UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ENGENHARIA ELETRÔNICA

ALEXANDRE OPECK DE MORAIS BORDIGNON
ANDERSON HIDEYUKI TURUTA
GUILHERME ELEUTERIO ARIELLO

IDENTIFICAÇÃO DE BOVINOS PELO ESTUDO DO ESPELHO NASAL

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2018

ALEXANDRE OPECK DE MORAIS BORDIGNON
ANDERSON HIDEYUKI TURUTA
GUILHERME ELEUTERIO ARIELLO

IDENTIFICAÇÃO DE BOVINOS PELO ESTUDO DO ESPELHO NASAL

Trabalho de conclusão de curso apresentado a disciplina Trabalho de Conclusão de Curso 2, do curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica .

Orientador: Prof. Dr. André Eugenio Lazzaretti

CURITIBA

2018

ALEXANDRE OPECK DE MORAIS BORDIGNON
ANDERSON HIDEYUKI TURUTA
GUILHERME ELEUTERIO ARIELLO

IDENTIFICAÇÃO DE BOVINOS PELO ESTUDO DO ESPELHO NASAL

Este Trabalho de Conclusão de Curso de Graduação foi apresentado como requisito parcial para obtenção do título de Engenheiro Eletrônico, do curso de Engenharia Eletrônica do Departamento Acadêmico de Eletrônica (DAELN) outorgado pela Universidade Tecnológica Federal do Paraná (UTFPR). Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Curitiba, 6 de agosto de 2018.

_____
Prof. Dr. Robinson Vida Noronha
Coordenador de Curso
Engenharia Eletrônica

_____
Profª. Drª. Carmen Caroline Rasera
Responsável pelos Trabalhos de Conclusão de Curso
de Engenharia Eletrônica do DAELN

BANCA EXAMINADORA

_____
Prof. Dr. André Eugenio Lazaretti
Universidade Tecnológica Federal do Paraná
Orientador

_____
Prof. Dr. Ricardo Umbria Pedroni
Universidade Tecnológica Federal do Paraná

_____
Prof. Dr. Gustavo Benvenutti Borba
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia Eletrônica.

Dedicamos este trabalho às nossas famílias e aos colegas de pesquisa.

AGRADECIMENTOS

Ao nosso orientador Prof. Dr. André Eugenio Lazzaretti, pela dedicação e apoio na elaboração deste trabalho.

Aos colegas do Instituto Agronômico do Paraná (IAPAR) e da Universidade de São Paulo (USP), em particular ao Prof. Dr. Ernane Costa, pelas imagens fornecidas, que foram essenciais na nossa pesquisa.

Ao Prof. Dr. Gustavo Benvenutti Borba por ceder o laboratório para a realização de alguns testes.

Ao colega Fábio Crestani por compartilhar conhecimentos fundamentais na elaboração do trabalho escrito e ao colega Moritz Vogt por ajudar nos primeiros passos com a NVidia Jetson TX1. E a todos os que de alguma maneira contribuíram para a realização desse trabalho.

RESUMO

BORDIGNON, A. O. M.; TURUTA, A. H.; ARIELLO, G. E.. IDENTIFICAÇÃO DE BOVINOS PELO ESTUDO DO ESPELHO NASAL. 80 f. Trabalho de conclusão de curso – Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

A identificação rápida e confiável de bovinos em um sistema produtivo é importante sob diversos aspectos, e por isso tem ganhado cada vez mais atenção entre pesquisadores em anos recentes. Como exemplo disso pode-se citar o controle necessário sobre rebanhos enviados a frigoríficos ou fazendas de ordenha. Os sistemas de identificação tradicionais, presentes no mercado possuem diversas falhas, como perda de informações e possibilidades de fraudes. Esse fato, aliado às exigências na qualidade e rastreabilidade na exportação de bovinos, cria a necessidade de uma alternativa à esses sistemas. Partindo desse cenário este trabalho foi desenvolvido para analisar a possibilidade de identificação biométrica de bovinos a partir de imagens do espelho nasal, região central localizada abaixo das narinas e acima do lábio superior do animal, e imagens do focinho inteiro do bovino, tendo em vista que, assim como a impressão digital em seres humanos, os bovinos apresentam unicidade nas características biométricas dessas regiões. Nossa pesquisa investigou diversos algoritmos de processamento digital de imagens, extração de características e classificação, bem como a possibilidade de transferir o software desenvolvido para um sistema embarcado em uma placa Nvidia Jetson TX1, buscando abrir um precedente para o desenvolvimento futuro de um sistema de identificação de bovinos em tempo real. Taxas de acerto de 100% foram obtidas em um computador i7, através dos algoritmos extratores de características PCA, SIFT, SURF e de uma Rede Neural Convolucional, e dos algoritmos de classificação kNN e SVM, quando os mesmos foram aplicados ao conjunto de imagens cedidos pela Universidade de São Paulo. Estes mesmos testes, realizados no conjunto de imagens cedidos pelo Instituto Agronômico do Paraná geraram taxas de acerto de 100% apenas no caso em que a Rede Neural Convolucional foi usada como extrator de características. Na placa Nvidia, o conjunto de imagens da USP gerou taxas de acerto de 99,51%, enquanto o conjunto de imagens do IAPAR gerou taxas de acerto de 93,65%.

Palavras-chave: Sistema embarcado, Aprendizado de máquina, Biometria animal, Extração de características, Rastreabilidade

ABSTRACT

BORDIGNON, A. O. M.; TURUTA, A. H.; ARIELLO, G. E.. CATTLE BIOMETRIC IDENTIFICATION FROM MUZZLE PATTERN IMAGES. 80 f. Trabalho de conclusão de curso – Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

The identification of animals inside a production chain, in a fast and reliable way, is important in many different aspects, and for this reason it has been gaining more attention of researchers over the recent years. As an example of this, it can be cited the necessary control over cattles sent to a frigorific or to milking farms. The traditional identification systems available in the current market have various issues, as information loss and fraud possibilities. This fact, allied with the demand of quality and traceability of bovines exported, creates a necessity for an alternative to this system. Starting from this scenario, this work was developed to analyse the possibility of biometrical identification of cattles through the image of their central nasal pattern, center region, located below the nostril and above the upper lip of the animal, and the image of the whole muzzle of the bovine. In view of, as fingerprint in human beings, bovines show oneness at the biometrical characteristics at these regions. Our research investigated various algorithms of digital imaging processing, characteristics extraction and classification, as well as the possibility of moving the developed software to a embedded system over NVidia Jetson TX1 board, looking for setting a precedent to developing a real time bovines identification system. Hit percentages of 100% were obtained using a i7 computer, using characteristics extraction algorithms PCA, SIFT, SURF and a Convolutional Neural Network, and the classification algorithms kNN and SVM, when they were applied to a image set provided by the University of São Paulo. The same tests, when applied at the image set provided by the Agronomic Institute of Paraná had a hit percentage of 100% only in the case when the Convolutional Neural Network was used as an extractor of characteristics. With the NVidia board, USP's image set had a hit percentage of 99.51%, while IAPAR image set had a hit percentage of 93.65%.

Keywords: Embedded system, Machine Learning, Animal biometry, Feature extraction, Traceability

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

RFID	Radio-Frequency Identification
MLP	Multilayer Perceptron
kNN	k-Nearest Neighbor
WLD	Weber Local Descriptor
AUC	Area Under Curve
EER	Equal Error Rate
Fk-NN	Fuzzy-k-Nearest Neighbor
LBP	Local Binary Pattern
GSRC	Group Sparse Representation based Classification
SRC	Sparse Representation Classifier
HOG	Histogram of Oriented Gradient
LTE	Laws Texture Energy
Fuzy-LBP	Fuzzy-Local Binary Pattern
RBPN	Radial Basis Probability Network
PNN	Probabilistic Neural Network
DT	Decision Tree
GMM	Gaussian Mixture Model
FLPP	Fisher locality preserving projections
OSS	One-shot similarity
ISVM	Incremental Support Vector Machines
LDA	Linear Discriminant Analysis
ICA	Independent Component Analysis
FCL	Fully Connected Layer
SDAE	Stacked Denoising Auto-encoder
DBN	Deep Belief Network
RBM	Restricted Boltzmann Machine
VLAD	Vector of Locally Aggregated Descriptor
SSCP	Pure Sums of Squares and Cross Products
CNN	Convolutional Neural Network
ROI	Region of Interest
IAPAR	Instituto Agronômico do Paraná
USP	Universidade de São Paulo
PDI	Processamento Digital de Imagens
CSV	Comma-separated values
CLAHE	Contrast-limited adaptive histogram equalization
SVM	Support Vector Machine
PCA	Principal Component Analysis
RBF	Radial Basis Function
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features

SUMÁRIO

1 INTRODUÇÃO

No mês de janeiro de 2018, o faturamento do Brasil, somente com exportação de carne bovina foi de 425,8 milhões de dólares (FORMIGOI, 2018). Levando em conta que historicamente o mês de janeiro não é o melhor período do ano para exportação de carne, e que nesse montante, não foram considerados os valores obtidos no consumo interno do país e em outros segmentos relacionados a pecuária, como por exemplo o consumo de leite, podemos perceber a importância econômica da atividade pecuária para o país.

Atualmente o processo de identificação de bovinos é passível de fraudes e falhas, o que possibilita embargos sanitários por parte dos importadores. Os métodos mais comuns de identificação presentes no mercado são anéis auriculares e marcação a ferro, no qual uma marca é feita no animal através de calor ou de congelamento. Esses métodos, apesar de serem os mais utilizados, apresentam vários problemas como a perda dos brincos e também a facilidade de modificação e fraude.

A proposta desse trabalho é investigar métodos de identificação utilizando o espelho nasal e o padrão de focinho dos bovinos, tendo em vista que essa é uma característica biométrica dos animais, e portanto é única e não é passível de fraudes. Além disso, é um método de identificação de baixo custo.

Outro fator importante considerado em nossa pesquisa foi a possibilidade de identificação dos bovinos em tempo real. Por isso os algoritmos foram testados em uma placa Nvidia Jetson TX1, de maneira que possam ser integrados futuramente com algoritmos de extração da ROI automática e que um sistema completo possa ser desenvolvido.

1.1 MOTIVAÇÃO

A tendência de mercado para o futuro, não só no Brasil, mas no mundo inteiro é a automatização de processos e atividades, levando em conta fatores como custo, praticidade e portabilidade. Dessa maneira acreditamos que esse seja um momento ideal para que

pesquisas em soluções embarcadas e inteligentes, que abrangem processamento em tempo real e algoritmos de Machine Learning e Deep Learning, sejam realizadas.

Além disso, a crescente demanda por rastreabilidade e segurança apresentada pelo mercado importador de bovinos, cria a necessidade do desenvolvimento de novas tecnologias, que forneçam uma solução melhor e mais barata para o problema.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

- Estudar alguns dos algoritmos existentes de processamento digital de imagens, extração de características e classificação de imagens buscando realizar a identificação biométrica de bovinos de maneira rápida e confiável. Analisar também o desempenho do sistema desenvolvido na plataforma NVidia Jetson TX1.

1.2.2 OBJETIVOS ESPECÍFICOS

- Tratamento inicial das imagens por extração da ROI.
- Investigar qual é o melhor algoritmo, ou combinação de algoritmos, para extração de características.
- Obter um classificador rápido e confiável para classificação em tempo real.
- Investigar o comportamento do sistema desenvolvido em um sistema embarcado sem custos elevados.

1.3 ESTRUTURA DO TRABALHO

No primeiro capítulo apresenta-se o projeto, uma breve contextualização e a problemática abordada, assim como os objetivos geral e específicos.

O segundo capítulo apresenta uma revisão da literatura, onde citamos alguns dos resultados que já foram obtidos em pesquisas relacionadas.

O terceiro capítulo apresenta a fundamentação teórica dos diversos métodos computacionais utilizados nesse trabalho para processamento digital das imagens, extração de características e classificação.

O quarto capítulo descreve a metodologia utilizada em nosso trabalho, aqui escrevemos como foram obtidos os datasets e também como foi feito o pré processamento das imagens.

No quinto capítulo apresentamos e discutimos os resultados obtidos com o auxílio de diversos gráficos e tabelas.

E por último, no sexto capítulo temos a conclusão e considerações finais.

2 REVISÃO DE LITERATURA

Nesta sessão, uma revisão de literatura é apresentada. Nela nós descrevemos importantes resultados obtidos em pesquisas relacionadas à identificação de bovinos, através de métodos biométricos, tendo em vista a crescente importância atribuída ao assunto nos últimos anos, devido a questões de segurança e confiabilidade (KUMAR et al., 2018).

Em se tratando de classificação animal, pesquisas recentes tem trazido ao cenário científico de Machine Learning e Deep Learning, casos com altos índices de acerto em classificação. Schutera et al. (2016), atingiram acurácia de 79-99% na classificação de fenótipos de peixe-zebra, utilizando SVM e sem nenhuma interação do usuário.

No caso específico de bovinos, o processo de classificação e identificação pode ser realizado através de diferentes métodos, como por exemplo métodos clássicos permanentes: entalhe de orelha, tatuagem de orelha, marcação com ferro quente e marcação com ferro congelado; métodos clássicos temporários como o uso de brincos e métodos elétricos como a tecnologia RFID. Em todos os casos, o objetivo é proporcionar um identificador ou marcador único para cada bovino (AWAD, 2016).

Recentemente, a identificação e classificação de bovinos tem desempenhado um papel importante na compreensão da trajetória de doenças, na gestão de produção e da vacinação, na rastreabilidade e na atribuição da propriedade dos animais. Contudo, os métodos clássicos tem performance limitada pela sua vulnerabilidade a perdas, duplicação e fraude. Métodos biométricos de identificação animal tem sido amplamente pesquisados nos últimos anos, dado o fato de que identificadores biométricos são únicos, imutáveis e possuem baixo custo. No caso de bovinos, os identificadores biométricos incluem o padrão de focinho ou espelho nasal, padrões de íris e padrões vasculares da retina (AWAD, 2016).

Em se tratando do padrão de focinho, inúmeros métodos de extração de características e classificação tem sidos pesquisados e testados, visando encontrar um sistema robusto, seguro e confiável na identificação dos animais. Gimenez (2011) obteve taxas de precisão superiores a 70% em um conjunto de 816 imagens e 51 bovinos. Os algoritmos de extração

de características e classificação utilizados foram uma rede neural MLP com uma camada escondida e posteriormente um kNN recebendo os valores dos três neurônios de saída da rede neural. Gimenez (2015) obteve índices de acerto de 95,33% a 99,52% utilizando o algoritmo PCA como extrator de características e o algoritmo SVM como classificador, após pré processar as imagens com algoritmos de aumento de contraste, detecção de bordas e esqueletização. Os resultados foram obtidos em um conjunto de 187 bovinos.

Em estudo posterior, Gaber et al. (2016) obtiveram aproximadamente 99,5% de acerto em um conjunto de 31 cabeças de bovinos, empregando o extrator de características WLD e o classificador AdaBoost, eles também mostraram através de quatro métodos diferentes: AUC, Sensibilidade e Especificidade, taxa de precisão e EER que o classificador AdaBoost teve desempenho superior aos classificadores kNN e Fk-NN.

