativamente da média aritmética, pois ao considerarmos os limites

críticos superiores de cada classe, mais de 10% das amostras podem ser consideradas outliers (5,49% (A), 2,23% (AA) e 3,78% (AAA)).

Tabela 11 - Estatísticas descritivas para 1746 amostras (582A, 582AA e 582 AAA)

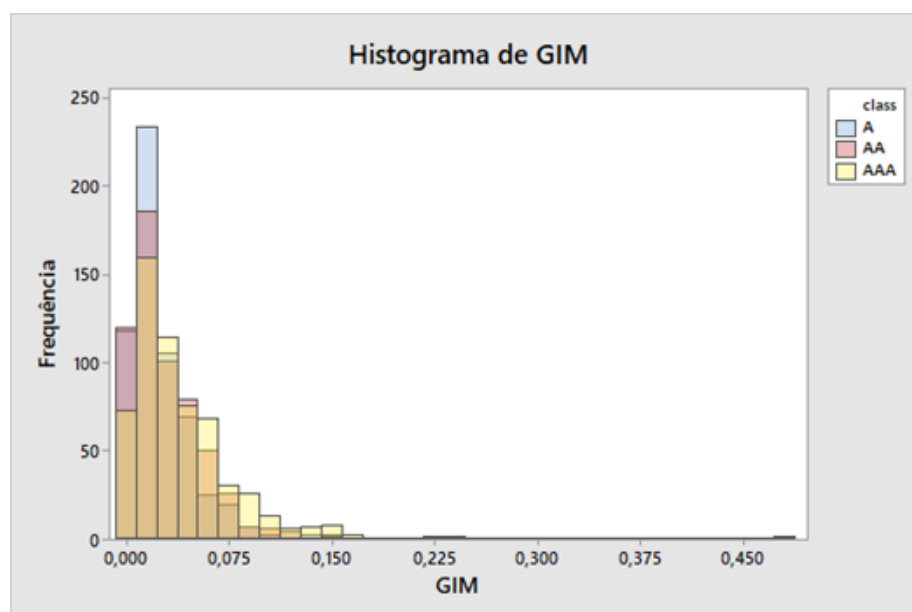
Classes	N	Médias	Desvio Padrão	Mínimo	Q1	Mediana	Q3	Máximo
A	582	0,0243	0,0219	0,0000	0,0092	0,0174	0,0326	0,1469
AA	582	0,0296	0,0321	0,0005	0,0092	0,0208	0,0433	0,4869
AAA	582	0,0390	0,0337	0,0002	0,0146	0,0290	0,0552	0,2371

Fonte: autoria própria.

Conforme Figura 30, se olharmos a sobreposição dos histogramas de frequência da GIM para cada uma das classes (A, AA e AAA), todos apresentam assimetria positiva a direita, além de grande semelhança gráfica, o que caracteriza como possível justificativa para a dificuldade obtida durante as classificações dos elementos de cada grupo utilizando-se apenas a proporção de GIM contida na AOL.

Para um melhor direcionamento sobre possíveis erros atribuídos ao processo de classificação, faz-se necessário nova definição/avaliação das imagens, se possível, conduzidas por um conjunto amplo de avaliadores independentes. Em um cenário experimental ideal, vislumbra-se também a submissão de novas amostras para verificações laboratoriais capazes de se obter mensurações pontuais sobre o teor de gordura contido em cada amostra. Novos testes são planejados como passos futuros de análise, combinando dessa vez métodos de aprendizado profundo.

Figura 30 - Histograma de frequência do percentual de GIM presente em 1746 amostras por classes



Fonte: autoria própria.

3 CONCLUSÃO

Em conclusão, os resultados deste trabalho apresentam consonância com as técnicas de visão computacional e análise de imagens aplicadas por outros autores dentro da mesma área de atuação. Por meio de métodos simplificados (Contagem de pixels), foi possível prever a quantidade de gordura intramuscular (GIM) contida em diferentes amostras do contrafilé. Em relação à máscara preditiva gerada pelo método de segmentação, problemas na determinação correta das bordas foram detectados (falhas em relação a separação das bordas envolvendo a área de olho de lombo (AOL)), exigindo-se que novos estudos sejam conduzidos especificamente para o pleno entendimento deste tópico.