Através do uso dos descritores de característica de textura SURF e LBP, em diferentes níveis da pirâmide Gaussiana, Kumar et al. (2016) obtiveram uma taxa de precisão de 93,87% em um conjunto de 500 bovinos. Nesse caso, os descritores de características adquiridos em cada nível Gaussiano, foram combinados através do método de fusão usando a regra de soma ponderada. O método de classificação *nearest-neighbour* foi usado para avaliar a correspondência entre o histograma LBP e os vetores de características SURF de uma imagem de entrada e os padrões para cada classe.

Kumar et al. (2017) propõem um método baseado em GSRC usando minimização-L2 e fusão através da regra da soma. Nesse método, o algoritmo pôde aprender uma representação de características discriminatória de imagens da face do bovino e de imagens do padrão de focinho, em um conjunto de 5000 imagens. As taxas de acerto relatadas foram 84,96% usando a regra da soma, 89,99% usando troca de contexto, 91,94% usando SRC clássico e 96,56% usando o método proposto.

Buscando a identificação e classificação de diferentes raças de bovinos, Kumar e Singh (2016) documentaram o uso de diversos algoritmos de extração de características e classificação em imagens adquiridas através de uma câmera de baixo custo. Os algoritmos de extração de características usados foram: algoritmos de características de textura Haralick, características baseadas em morfologia, características baseadas em forma, HOG, característica Wavelet, características baseadas em cor, características Tamura, LTE, características de textura LBP e Fuzy-LBP. Os algoritmos de classificação usados foram: kNN, Fk-NN, RBPN, PNN, DT, GMM, MLP e classificador baseado em Naive Bayes.

Por meio de uma câmera de segurança e uma conexão wireless, Kumar et al. (2017) capturaram imagens do padrão de focinho, em diferentes tamanhos, de diversos bovinos e

transferiram as mesmas para um servidor onde foi feita a classificação e o reconhecimento do bovino. O principal objetivo da pesquisa foi investigar uma alternativa para reconhecimento de bovinos em tempo real. O extrator de características proposto nesse trabalho foi o FLPP, os algoritmos de classificação usados foram OSS com ISVM. A taxa de precisão alcançada por meio dessa abordagem foi de 96,87% com um tempo de processamento de 10,25s. Para fins de avaliação e comparação foram usados os algoritmos clássicos de extração de características: SURF, LBP, PCA, LDA e ICA.

Seguindo a recente tendência de abordar problemas de Machine Learning através de algoritmos de Deep Learning, Kumar et al. (2018) propuseram um framework para reconhecimento de bovinos usando padrão de focinho. Nesse trabalho, a extração de características bem como a classificação das imagens foram feitas através de uma CNN com uma camada final FCL, uma SDAE e uma DBN com RBM. As taxas de precisão foram 75,98%, 88,46% e 95,99% respectivamente, em um conjunto de 5000 imagens de 500 bovinos. Para fins de avaliação e comparação foram usados os algoritmos clássicos de extração de características: LBP, Circular-LBP, SIFT, Dense-SIFT, SURF e VLAD.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 SUPPORT VECTOR MACHINES (SVM)

Traduzindo para o português *Support Vector Machines* (SVM) torna-se Máquina de Vetores de Suporte, é um conceito conhecido na computação como um conjunto de métodos que analisam dados e reconhecem padrões, normalmente ele é utilizado para classificação. Em outras palavras é um padrão que utiliza um conjunto de dados como entrada e consegue predizer a categorias ou classes, as quais esses dados pertencem. Ele é um classificador binário linear, dessa maneira o algoritmo encontra um plano n-dimensional que fica o mais distante possível das categorias pré definidas.

Para o treinamento foram utilizados todos os dados, porém na avaliação em si foram utilizados apenas os vetores suporte, como visto na Figura 1.

Ainda da Figura 1, chama-se as linhas pontilhadas de planos de suporte, pois eles são adequados para encontrar a linha de decisão.

No nosso caso, contudo, os dados não podem ser separados em dois por um plano rígido, pois temos a presença de ruídos. Então nesse caso são utilizadas margens suaves ao invés das rígidas (LORENA; CARVALHO, 2007) permitindo que alguns dados permaneçam entre os hiperplanos ocasionando alguns erros de classificação.

Existe também a situação em que os dados não são linearmente separáveis, a versão suave tolera apenas poucos ruídos, porém existem muitos casos em que não é possível dividir satisfatoriamente o hiperplano como por exemplo da Figura 2. Nesse caso o espaço é transformado utilizando uma função Kernel (KOERICH, 2003).

As SVMs são robustas diante de dados de grande dimensão, enquanto outras técnicas de aprendizado normalmente obtém classificadores super ou sub ajustados, e também a utilização de Kernel faz com que o algoritmo seja eficiente, pois permite a construção de simples hiperplanos em um espaço de alta dimensão de forma tratável do ponto de vista computacional. (LORENA; CARVALHO, 2007)

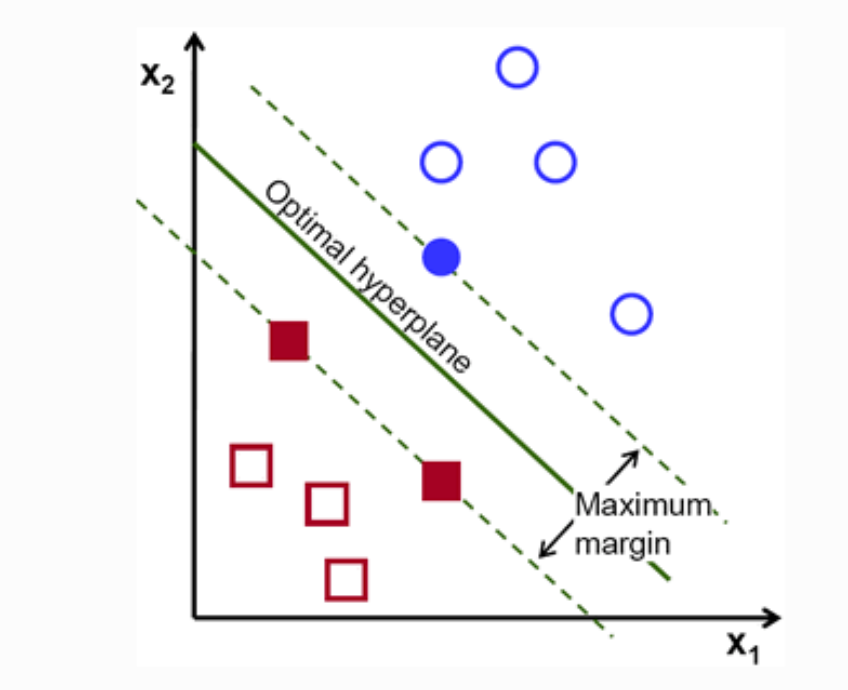

Figura 1: Lógica do SVM

Fonte: (MORDVINTSEV, 2013)

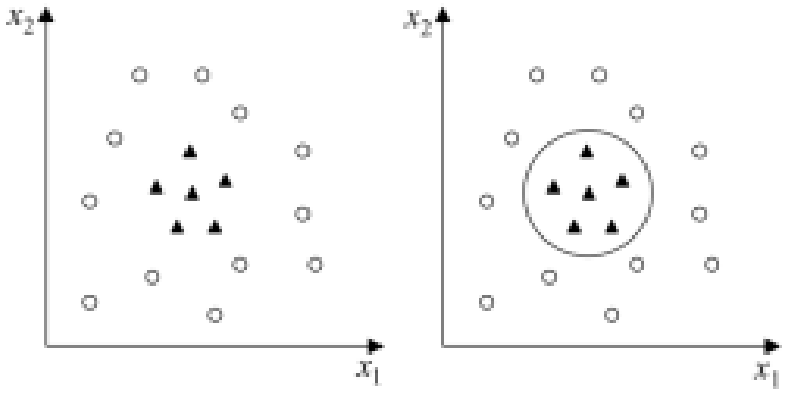

Figura 2: SVM com conjunto de dados não linear

Fonte: (LORENA; CARVALHO, 2007)

3.2 PRINCIPAL COMPONENT ANALISYS (PCA)

O *Principal Component Analysis* (PCA) é um processo estatístico que extrai as características mais importantes de um dataset. Um dos principais efeitos desse algoritmo é a redução do número de dimensões do pacote de dados utilizado.

Esse algoritmo permite encontrar a direção na qual a variância dos dados é maior. Então no resultado temos basicamente autovetores que definem os principais componentes dos dados analisados. O tamanho do vetor principal, ou componente principal, depende de quanto os dados variam na direção de maior variância, sendo que o começo do vetor está localizado no centro de todos os pontos do dataset. O número de vetores é o mesmo número de dimensões que os dados possuem.

Na Figura 3 temos um exemplo do algoritmo PCA em funcionamento, primeiro é detectado as bordas da figura e depois determinada em qual direção temos a maior variância.

Normalmente a sua análise é feita em uma matriz quadrada, que pode ser do tipo SSCP, covariância ou correlação. Nas duas primeiras os resultados dos objetos não são diferentes, pois a principal diferença está no fator de escala global. Já a matriz de correlação é utilizada se as variâncias das variáveis individuais são muito grandes (NCSU, 2007).

3.3 SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

Scale-invariant Feature Transform (SIFT) é um algorítimo de descrição de imagens utilizado em reconhecimento e descrição de pontos em comum de diferentes vistas de um objeto em 3D. Desenvolvido por David Lowe, SIFT tem como sua principal aplicação no âmbito da visão computacional. Uma de suas características é o fato de ser invariante à translações, rotações e transformações de escala (LINDEBERG, 2015). Existem quatro passos que envolvem o método: detecção de extremos, localização de pontos-chave, orientação e descrição (LOWE, 2004).

A detecção de extremos utiliza de diferenças de Gaussianas para identificar pontos de interesse que são invariantes em escala e orientação, isto quer dizer, pontos que são identificados com a câmera próxima ou longe do objeto desejado. Essa função Gaussiana usa subtrações de uma imagem com outra para aprimorar seus pontos em comum, e tem como objetivo eliminar ruídos e detalhes indesejados, e ainda, realçar características importantes. A Figura 4 é um exemplo visual da geração de funções gaussianas para diferentes escalas de uma imagem.

O segundo passo do algoritmo é a localização dos pontos-chave. Isto é feito através

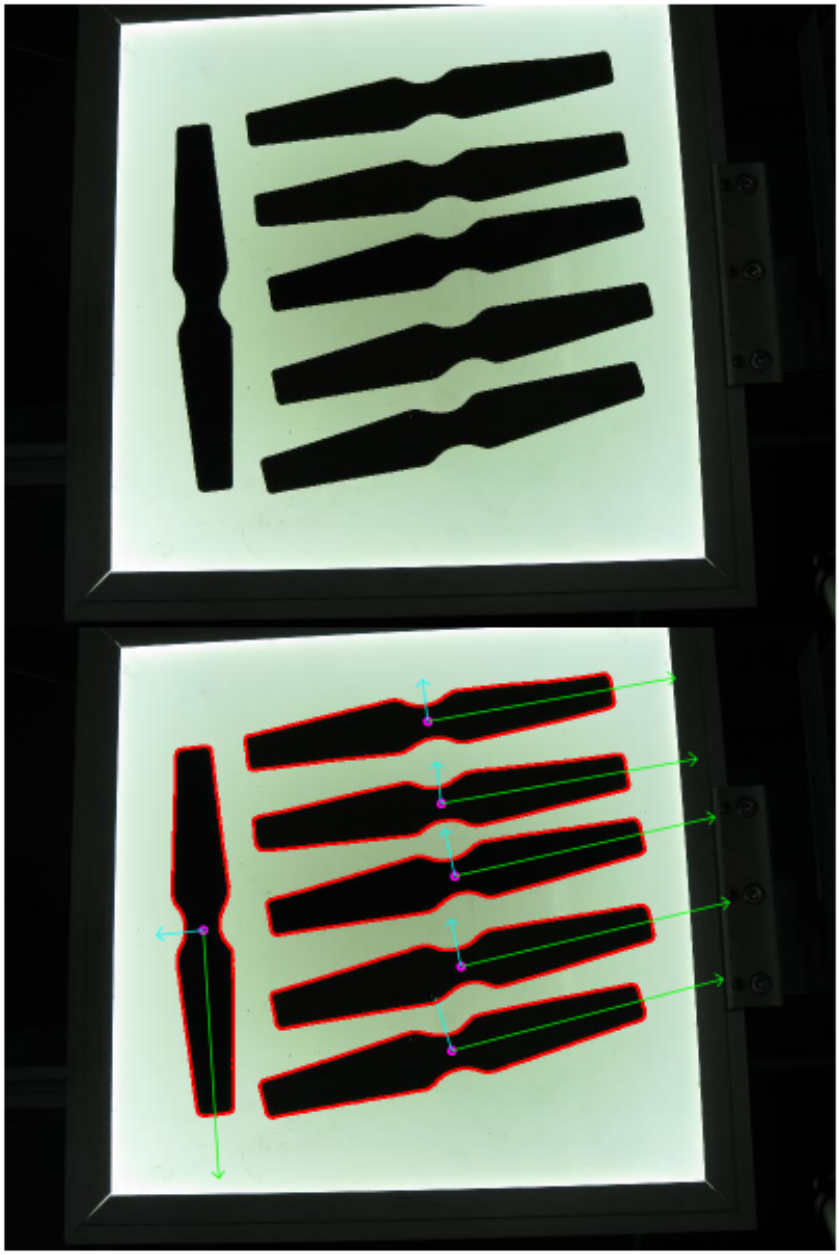

Figura 3: Exemplo do funcionamento do PCA

Fonte: (HEESCH, 2015)

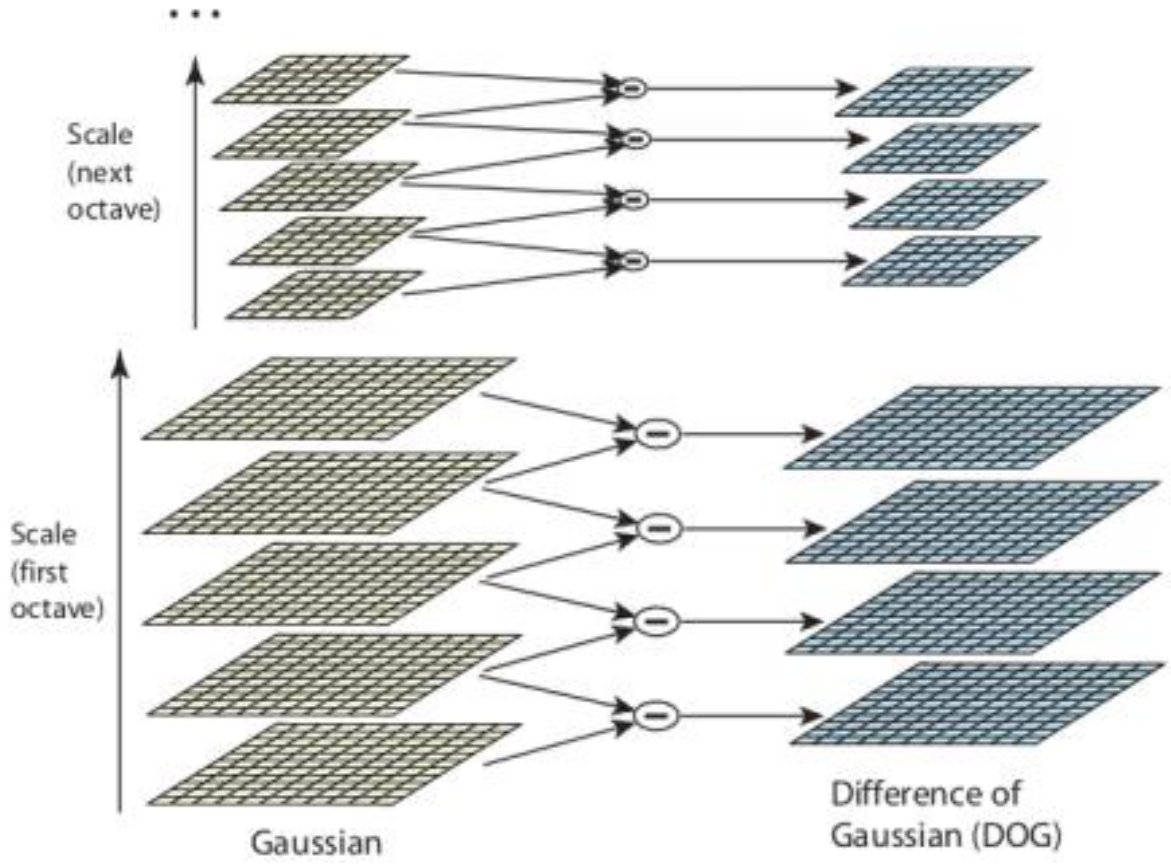

Figura 4: Processo de subtração por Difference of Gaussian
Fonte: (LOWE, 2004)

da criação de modelos que determinam localização e escala. Desta maneira, os mais estáveis são selecionados. Para isto, uma expansão de Taylor da função de Diferenças de Gaussianas é aplicada de forma que sua origem esteja localizada no ponto de amostragem.

Cada modelo de localização recebe uma ou mais orientações, baseadas na direção do seu gradiente. Desta forma, todas as outras operações sobre imagens são baseadas nestes gradientes, fazendo com que sejam invariantes às transformações.

Após terem sido criados e identificados os modelos mais estáveis, analisa-se os pontos em comum dos vizinhos mais próximos de duas imagens distintas e cria-se histogramas dos mesmos para análise de similaridades, alcançando assim robustez quanto à iluminação, escala, translação, etc (LOWE, 2004). A Figura 5 mostra um exemplo de aplicação do método SIFT para identificação de pontos-chave no focinho de um bovino.

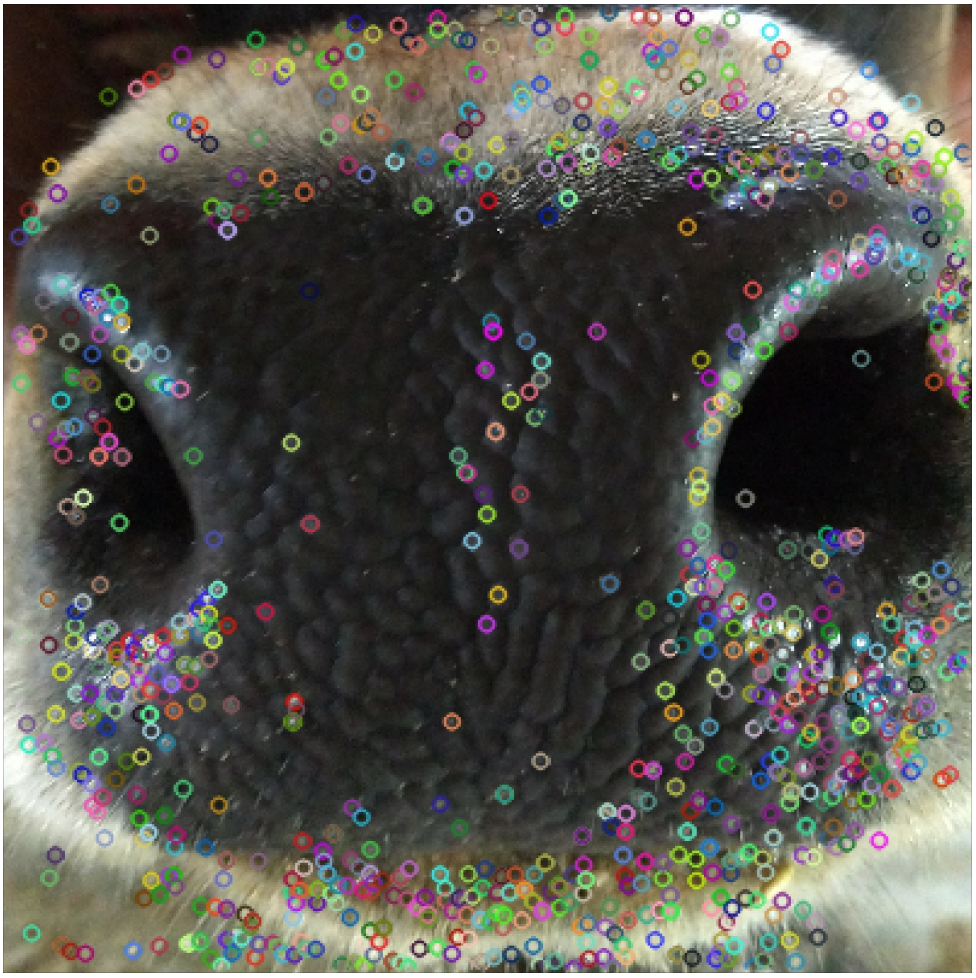

Figura 5: Aplicação do SIFT para identificação de pontos-chave em bovinos

Fonte: Autores(2018)

3.4 SPEEDED-UP ROBUST FEATURES (SURF)

Com o intuito de criar um algoritmo mais rápido, Hebert Bay, Tinne Tuytelars e Luc Van Gool, desenvolveram o SURF, *Speed-Up Robust Features*. Sua principal característica é a facilidade no cálculo de convoluções com base em imagens integrais, além de que pode ser feita em paralelo com outras escalas (BAY et al., 2008). E sua implementação tenta ser invariante à rotação da imagem.

O princípio de funcionamento do método SIFT é similar ao SURF, onde os pontos de interesse, para que sejam invariantes à rotação, são dotados de uma orientação. Esta é definida através do cálculo da resposta Haar-wavelet tanto no eixo horizontal quanto vertical para um vizinho de tamanho $6s$, sendo s a escala na qual o ponto de interesse foi identificado.

Após calculado as respostas de wavelet com sua Gaussiana centrada no ponto de interesse as respostas são mostradas como vetores. Em seguida, deve-se calcular a orientação dominante, que é feito através da soma de todas as respostas deslocadas de uma janela com ângulo final de 60 graus. As respostas verticais e horizontais dessa janela são somadas, criando assim um novo vetor.

Para a criação do descritor, uma vizinhança de dimensões 20sx20s é definida ao redor do ponto de interesse. Essa vizinhança é novamente dividida em quatro sub-regiões e suas respostas de wavelet são novamente calculadas. Estas dimensões, quando representadas como vetores, dão um total de 64 dimensões, onde, quanto menor for seu tamanho, mais rápido e menos robusto é sua computação. No passo de comparação, apenas partes com o mesmo tipo de contraste são comparadas (BAY et al., 2008).

A Figura 6, demonstra o resultado gerado pela aplicação do método de extração de características SURF, em uma das imagens analisadas nesse trabalho.

3.5 K-NEAREST NEIGHBOUR (KNN)

É um dos algoritmos de classificação mais simples disponíveis para o aprendizado de máquina.

Sua lógica pode ser explicada com o auxilio da Figura 7, nela temos duas famílias, os quadrados azuis e os triângulos vermelhos, observamos que cada uma delas tem a sua respectiva área de predominância, que é denominada cidade. Em seguida é introduzido um novo membro, o círculo verde, ele deve ser adicionado a uma das duas famílias.

O método kNN consiste inicialmente na verificação de qual é o seu vizinho mais próximo, e pela imagem nota-se que é o triangulo vermelho, nessa situação foi aplicado somente o *Nearest Neighbour*.

Mas então surge um problema, o triângulo vermelho é o mais próximo do circulo verde, porém existe mais quadrados azuis na sua proximidade, com isso somente o método do *Nearest Neighbour* não é suficiente. Então é checado as k famílias mais próximas.

Para a Figura 7 se for utilizado um valor de k = 3 os triângulos vermelhos vencem, mas se for utilizado um k = 5, então os quadrados azuis levam vantagem.

Esse método pode não ser muito justo, como no caso de k = 4, que seria um empate, por isso o valor de k sempre é um número ímpar. E também são adicionados ”pesos”que dependem da distância do novo membro da família. Sendo mais peso para os mais próximos e menos peso

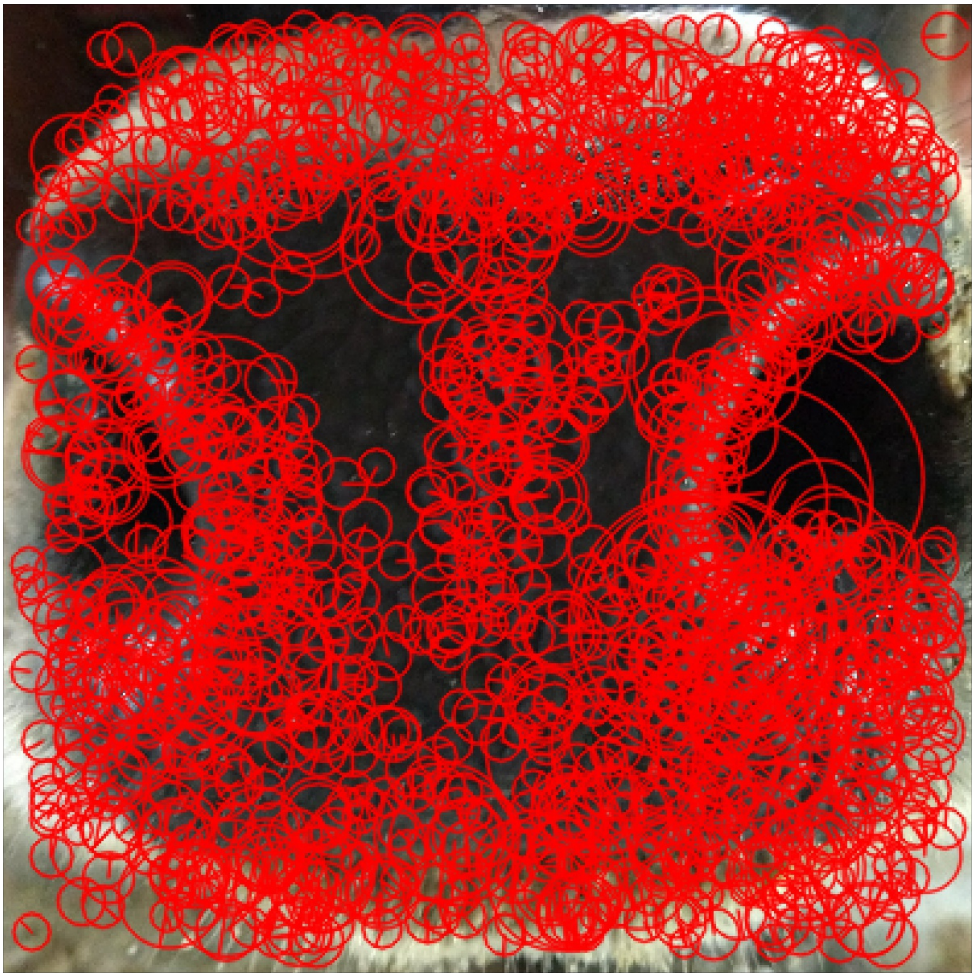

Figura 6: Aplicação do SURF

Fonte: Autores(2018)

para os mais longe distantes (OPENCV, 2014).

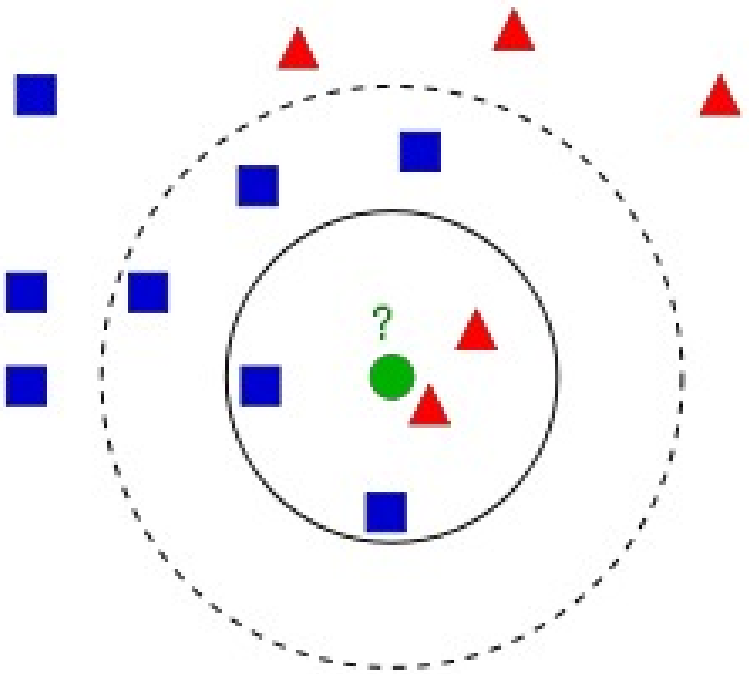

Figura 7: Lógica do kNN

Fonte: (OPENCV, 2014)

3.6 CONVOLUTIONAL NEURAL NETWORK (CNN)

Redes Neurais Convolucionais ou *Convolutional Neural Networks* (CNNs ou ConvNets) são tipos especiais de Redes Neurais regulares, ou Redes Neurais do tipo *feed-forward*. As CNNs apresentam a característica de funcionarem muito bem em tarefas de reconhecimento visual, uma vez que elas foram concebidas para emular o Córtex Visual (MOUJAHID, 2016).

Essa categoria de Redes Neurais foi introduzida em 1998, em um paper publicado por Bengio, Le Cun, Bottou and Haffner, essa primeira CNN foi chamada de LeNet-5 e era capaz de classificar dígitos escritos a mão (CORNELISSE, 2018).

A principal operação executada por uma CNN é a convolução, daí seu nome. As CNNs apresentam *layers* (camadas) especiais que conseguem extrair características de uma determinada imagem de entrada, desde características simples como linhas horizontais e verticais até características mais complexas, como determinar se um objeto é um carro ou uma pessoa. A extração dessas características é feita através da operação de convolução entre os filtros presentes um uma determinada *layer* e a imagem de entrada e através da operação de *pooling*. A operação de convolução é realizada pelo posicionamento do filtro sobre uma determinada região da imagem seguido da multiplicação do filtro pela mesma região, elemento por elemento e da soma de todas as multiplicações, conforme mostrado na Figura 8.

A operação de *pooling* é executada geralmente após uma operação de convolução, e tem como principal objetivo a redução de dimensões da matriz que resulta da aplicação dos filtros. O *pooling* mais comumente utilizado é o *max pooling*, que simplesmente seleciona o maior valor dentro da janela de seleção, conforme mostrado na Figura 9, existe também o *pooling* que faz a média de todos os valores contidos na janela, mas essa modalidade é menos comum.