Os estudos iniciais utilizando o método para extração de atributos LBP, mostraram resultados bem expressivos para classes extremas de marmoreio. Ainda faz-se necessário maior entendimento biológico e visual sobre os padrões encontrados em cada grupo de amostras. Métodos de clusterização e classificação não supervisionados estão sendo delineados a fim de se identificar o impacto da subjetividade humana durante as classificações adotadas.

REFERÊNCIAS

AASLYNG, M. D. Quality indicators for raw meat. **Meat Processing**, United Kingdom, p. 157–174, 2002.

ABDERRAZZAQ, S. **Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow**. 2021. Disponível em: <<https://github.com/sabderra>>. Acesso em: 3 dez. 2021.

AHONEN, T.; HADID, A.; PIETIKÄINEN, M. Face description with local binary patterns: application to face recognition. **IEEE Trans Pattern Anal Mach Intell**, [S.l.], v. 28, n. 12, p. 2037-2041, 2006.

AWAD, M.; KHANNA, R. **Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers**. New York: Apress, 2015.

BALLARD, D. H.; BROWN, C. M. **Computer Vision**. New Jersey: Prentice-Hall, 1982.

BALLARD, W. **Hands-On Deep Learning for Images with TensorFlow: Build inteligente computer vision applications using TensorFlow and Keras**. Birmingham: Packt Publishing, 2018.

BARBON, A. P. A. d. C. et al. Development of a flexible Computer Vision System for marbling classification. **Computer and Electronics in Agriculture**, Amsterdam, v. 142, p. 536-544, 2017.

BRAGA, A. de P.; LUDERMIR, T. B.; CARVALHO, A. C. P. de L. P. **Redes neurais artificiais: teoria e aplicações**. Rio de Janeiro: LTC, 2007.

CANADA. **Beef, Bison and Veal Carcass Grade Requirements**. Calgary, AB: CBGA, 2020.

CBGA. **Canadian Beef Grading Agency**, 2017. The agency. Disponível em: <https://beefgradingagency.ca/>. Acesso em: 1 dez. 2021.

CHENG, W. et al. Marbling analysis for evaluating meat quality: Methods and techniques. **Comprehensive Reviews in Food Science and Food Safety**, Chicago, v. 14, n. 5, p. 523–535, 2015.

DANSON-HOWE, K. **A Practical Introduction to Computer Vision with OpenCV**. New Jersey: Wiley, 2014.

FORSYTH, D. A.; PONCE J. **Computer Vision: A Modern Approach**. 2. ed. [S.l.]: Pearson, 2012.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn & TensorFlow: concepts, tools, and techniques to build inteligente systems**. 2. ed. Sebastopol: O'Reilly, 2019.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. New Jersey: Pearson Prentice Hall, 2008.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA: MIT Press, 2017.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural Features for Image Classification. **IEEE Trans Syst Man Cybern**, [S.l.], v. 3, n.6, p. 610-621, 1973.

HE, K. et al. Mask R-CNN. 2017 **IEEE International Conference on Computer Vision (ICCV)**, Venice, p. 2980-2988, 2017.

HOSSEINPOUR, S.; ILKHCHI, A. H.; AGHBASHLO, M. An intelligent machine vision-based smartphone app for beef quality evaluation. **Journal of Food Engineering**, Amsterdam, v. 248, p. 9-22, 2019.

HUANG, D. et al., Local Binary Patterns and Its Application to Facial Image Analysis: A Survey, **IEEE Trans Syst Man Cybern**, [S.l.], v. 41, n. 6, p. 765-781, 2011.

IGLESIA, D. H. de La. et al. Non-invasive automatic beef carcass classification based on sensor network and image analysis. **Future Generation Computer Systems**, Amsterdam, v. 113, p. 318-328, 2020.

JAMES, G. et al. **An Introduction to Statistical Learning: with Applications in R**. New York: Springer, 2015.