As CNNs possuem diversas arquiteturas diferentes, entre elas está a VGG16, arquitetura introduzida por (SIMONYAN; ZISSERMAN, 2014) em 2014. Essa arquitetura é caracterizada pela sua simplicidade, usando apenas camadas convolucionais 3x3, empilhadas no topo umas das outras, com profundidade crescente, a redução de volume é tratada pelos filtros *max pooling*, e no final da arquitetura existem duas camadas FCL seguidas por um classificador siftmax (ROSEBROCK, 2017). A Figura 10 mostra a arquitetura VGG16.

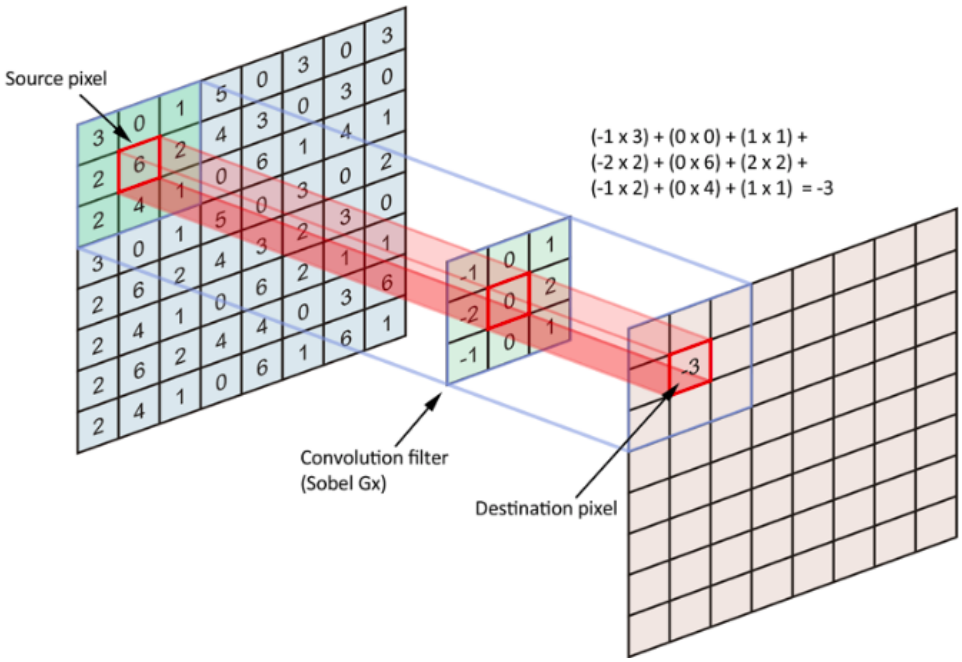

Figura 8: Convolução realizada em uma CNN

Fonte: (CORNELISSE, 2018)

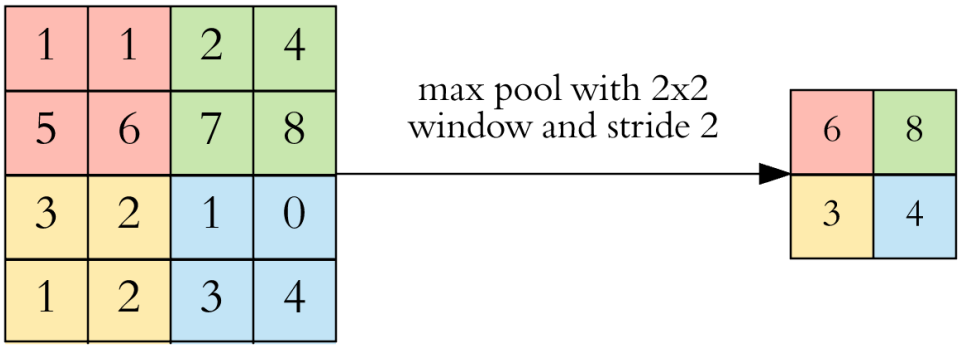

Figura 9: Max pooling

Fonte: (DERTAT, 2017)

3.7 TRANSFER LEARNING

Quando o conjunto de imagens, disponíveis para o treinamento de uma CNN, não é grande o suficiente, uma alternativa rápida e de baixo custo computacional, é a aplicação do conceito de *Transfer Learning*. Nessa técnica, as últimas camadas de uma rede neural, treinada para resolver um problema diferente do problema em questão, são retiradas, e novas camadas são definidas e treinadas para o novo problema. Desta maneira, o custo computacional exigido, é apenas aquele necessário para treinar as últimas camadas da rede, uma vez que os pesos, nos

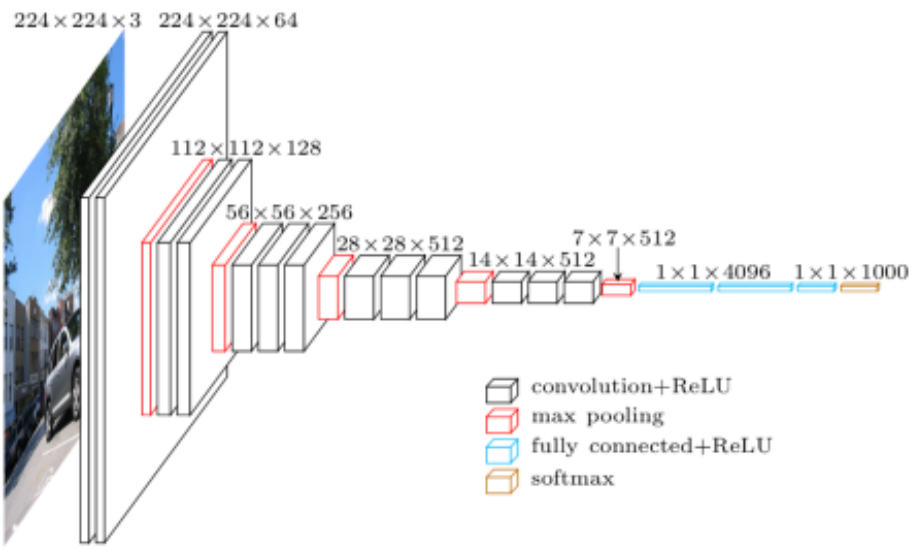

Figura 10: VGG16

Fonte: (ROSEBROCK, 2017)

nós das camadas iniciais, são mantidos.

Na utilização da técnica de Transfer Learning, o projetista tem liberdade para retirar apenas uma camada, ou mais que uma. Nesse sentido existe uma relação de *trade off*, uma vez que, quanto mais camadas forem retiradas, mais custo computacional será necessário para retreinar a rede em função do novo problema, contudo quanto maior o número de camadas retreinadas, mais personalizada a rede será para o novo problema. Desta maneira, o projetista pode retirar as camadas conforme o poder computacional disponível para treinar as novas camadas.

No caso da rede neural com arquitetura VGG16, uma possibilidade razoável é a retirada das duas últimas camadas FCL e a utilização das camadas anteriores como extratoras de características.

Quando a técnica de *Transfer Learning* é utilizada, uma alternativa razoável é a utilização das redes já treinadas, disponíveis no desafio ILSVRC (RUSSAKOVSKY et al., 2015). Este desafio é uma fonte conhecida de CNNs. Nele as CNNs são treinadas em aproximadamente 1,2 milhões de imagens com outras 50000 para validação e 100000 imagens para teste. O objetivo desse desafio é treinar um modelo que consiga classificar uma imagem em alguma das 1000 categorias disponíveis (ROSEBROCK, 2017). Essa foi a abordagem utilizada nesse trabalho.

3.8 ALGORITMOS DE PROCESSAMENTO DIGITAL DE IMAGENS

3.8.1 CONTRAST-LIMITED ADAPTIVE HISTOGRAM EQUALIZATION (CLAHE)

Métodos convencionais que utilizam equalização de histogramas assumem que a qualidade da imagem é uniforme sobre todas as áreas e um único mapeamento em escala de cinza fornece realce semelhante para todas as regiões da imagem. Contudo, se esta suposição não for mais verdadeira, técnicas com histograma adaptativo possuem uma performance superior na equalização da imagem.(REZA, 2003)

O CLAHE é uma variação do AHE (adaptative histogram equalization), esse algoritmo é conhecido por melhorar o contraste das imagens. A principal diferença em relação aos outros métodos de equalização de histograma é que ele computa inúmeros histogramas para cada seção da imagem, e com isso redistribui os valores de iluminação. Esse processo se repete para cada pixel da imagem, fazendo com que seja necessário um grande esforço computacional, e por isso não pode ser utilizado em manipulação de imagens em tempo real (PISANO S. ZONG, 1990).

3.8.2 FILTRO SOBEL

O Sobel é uma técnica em processamento digital de imagens que foca em encontrar o contorno dos objetos, foi criada por Irwin Sobel e Gary Feldman, que estudaram junto na SAIL (Stanford Artificial Intelligence Laboratory).

Basicamente esse método consiste num operador que consegue calcular o número de diferenças finitas, com uma aproximação do gradiente da intensidade dos pixels da imagem, ele fornece a direção da maior variação de claro para escuro em cada ponto.

As variações intensas entre o claro e o escuro correspondem as fronteiras, e consequentemente aos contornos dos objetos.

Matematicamente, são utilizadas duas matrizes 3x3 que convoluem com a imagem original para obter as aproximações das derivadas, tanto na vertical quanto na horizontal (SOBEL, 2014).

3.8.3 ESCALA DE CINZA

Uma das peças fundamentais para executar todos os algoritmos de processamento digital de imagens é a conversão das fotos para escala de cinza. Este passo foi dado logo no pré processamento, quando foi feito o modelo de dados HDF5. A sua importância se dá por

diversos fatores, sendo alguns dos principais, a relação sinal e ruído, complexidade do código, visualização, velocidade de execução e aprendizado de máquina (BURGUER, 2009).

Converter as imagens para escala de cinza reflete na possibilidade de trabalhar com a relação sinal e ruído desta imagem, sendo importante para a detecção de bordas. Existem algumas exceções, como a aplicação na agricultura de precisão, onde as cores ajudam a diferenciar uma laranja de sua folha, por exemplo.

A complexidade do código e sua velocidade de execução estão muito atreladas, é fácil perceber que trabalhar com uma escala só exige menos processamento do que trabalhar com três(sendo este o caso do RGB), logo, a conversão para escala de cinza torna o processamento mais eficiente e veloz.

No âmbito de nosso trabalho, quando se trabalha com um conjunto de imagens grande é necessário padroniza-lo para que a métodos de aprendizado de máquina e a própria rede neural consigam ser mais precisas. Por isso, ao converter as imagens, diferenças de como por exemplo luminosidade em diversos ângulos se minimizam, facilitando o próprio algorítimo a criar seus padrões.

3.8.4 BINARIZAÇÃO POR OTSU

Usando o princípio da conversão em escala de cinza, Nobuyuki Otsu desenvolveu o método que leva seu sobrenome e tem como objetivo criar um limiar que determine os elementos de fundo e frente da imagem. Com otimização para imagens com histogramas bimodais, este método cria dois clusters atribuindo a cor branca e preta para cada um deles (TOROK, 2015).

O cálculo da variância leva em consideração três fatores para determinar o limiar ótimo, v_W^2, que é a variância dentro das classes, v_B^2 numa variância entre as classes, e v_T^2 a variância total (OTSU, 1979). Mesmo podendo criar uma comparação entre às três variâncias, uma maneira mais simples de descobrir o limiar ótimo é através da relação entre v_B^2 e v_T^2, sendo definida como: $N = \frac{v_B^2}{v_W^2}$ E tendo como limiar ótimo: $t = arcMax(N)$. Como pode ser visto na Figura 11, onde o método Otsu tenta evidenciar objetos mais próximos.

3.8.5 ZHANG-SUEN THINNING

O último método de processamento digital de imagens, proposto por Zhang e Suen é o método de afinamento. Segundo os autores (ZHANG; SUEN, 1984), o processamento se dá

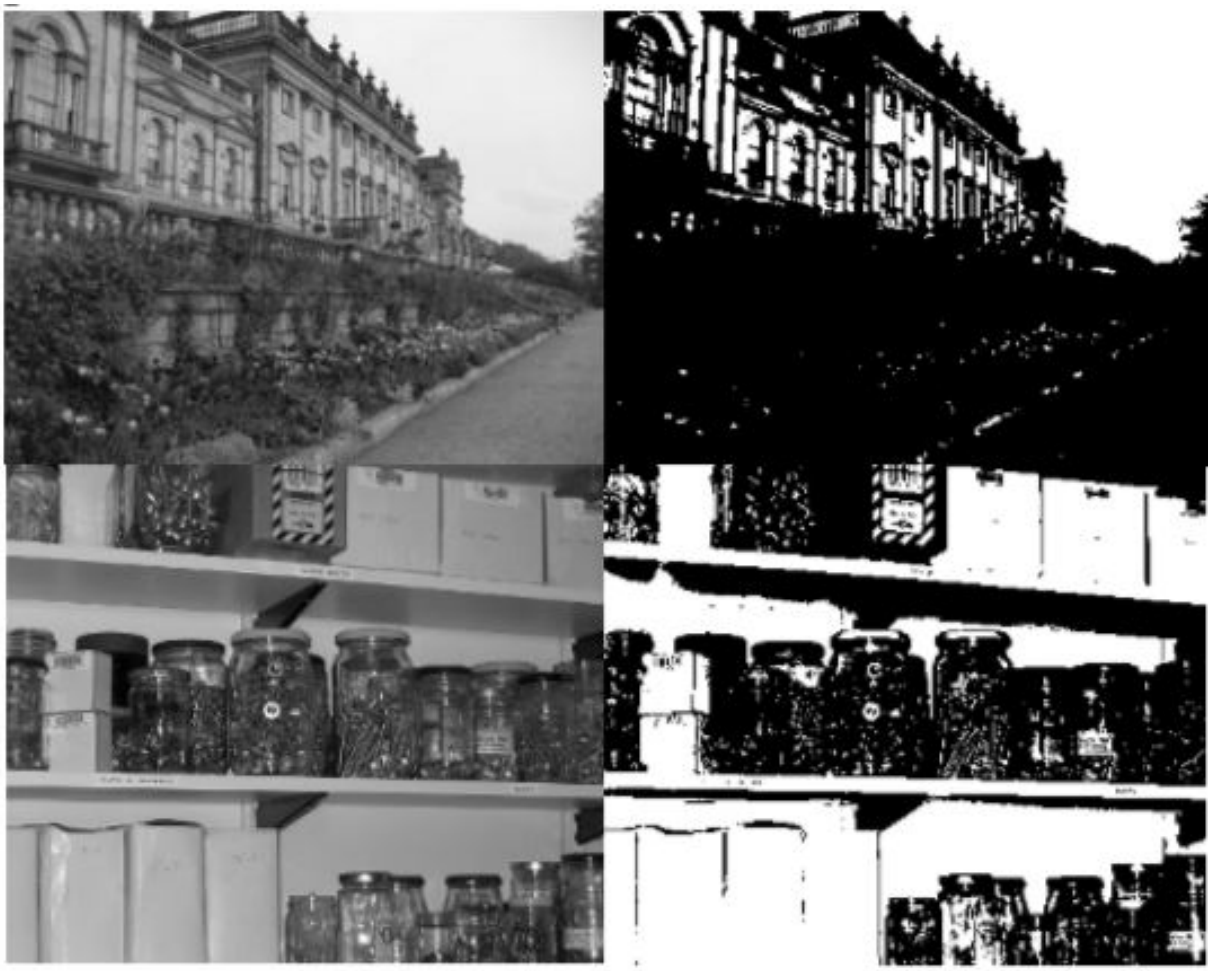

Figura 11: Exemplo de imagem em tons de cinza e seus resultados após a limiarização pelo método Otsu (TOROK, 2015)

de duas formas: deletando os pontos dos cantos superior esquerdo e inferior direito, enquanto a outra forma deleta os pontos dos cantos opostos. O objetivo deste método é criar um esqueleto de dimensão unitária, assim, preservando pixels e pontos de conexão nos cantos deletados. Seu principal objetivo é aumentar a eficiência e efetividade no reconhecimento de padrões. Isto pode ser visto na Figura 12, onde através do afinamento, encontrou-se um esqueleto da letra H.

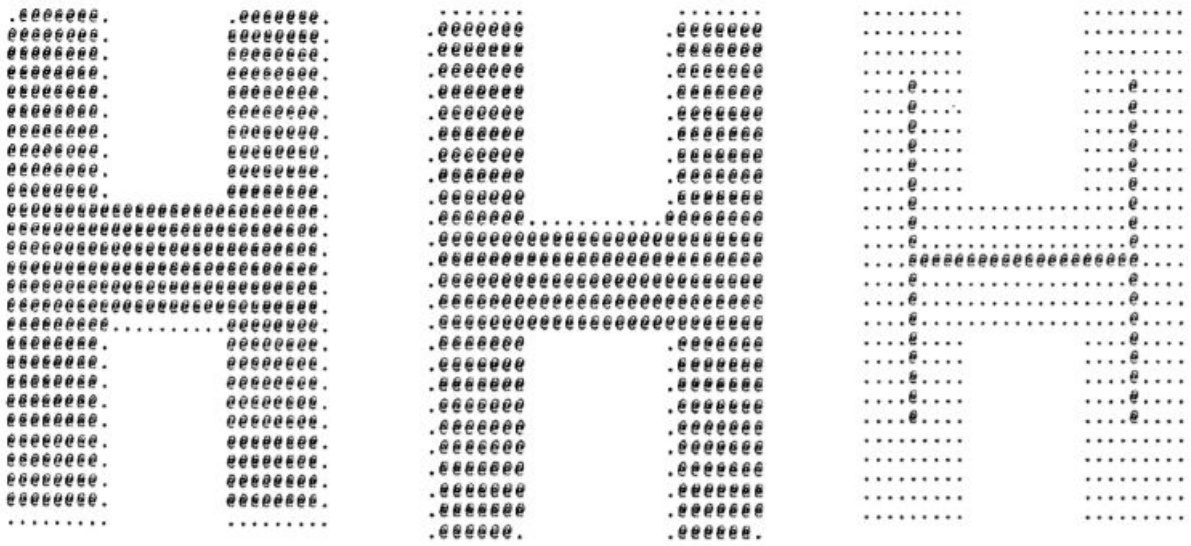

Figura 12: Resultado do afinamento da letra ”H”pelo método de Thinning

Fonte: Autores(2018)

4 METODOLOGIA

A Figura 13 mostra, através de um diagrama de blocos, a metodologia de classificação de imagens utilizada nesse trabalho.

O primeiro passo é a extração da ROI das imagens, que será melhor explicado na seção **??**, em seguida as imagens foram pré processadas, de maneira que diversas combinações de algoritmos de PDI foram aplicadas (GIMENEZ, 2015) e em seguida as imagens foram redimensionadas para 224x224 pixels. Após o redimensionamento, a imagens foram separadas em dois conjuntos, de treinamento e teste.

4.1 DESCRIÇÃO DOS DATASETS

Como parte fundamental deste trabalho, a aquisição de diferentes conjuntos de imagens foi necessária para realização dos testes. Dois conjuntos de imagens foram usados. O primeiro, cedido pela Universidade de São Paulo (USP) e o segundo, pelo Instituto Agronômico do Paraná (IAPAR).

4.1.1 DATASET IAPAR

O primeiro conjunto de imagens foi cedido pelo IAPAR, este contava com 18 bovinos diferentes, e cada bovino possuía cerca de 120 imagens. O padrão de organização das imagens

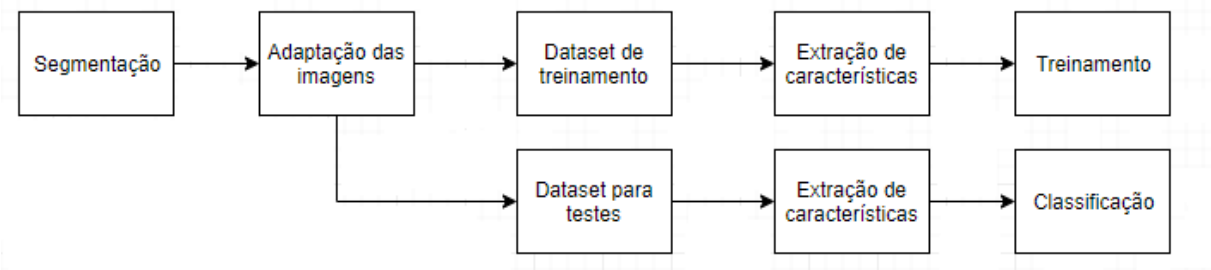

Figura 13: Diagrama de Blocos

também seguia um padrão comum a todos os bovinos. A primeira imagem possuía apenas o nome do boi e era dispensável no processamento. Logo em seguida, um conjunto de aproximadamente seis fotos eram tiradas na mesma posição e, após, o processo era repetido para um novo ângulo. A Figura 14 e 15 mostram imagens de um dos bovinos cedidos pelo IAPAR para o desenvolvimento deste trabalho.

Figura 14: Exemplo de foto com o nome do bovino cedida pelo IAPAR

Como foram fornecidas muitas imagens no mesmo ângulo pelo IAPAR, foi tomada a decisão de selecionar apenas três delas para o enfoque no focinho e espelho nasal. A partir de então, o processo de extração da ROI manual foi feito e cada imagem escolhida foi retratada nas

duas posições citadas. As características da imagem do focinho e espelho nasal foram comuns no banco de imagem fornecido tanto pela USP como IAPAR, possuindo dimensões específicas. As Figuras 15, 16 e 16 são exemplos da foto original e suas respectivas obtenções de ROI.

Figura 15: Exemplo de foto original antes da extração da ROI, cedida pelo IAPAR

Figura 16: Exemplo da figura anterior com enfoque no espelho nasal

Figura 17: Exemplo da figura anterior com enfoque no focinho

4.1.2 DATASET USP

Para o segundo conjunto de imagens, cedido pela USP, algumas diferenças foram notadas. O número de bovinos era de 51, significativamente maior que o cedido pelo IAPAR. Mas, em contrapartida, cada bovino possuía cerca de 16 fotos, um número consideravelmente menor se comparado com as 124 fotos de cada um no conjunto da IAPAR. Outro ponto importante a ser notado é que todas as cerca de 16 fotos, de cada bovino, estavam na mesma posição, isto é, o ângulo da câmera para captura das imagens não foi alterado. Logo, o processo de extração da ROI manual foi diferente, pois, ao invés três imagens, trabalhou-se com todo o dataset disponível. Nesse caso, assim como no dataset do IAPAR, cada imagem teve cortes com enfoque no focinho e espelho nasal. As Figuras 18, 19 e 20 mostram a foto de um bovino feito pela USP e suas respectivas obtenções de ROI.

4.2 OBTENÇÃO DA REGIÃO DE INTERESSE

Após a definição de quantas imagens de cada classe seriam utilizadas, bem como de quais imagens seriam utilizadas em cada dataset, as regiões de interesse das imagens foram cortadas com dimensões quadradas por meio do software GIMP2.

Após o corte das imagens obteve-se 4 datasets distintos. O dataset das imagens do focinho dos bovinos do IAPAR, o qual foi chamado de muzzle_IAPAR. O dataset das imagens do espelho nasal dos bovinos do IAPAR, o qual foi chamado de nasal_pattern_IAPAR. O dataset

Figura 18: Exemplo de foto original antes da extração da ROI, cedida pela USP

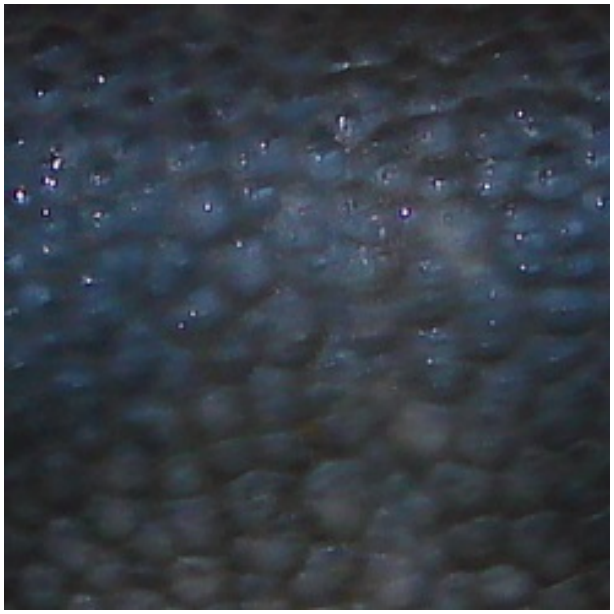

Figura 19: Exemplo da figura 18 com enfoque no espelho nasal

das imagens do focinho dos bovinos da USP, o qual foi chamado de muzzle_USP. E o dataset das imagens do espelho nasal dos bovinos da USP, o qual foi chamado de nasal_pattern_USP.

4.3 GERAÇÃO DOS DATASETS PRÉ PROCESSADOS

Nessa seção são apresentados os experimentos realizados com diferentes combinações de algotimos de PDI. A partir dos resultados obtidos e documentados por Gimenez (2015), os datasets fornecidos pelo IAPAR e pela USP, após a obtenção da região de interesse manualmente, foram pré processados, submetidos a deferentes combinações de algoritmos de PDI e armazenados em arquivos HDF5. O controle dos algoritmos de PDI utilizados foi feito através de um arquivo CSV mostrado na Figura 21. Os experimentos são explicados abaixo.

Figura 20: Exemplo da figura 18 com enfoque no focinho

	A	B	C	D	E	F	G	H	I	J
1	StudyCase1	StudyCase2	StudyCase3	StudyCase4	StudyCase5	StudyCase6	StudyCase7	StudyCase8	StudyCase9	StudyCase10
2		CLAHE	CLAHE	Gray	Gray	Gray	Gray	Gray	Gray	Gray
3			Sobel		CLAHE	CLAHE	thinning	CLAHE	CLAHE	CLAHE
4						Sobel		thinning	Sobel	Sobel
5						Otsu			thinning	Otsu
6										thinning

Figura 21: Arquivo de entrada utilizado pelo script gerador do arquivo HDF5

4.3.1 EXPERIMENTO 1

Nesse experimento as imagens não foram alteradas e nem pré processadas. O objetivo desse experimento foi determinar a eficiência dos algoritmos de extração de características e classificação nas imagens originais, para que posteriormente fosse possível quantificar a influência positiva ou negativa de cada algoritmo de PDI na porcentagem de acerto. Na Figura 22 são mostradas duas imagens desse experimento, dos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

4.3.2 EXPERIMENTO 2

Nesse experimento o algoritmo CLAHE foi aplicado às imagens de maneira que o contraste em cada imagem foi aumentado, conforme explicado na seção 3.8.1. Na Figura 23 são mostradas duas imagens desse experimento, dos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

Figura 22: Imagens do experimento 1 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

Figura 23: Imagens do experimento 2 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

4.3.3 EXPERIMENTO 3

Nesse experimento os algoritmoa CLAHE e Sobel foram aplicados às imagens, respectivamente, de maneira que o contraste em cada imagem foi aumentado e as bordas detectadas, conforme explicado na seções 3.8.1 e 3.8.2. Na Figura 24 são mostradas duas imagens desse experimento, dos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

4.3.4 EXPERIMENTO 4

Nesse experimento as imagens foram convertidas para a escala de cinza conforme explicado na seção 3.8.3. Na Figura 25 são mostradas duas imagens desse experimento, dos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

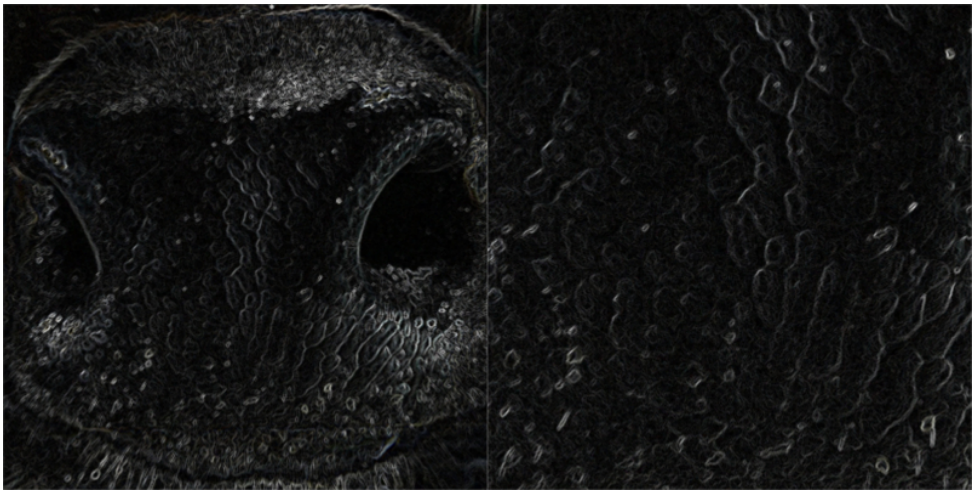

Figura 24: Imagens do experimento 3 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

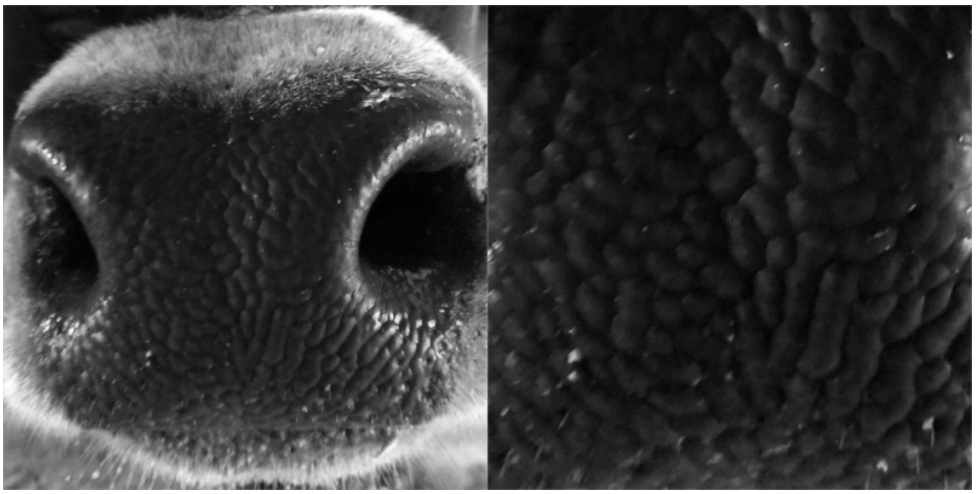

Figura 25: Imagens experimento 4 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