JAY, M. **Mask RCNN with Keras and Tensorflow (pt.1) Setup and Installation**. 2018. (13m48s). Disponível em:

<<https://www.youtube.com/watch?v=2TikTv6PWDw&list=PLX-LrBk6h3wRAF22jBUxDgOvyhlgLN4Cg>> Acesso em: 03 dez. 2021.

KENTARO, W. **Labelme: Image Polygonal Annotation with Python**. 2016. Disponível em: <<https://github.com/wkentaro/labelme>>. Acesso em: 01 dez. 2021.

KUBAT M. **An Introduction to Machine Learning**. 2. ed. Switzerland: Springer, 2017.

KUZILEK, J. et al. Independent Component Analysis and Decision Trees for ECG Holter Recording De-Noising. **PloS One**, [S.l.], v. 9, n. 6, p. e98450, 2014.

LEE, B.; YOON, S.; CHOI, Y. M. Comparison of marbling fleck characteristics between beef marbling grades and its effect on sensory quality characteristics in high-marbled Hanwoo steer. **Meat Science**, Amsterdam, v. 152, p. 109-115, 2019.

LI, S. Z.; JAIN, A. K. **Handbook of face recognition**. 2. ed. London: Springer, 2011.

LIU, J. -H. et al. Predicting pork loin intramuscular fat using computer vision system. **Meat Science**, Amsterdam, v. 143, p. 18-23, 2018.

MEUNIER, B. et al. An open-access computer image analysis (CIA) method to predict meat and fat content from an android smartphone-derived picture of the bovine 5th-6th rib. **Methods**, [S.l.], v.186, p. 79-89, 2021.

MÜLLER, A. C.; GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists**. Sebastopol: O'Reilly, 2016.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Cambridge: MIT Press, 2012.

NIXON, M. S.; AGUADO, A. S. **Feature Extraction and Image Processing for Computer Vision**. 4. ed. United Kingdom: Elsevier, 2020.

OJALA, T.; PIETIKÄINEN, M; HARWOOD, M. A comparative study of A Comparative Study of Texture Measures with Classification Based on Featured Distribution. **Pattern Recognit.** [S.l.], v. 29, p. 51-59, 1996

OJALA, T.; PIETIKÄINEN, M; MÄENPÄÄ, T. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v. 24, n. 7, p. 971-987, 2002.

PANG, B. et al. Grading of beef marbling based on image processing and support vector machine. **Computer Modelling and New Technologies**, [S.l.], v. 17, n. 3, p. 87-92, 2014.

PATTERSON, J.; GIBSON, A. **Deep Learning: A Practitioner's Approach**. Sebastopol: O'Reilly, 2017.

PLATTER, W. J. et al. Effects of marbling and shear force on consumers' willingness to pay for beef strip loin steaks. **Journal of Animal Science**, Oregon, v. 83, n. 4, p. 890–899, 2005.

RASCHKA, S.; MIRJALILI, V. **Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow**. 2. ed. [S.l.]: Packt Publishing Ltd, 2017.

ROSEBROCK, A. **Deep Learning for Computer Vision with Python: Starter Bundle**. [S.l.]: PyImageSearch, 2017.

ROSEBROCK, A. **Local Binary Patterns with Python & OpenCV**. 2015. Disponível em: <<https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>>. Acesso em: 1 dez. 2021.

SHALEV-SHWARTZ, S. BEN-DAVID, S. **Understanding Machine Learning: From Theory to Algorithms**. New York: Cambridge University Press, 2014.

SHIN, J. **Iris.csv**. 2014. Disponível em: <<https://gist.github.com/netj/8836201>>. Acesso em: 5 dez. 2021.

SINGH, P.; MANURE, A. **Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python**. New York: Apress, 2020.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. London: Springer-Verlag, 2010.

TAN, P. et al. **Introduction to Data Mining**. 2. ed. [S.l.]: Pearson, 2019.

UNITED STATE OF AMERICA. **Method for Grid Assessment of Beef Carcass Ribeye Area**. Washington, DC: USDA, 2011.

ZANGH; A. et al. Dive into Deep Learning. *In*: Cornell University. **arXiv**, 2021.
Disponível em: <<https://arxiv.org/abs/2106.11342>>. Acesso em: 1 dez. 2021.