4.3.5 EXPERIMENTO 5

Nesse experimento as imagens foram convertidas para a escala de cinza e o algoritmo CLAHE foi aplicado, de maneira que o contraste em cada imagem cinza foi aumentado, conforme explicado na seções 3.8.3 e 3.8.1. Na Figura 26 são mostradas duas imagens desse experimento, dos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

4.3.6 EXPERIMENTO 6

Nesse experimento as imagens foram convertidas para a escala de cinza e o algoritmos CLAHE, Sobel e Otsu foram aplicados, de maneira que o contraste em cada imagem cinza foi aumentado, as bordas detectadas e os elementos do fundo da imagem foram separados

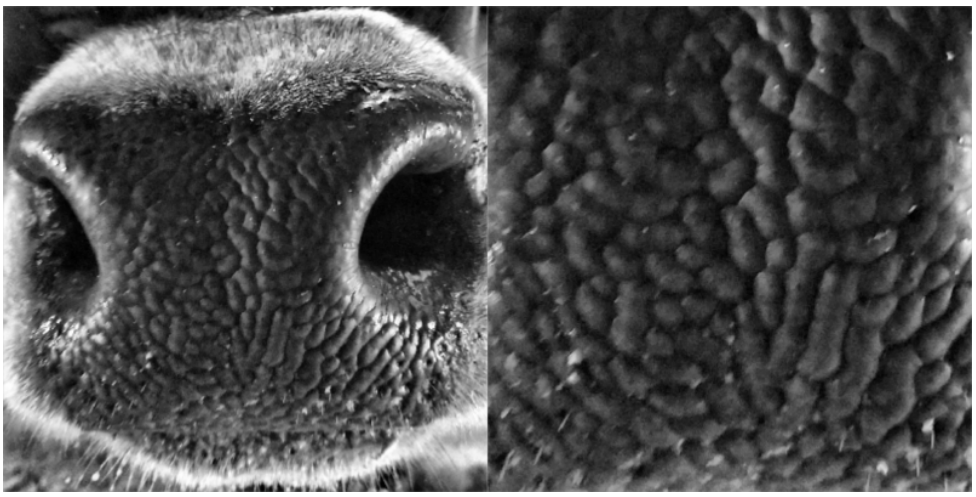

Figura 26: Imagens do experimento 5 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

dos elementos da frente, conforme explicado na seções 3.8.3, 3.8.1, 3.8.2 e 3.8.4. Na Figura 27 são mostradas duas imagens desse experimento, dos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

Figura 27: Imagens do experimento 6 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

4.3.7 EXPERIMENTO 7

Nesse experimento as imagens foram convertidas para a escala de cinza e o algoritmo thinning foi aplicado de maneira a gerar a esqueletização de cada imagem cinza, conforme explicado na seções 3.8.3 e 3.8.5. Na Figura 28 são mostradas duas imagens desse experimento, dos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

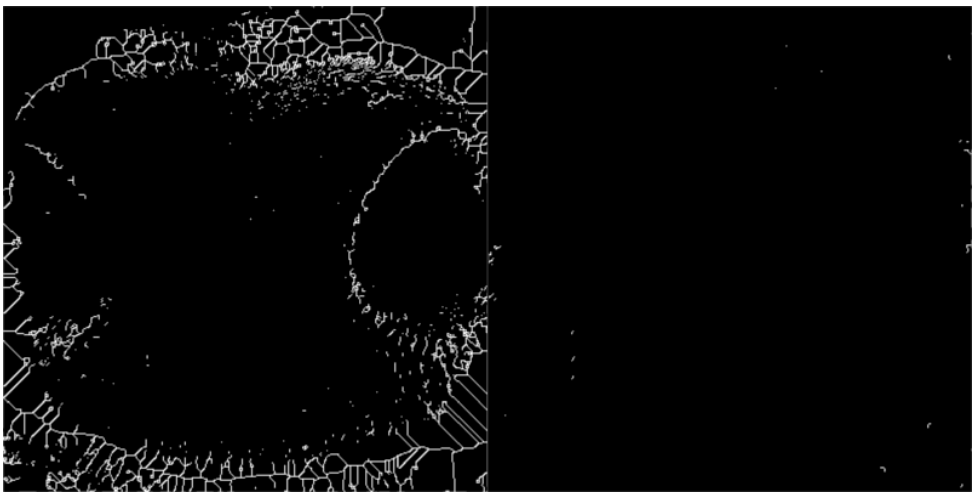

Figura 28: Imagens do experimento 7 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

4.3.8 EXPERIMENTO 8

Nesse experimento as imagens foram convertidas para a escala de cinza e o algoritmos CLAHE e thinning foram aplicados de maneira que o contraste em cada imagem cinza foi aumentado e em seguida a imagem foi esqueletizada, conforme explicado na seções 3.8.3, 3.8.1 e 3.8.5. Na Figura 29 são mostradas duas imagens desse experimento, dos datasets muzzle_IAPAR e nasal_pattern_IAPAR.

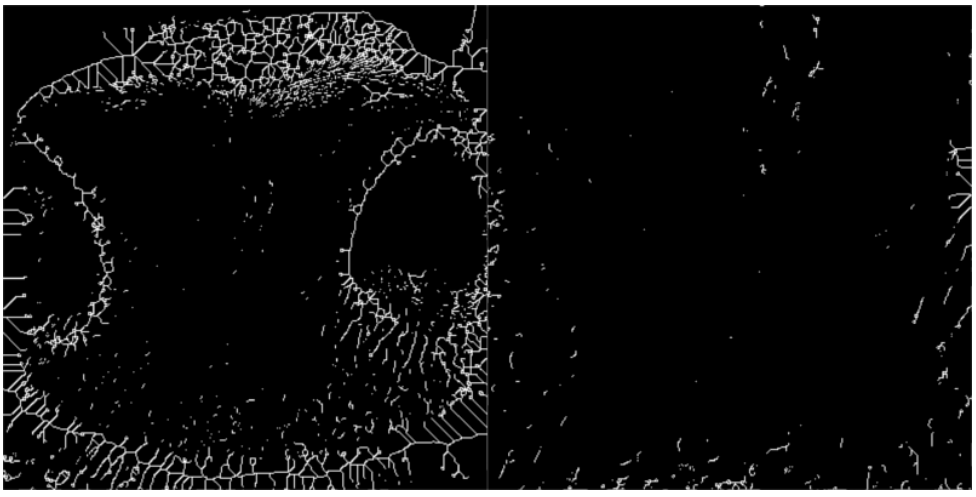

Figura 29: Imagens do experimento 8 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

4.3.9 EXPERIMENTO 9

Nesse experimento as imagens foram convertidas para a escala de cinza e o algoritmos CLAHE, Sobel e thinning foram aplicados, de maneira que o contraste em cada imagem cinza

foi aumentado, as bordas foram detectadas e a imagem foi esqueletizada, conforme explicado na seções 3.8.3, 3.8.1, 3.8.2 e 3.8.5. Na Figura 30 são mostradas duas imagens desse experimento, dos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

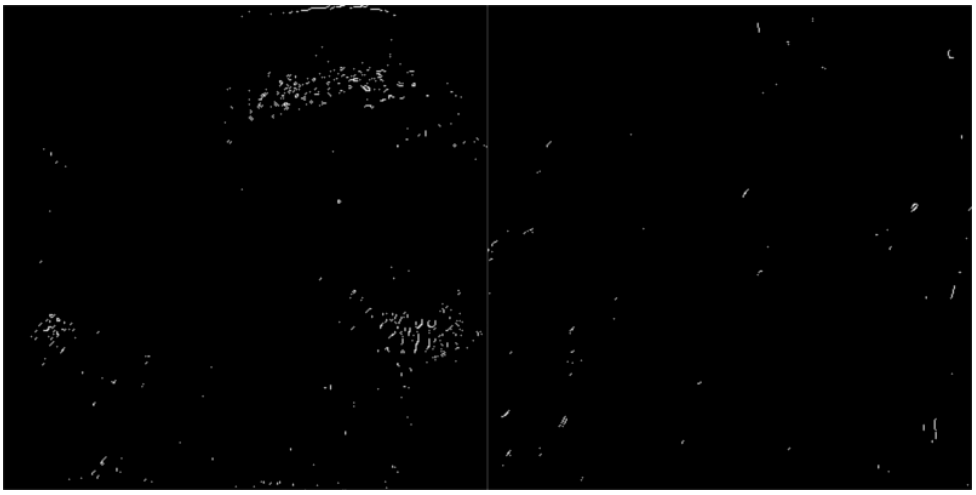

Figura 30: Imagens do experimento 9 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

4.3.10 EXPERIMENTO 10

Nesse experimento as imagens foram convertidas para a escala de cinza e o algoritmos CLAHE, Sobel, Otsu e thinning foram aplicados, de maneira que o contraste em cada imagem cinza foi aumentado, as bordas foram detectadas, os elementos do fundo da imagem foram separados dos elementos da frente e a imagem foi esqueletizada, conforme explicado na seções 3.8.3, 3.8.1, 3.8.2, 3.8.4 e 3.8.5. Na Figura 31 são mostradas duas imagens desse experimento, dos coonjuntos muzzle_IAPAR e nasal_pattern_IAPAR.

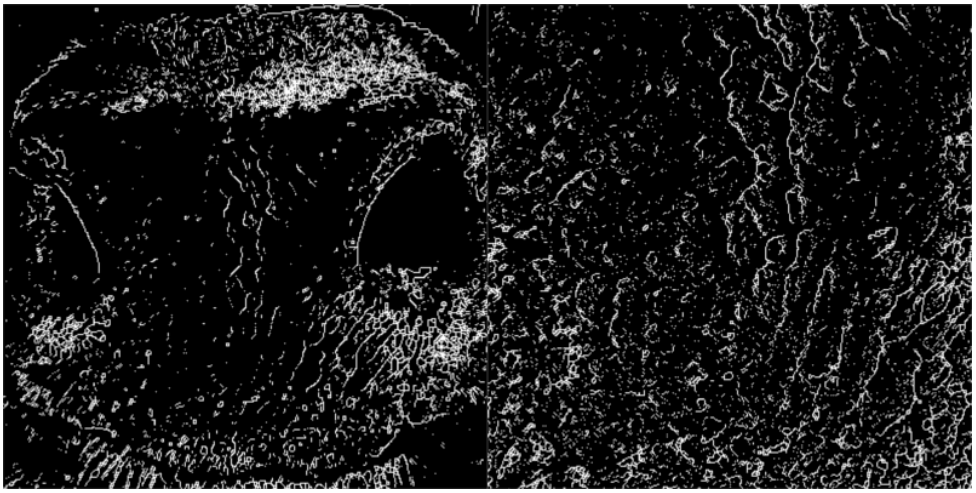

Figura 31: Imagens do experimento 10 pertencentes aos conjuntos muzzle_IAPAR e nasal_pattern_IAPAR respectivamente

5 RESULTADOS

5.1 ESCOLHA DOS PARÂMETROS

Os algoritmos de extração de características e classificação utilizados nesse trabalho exigiram uma escolha inicial de uma série de parâmetros, bem como o posterior refinamento dos mesmos de acordo com os testes realizados e os resultados obtidos.

O primeiro parâmetro necessário, antes que extração de características ou classificação pudesse ser feita, foi a porcentagem de divisão entre o dataset de treino e o dataset de teste. Uma convenção bastante aceita e comumente usada na comunidade de pesquisadores em Machine Learning, para datasets pequenos como no nosso caso, é dividir o conjunto de dados em 70% para treino e 30% para teste (GHOLAMI et al., 2015), a partir disso nós definimos um conjunto de porcentagens variando entre 5% e 40%, com divisões de 5%. Essas divisões foram usadas nos testes com os algoritmos de extração de características CNN e PCA e o algoritmo de classificação SVM. A partir dos testes feitos em todos os casos de estudos, nós definimos como uma divisão ótima, capaz de generalizar bem os resultados obtidos no dataset de treino para o dataset de teste a divisão de 80% para o datadset de treino e 20% para o dataset de teste, a Figura 47 mostra os resultados obtidos na avaliação desse parâmetro para o dataset do IAPAR e o experimento 1.

Em se tratando dos algoritmos de extração de características, o primeiro parâmetro que precisou ser testado foi a variância retida pelo algoritmo PCA. Conforme explicado na seção 3.2, esse algoritmo realiza uma transformação linear entre os espaço vetorial original dos conjuntos de treino e teste, para um espaço de menor dimensão, para tanto, o algoritmo escolhe uma base na direção de maior variância do conjunto que está sendo analisado. O número de vetores que compõem essa base são dependentes da variância retida pelo algoritmo, conforme mostrado na figura 44 para o dataset muzzle_IAPAR e o caso de estudo 1. A partir dos testes realizados em todos os datasets e para todos os casos de estudos, com variâncias retidas variando entre 50% e 95%, com incrementos de 5%, nós definimos a variância retida com o melhor custo benefício entre taxa de acerto e tempo de computação como 85%.

O algoritmo SVM, conforme explicado na seção 3.1 exigiu a definição de um kernel, o qual iria determinar a distribuição na qual os dados seriam projetados. A partir dos testes realizados com todas as divisões entre dataset de treino e de teste, e todas as variâncias retidas definidas, conforme explicado acima, e com os kernels linear, polinomial, RBF e sigmoid, definiu-se que, para a distribuição apresentadas pelos conjuntos de imagens usados nesse trabalho, o kernel linear foi o mais adequado. Um exemplo de testes realizados com variação de kernel e variação de graus para o kernel polinomial pode ser vistos nas tabelas 1 e 2 para o dataset muzzle_IAPAR e o caso de estudo 1.

No casos dos algoritmos SIFT e SURF, o parâmetro que precisou ser escolhido foi a quantidade de imagens de referência para cada uma das imagens de entrada. Os testes nesses casos foram realizados sobre todas as imagens de cada dataset.

5.2 RESULTADOS OBTIDOS NO COMPUTADOR

Os resultados discutidos nessa seção foram obtidos em um computador Dell Inspiron 15 5000, com processador Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90GHz e 8.00 GB de memória RAM. O sistema operacional utilizado foi Windows 10 Pro, a linguagem utilizada para escrever o código de cada teste foi python3 e os frameworks utilizados foram OpenCV 3.4.1 e Keras 2.1.5 com Theano 1.0.2 como background.

Notou-se durante a realização dos testes que os conjuntos com imagens do focinho inteiro do bovino apresentavam índices de acerto maiores que os índices apresentados pelos conjuntos com imagens do espelho nasal, por isso nesta seção são mostrados apenas os resultados obtidos para conjuntos de imagens do focinho inteiro dos bovinos.

5.2.1 PCA E SVM

Nessa subseção são mostrados os índices de acerto bem como os tempos de computação na aplicação do algoritmo PCA, como extrator de características, seguido pela aplicação do algoritmo SVM como classificador. Conforme explicado na sessão 5.1, os testes foram realizados com todas as combinações de casos de estudos, divisões do conjunto de treino e do conjunto de teste, e porcentagens de variância retida pré definidas. Aqui são mostrados os resultados obtidos para cada um dos 10 casos de estudos utilizando-se os parâmetros ótimos de divisão dos conjuntos de treino e teste, e de variância retida pelo algoritmo PCA, valores estes encontrados durante os testes. Nos resultados mostrados o único pré processamento feito nas imagens de cada dataset foi a extração da ROI e o redimensionamento das mesmas para 224x224 pixels.

5.2.1.1 DATASET IAPAR

O conjunto com imagens do focinho inteiro, cedido pelo IAPAR, possuía imagens do bovino em várias posições diferentes, esse fator dificultou o aprendizado do algoritmo classificador, SVM com kernel linear nesse caso, e fez com que os índices de acerto do algoritmo, aplicado a esse conjunto, fossem menores que os índices de acerto do algoritmo aplicado as imagens cedidas pela USP. Nesse último caso, as imagens também eram do focinho inteiro dos bovinos.

Apesar disso, nessa combinação de algoritmos, o conjunto do IAPAR apresentou uma tendência que também apareceu em todos os demais testes e combinações de algoritmos.

Conforme mostrado na Figura 32, a aplicação de algoritmos de PDI trás pouca, ou nenhuma melhora no índice de acerto do algoritmo de classificação, o que se nota na verdade é na maioria dos casos, a aplicação de algoritmos de PDI causa uma piora de performance. Pode-se ver na Figura 32 que o melhor índice de acerto, 91,27%, é alcançado no caso de estudo 1. Ademais, a Figura 33 mostra que, no caso de imagens coloridas, os algoritmos de PDI além de causarem piora no índice de acerto, também exigem mais tempo de computação dos algoritmos de extração de características e classificação.

Um caso de estudo interessante de ser citado aqui é o caso de estudo 4. Nas Figuras 32 e 33 pode-se observar que a conversão das imagens para a escala de cinza trás um ganho significativo no tempo de computação do algoritmo, e o índice de acerto fica muito próximo ao índice máximo, alcançado no caso de estudo 1, o que seria um fator crítico em um sistema de identificação de bovinos em tempo real.

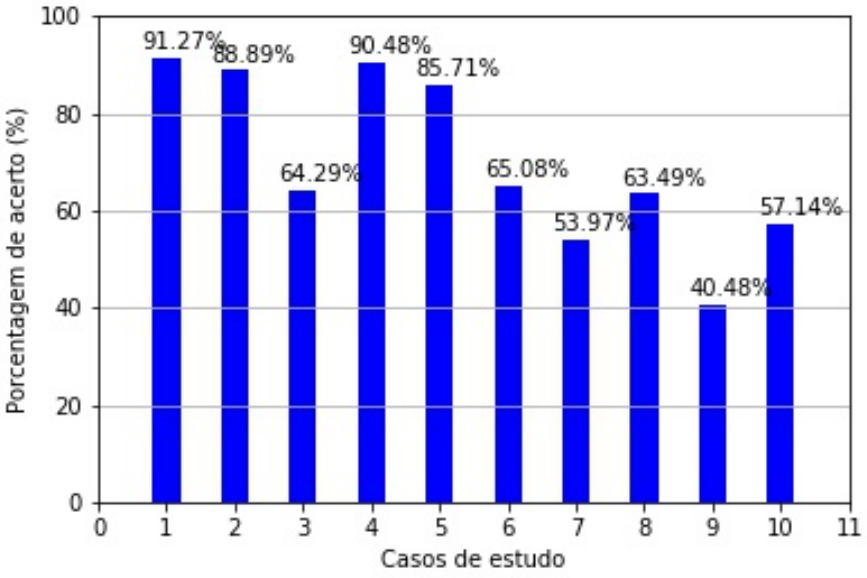

Figura 32: Porcentagem e acerto para cada caso de estudo com 85% de variância retida e 20% do dataset para teste

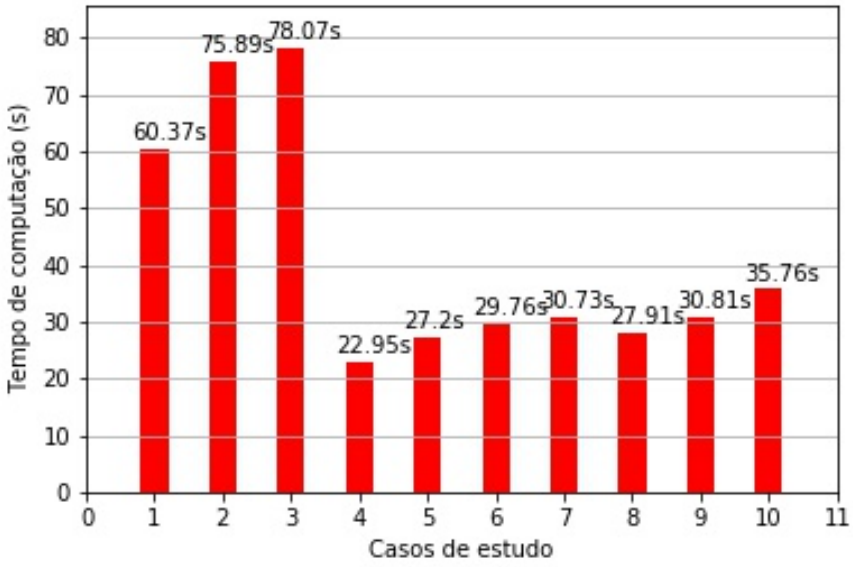

Figura 33: Tempo de computação para cada caso de estudo com 85% de variância retida e 20% do dataset para teste

5.2.1.2 DATASET USP

O conjunto de imagens do focinho inteiro dos bovinos, cedidas pela USP, era composto por imagens do bovino em uma mesma posição, fato esse que facilitou o aprendizado do algoritmo de classificação. O kernel usado, mais uma vez foi o linear, que se mostrou mais apropriado que os demais kernels, tanto para a distribuição apresentada pelas imagens do IAPAR quanto pada a distribuição apresentada pelas imagens da USP.

Na comparação entre casos de estudos, pode-se observar na Figura 34 que o melhor índice de acerto para os casos de estudo 1, 2 e 5 foram 100,0%, o que torna dispensável, mais uma vez, a aplicação de algoritmos de PDI.

O caso de estudo 4, assim como no conjunto do IAPAR, apresentou um ganho significativo no tempo de computação exigido pelos algoritmos de extração de características e classificação, e uma taxa de acerto muito próxima da máxima, conforme mostrado nas Figuras 34 e 35.

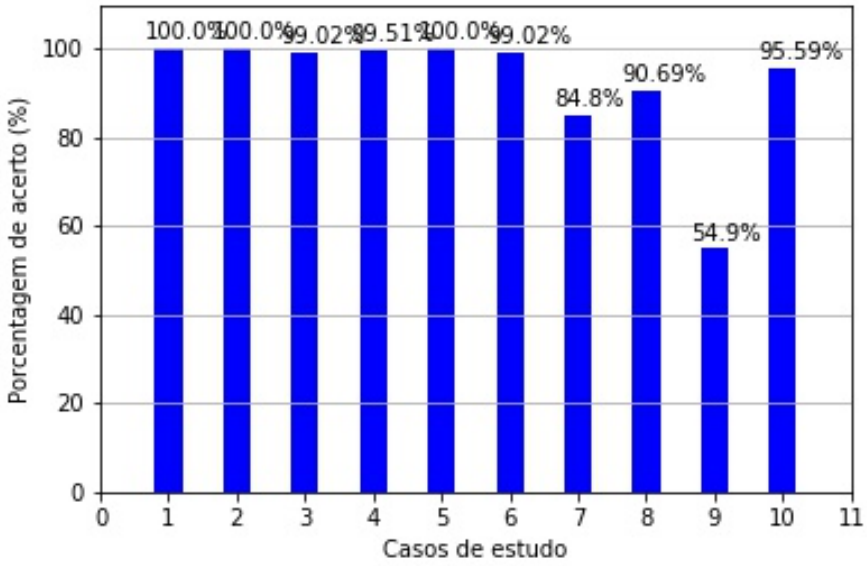

Figura 34: Porcentagem e acerto para cada caso de estudo com 85% de variância retida e 20% do dataset para teste

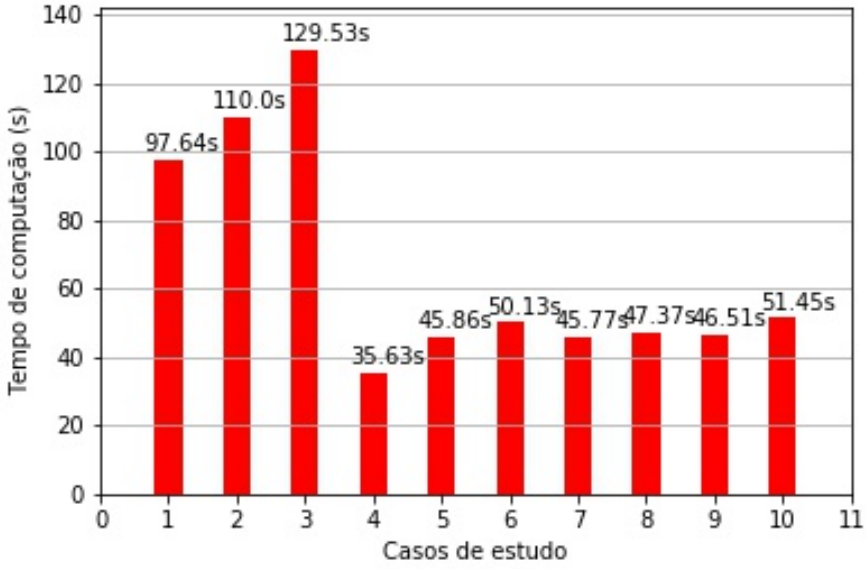

Figura 35: Tempo de computação para cada caso de estudo com 85% de variância retida e 20% do dataset para teste

5.2.2 SIFT

Nessa subsessão são apresentados os resultados obtidos no computador usando-se o algoritmo SIFT como extrator de características e o algoritmo kNN como classificador.

A imagem nesse caso, foi classificada pelo algoritmo kNN conforme o número de corresponência entre os key points extraídos da mesma pelo algoritmo SIFT, e os key points extraídos das imagens de referência. Os testes foram realizados com o número de imagens de referência variando de um até cinco, e observou-se uma tendência de crescimento no índice de acerto do algoritmo de classificação conforme o número de imagens de referência aumentava. Por isso, nessa subseção são mostrados os resultados obtidos para cada caso de estudo utilizando-se cinco imagens de referência.

5.2.2.1 DATASET IAPAR

No conjunto de imagens cedidas pelo IAPAR, a técnica descrita nessa subseção atingiu um índice de acerto inferior ao índice atingido pela técnica descrita na subseção 5.2.1. Ademais, o índice de acerto também foi inferior em comparação com o índice de acerto abtido quando esta mesma técnica foi aplicada ao conjunto de imagens cedidos pela USP.

Mais uma vez pode-se observar que, devido as imagens de cada bovino aparecerem em posições diferentes, o algoritmo tem uma dificuldade maior em classificar corretamente, deficuldade essa medida pelo índice de acerto do mesmo, e que o uso de um conjunto de imagens com essa característica exige um refinamento maior nos parâmetros dos algoritmos de extração de características e classificação.

A Figura 36 mostra mais uma vez que, a aplicação de algoritmos de PDI não trás um ganho significativo no índice de acerto do algoritmo de classificação, e que nesse caso, acarreta um aumento significativo no tempo de computação, confirme mostra a Figura 37, e que poderia até mesmo tornar a escolha dessa técnica inviável em um sistema de classificação de bovinos em tempo real.

A partir do exposto acima, mais uma vez podemos depreender que a opção ótima de casos de estudos foram os casos de estudo 1 e 4, onde nenhum algoritmo de PDI foi aplicado ou então apenas a conversão para a escala de cinza foi realizada.

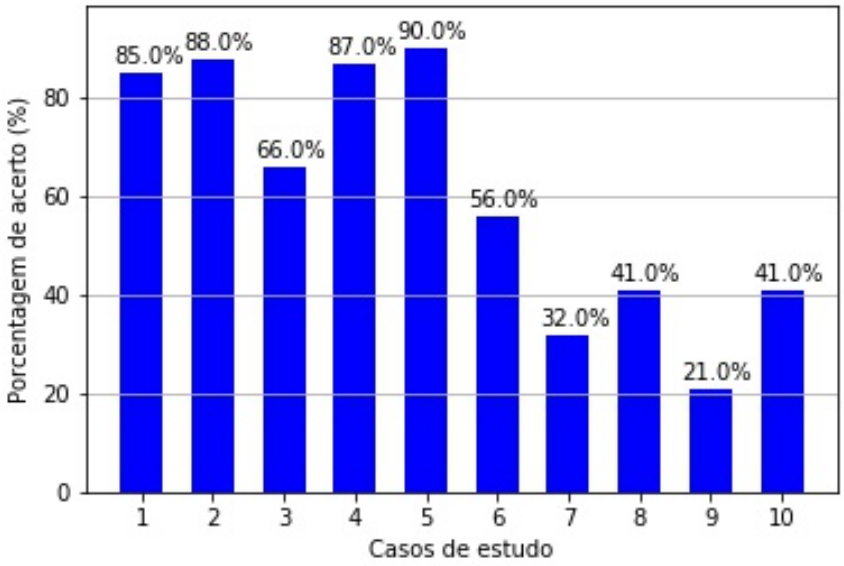

Figura 36: Porcentagem de acerto para cada caso de estudo com 5 imagens de referência

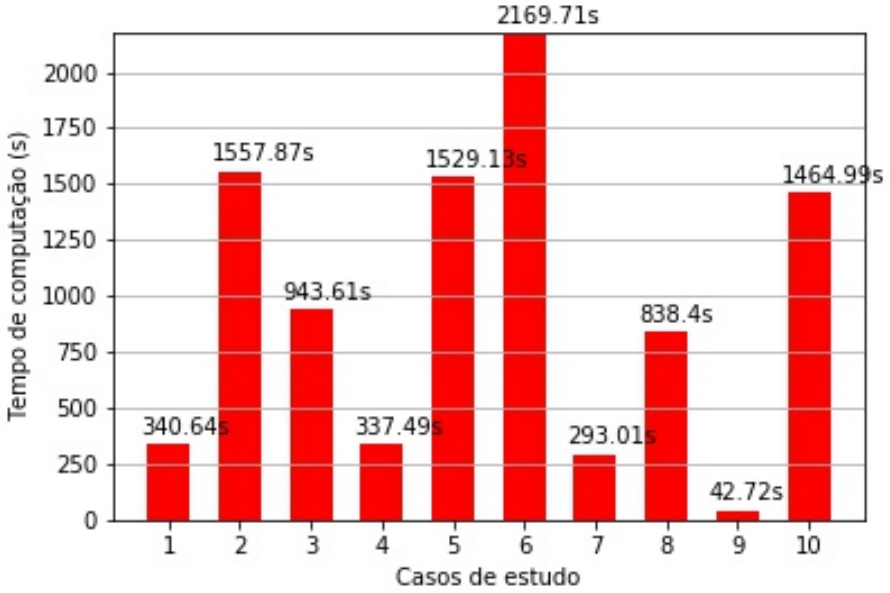

Figura 37: Tempo de computação para cada caso de estudo com 5 imagens de referência

5.2.2.2 DATASET USP

O conjunto de imagens cedidos pela USP, quando submetido a esta técnica, mostrou mais uma vez o quanto a distribuição das imagens influencia no índice de acertos. No caso desse conjunto, onde os bovinos estão todos na mesma posição, mais uma vez índices de 100% de acerto foram atingidos para alguns dos casos de estudo, conforme mostra a Figura 38.

A Figura 39 mostra que, para a distribuição apresentada pelo conjunto de imagens da USP, as mesmas têndências apresentadas pelo conjunto de imagens do IAPAR podem ser verificadas. Nesse caso, pode-se observar novamente que os casos de estudo ótimos são os casos de estudo 1 e 4, e que a aplicação de algoritmos de PDI não trás vantagens significativas, e alguns casos trás grandes desvantagens.

Pode-se observar nas figuras 38 e 39 que a técnica descrita nesta seção pode ser considerada uma escolha viável e razoável para um sistema de classificação de bovinos em tempo real, e que exija confiabilidade, se a distribuição de imagens apresentar o bovino na mesma posição em todas as imagens.

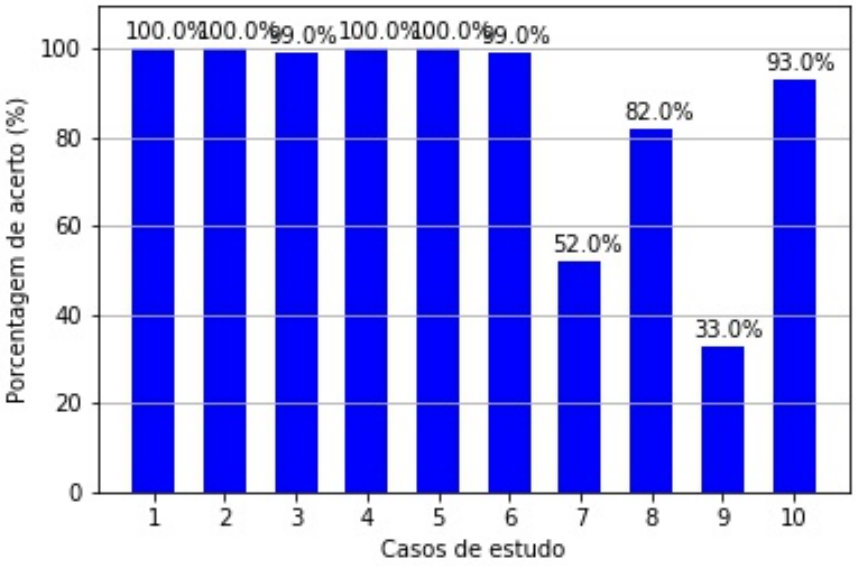

Figura 38: Porcentagem de acerto para cada caso de estudo com 5 imagens de referência

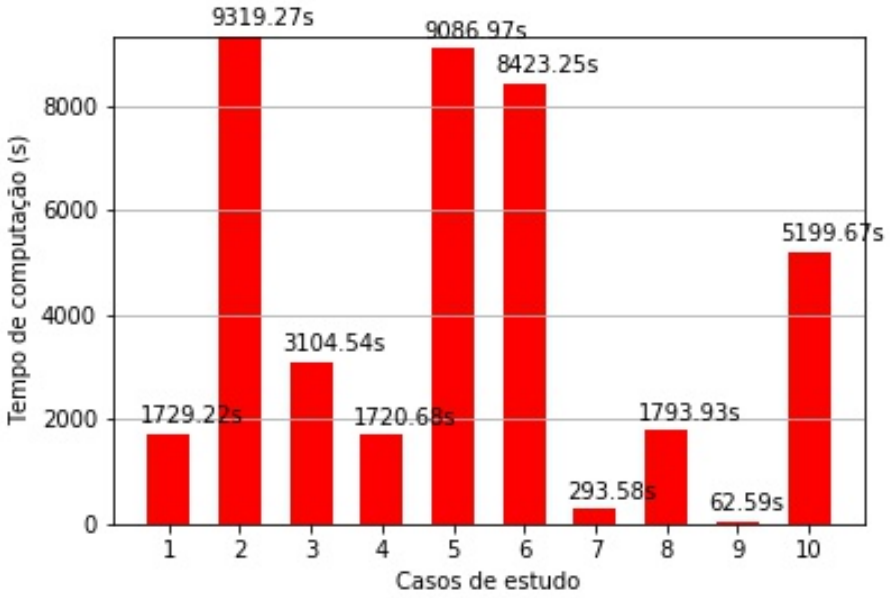

Figura 39: Tempo de computação para cada caso de estudo com 5 imagens de referência

5.2.3 SURF

Nessa subsessão são apresentados os resultados obtidos no computador usando-se o algoritmo SURF como extrator de características e o algoritmo kNN como classificador.

O procedimento utilizado nesta técnica foi o mesmo procedimento descrito na subseção 5.2.2, a imagem foi classificada pelo algoritmo kNN conforme o número de corresponência entre os key points extraídos da mesma pelo algoritmo SURF, e os key points extraídos das imagens de referência. Os testes foram realizados com o número de imagens de referência variando de um até cinco, e assim como na subseção 5.2.2, observou-se uma tendência de crescimento no índice de acerto do algoritmo de classificação conforme o número de imagens de referência aumentava.

Nessa subseção são mostrados os resultados obtidos para cada caso de estudo utilizando-se cinco imagens de referência.

5.2.3.1 DATASET IAPAR

A técnica descrita nesta subseção, apesar de apresentar um procedimento muito semelhante ao da técnica descrita na subseção 5.2.2, quando aplicada ao conjunto de imagens do IAPAR, apresentou índices de acerto inferiores, conforme mostrado na figura 40.

Conforme descrito na seção 3.4, o algoritmo SURF foi resultado de uma pesquisa que buscava por algoritmo extrator de caraterísticas mais rápido que o algoritmo SIFT. Essa característica pode ser observada na 37 e 41. Contudo para um sistema de classificação de bovinos que exige confiabilidade, essa técnica não se mostrou vantajosa a priori.

A tendência apresentada pelos algoritmos de PDI em todas as técnicas anteriores, pode ser observada mais uma vez para esta técnica. E novamente, os casos de estudos ótimos, em índice de acerto e tempo de computação, são os casos de estudo 1 e 4.

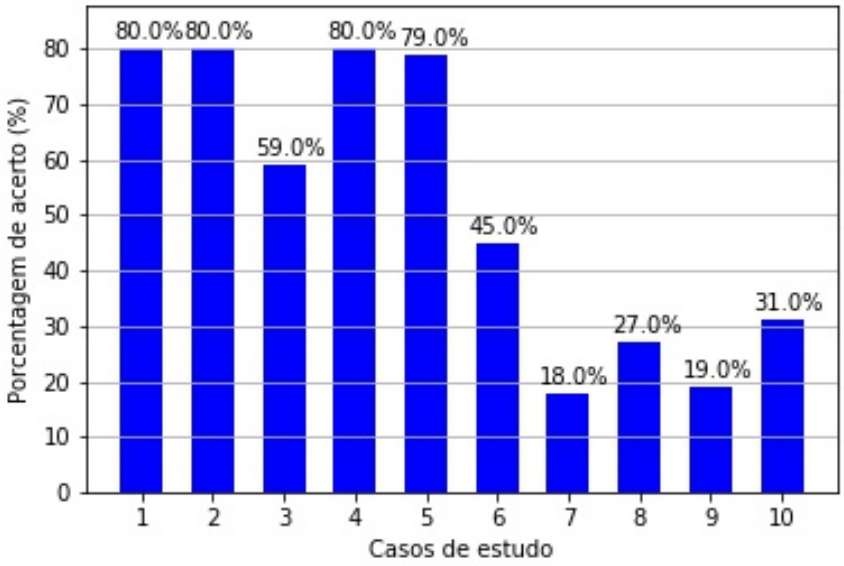

Figura 40: Porcentagem de acerto para cada caso de estudo com 5 imagens de referência

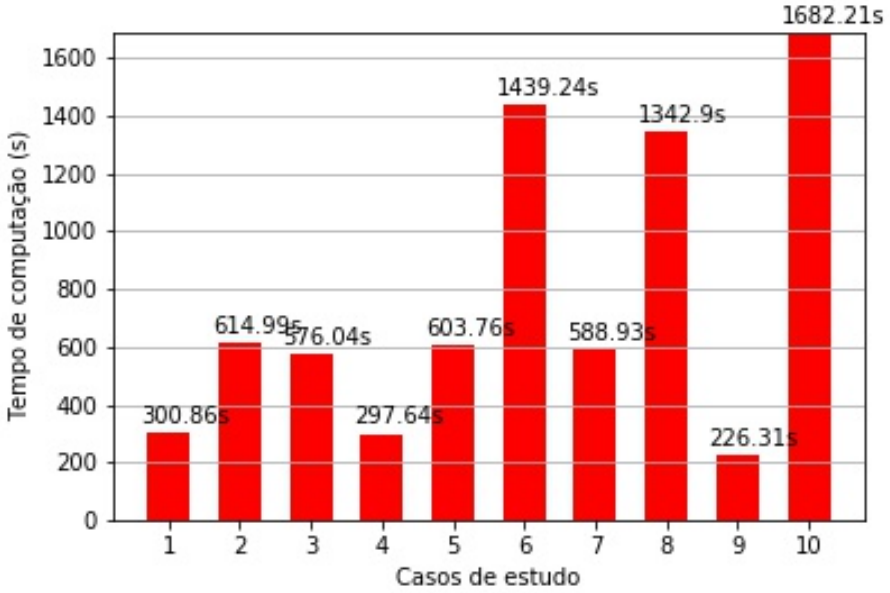

Figura 41: Tempo de computação para cada caso de estudo com 5 imagens de referência

5.2.3.2 DATASET USP

No caso do conjunto de imagens cedidos pela USP, mais uma vez alguns casos de estudos com 100% de acerto, devido a distribuição homogênea das imagens, conforme mostrado na figura 42. No entanto, a utilização desse técnica mostrou uma vantagem significativa sobre a técnica descrita na subseção 5.2.2 em se tratando de tempo de computação.

Os casos de estudos ótimos, mais uma vez foram os casos de estudo 1 e 4, porém a comparação das figuras 39 e 43 mostra que a técnica que utiliza o algoritmo SURF como extrator de características é aproximadamente duas vezes mais rápida que a técnica que utiliza o algoritmo SURF. Por isso, essa seria uma escolha mais viáel para a construção de um sistema de classificação de bovinos em tempo real, no caso da distribuição das imagens ser igual a distribuição do conjunto de imagens da USP.

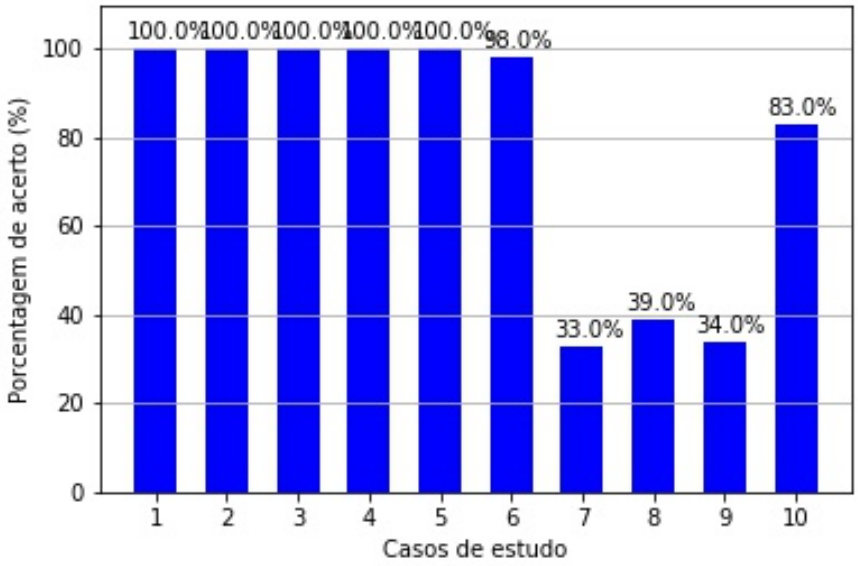

Figura 42: Porcentagem de acerto para cada caso de estudo com 5 imagens de referência

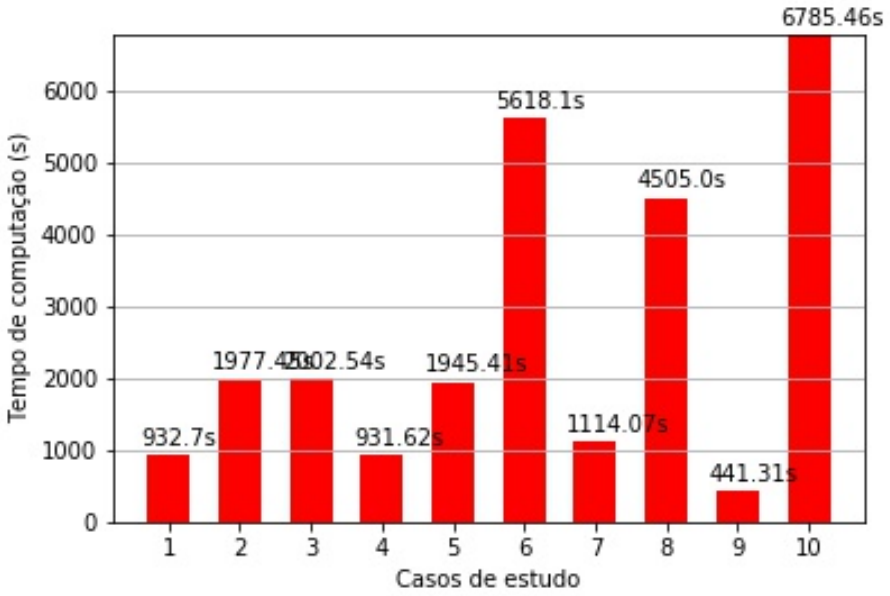

Figura 43: Tempo de computação para cada caso de estudo com 5 imagens de referência

5.2.4 CNN, PCA E SVM

Nessa subsessão são mostrados os resultados obtidos utilizando-se uma CNN em conjunto com o algoritmo PCA como extratores de características e o algoritmo SVM como classificador.

A arquitetura de CNN utilizada nesse trabalho foi a VGG16, e o conceito de Transfer Learning foi aplicado, conforme explicado na seções 3.6 e 3.7. Uma rede neural pré treinada, retirada do banco de dados Imagenet foi utilizada, essa rede foi treinada em um conjunto de aproximadamente 1,2 milhões de imagens divididas em 1000 classes. Apenas as camadas finais de classificação da rede neural em questão foram retiradas, de maneira apenas a CNN foi mantida, e a saída desta, para uma determinada imagem de entrada, foi usada como entrada para o algoritmo PCA.

5.2.4.1 DATASET IAPAR

O conjunto de imagens do IAPAR, apresentou em todas as técnicas anteriores, índices de acerto inferiores aos índices de acerto apresentados pelo conjunto de imagens da USP. Fato este que é uma consequência do bovino aparecer em posições diferentes nesse conjunto.

Na aplicação da técnica descrita nesta seção, a tendência nos índices de acerto se manteve, e o conjunto de imagens da USP, aprensentou índices melhores novamente. Contudo, os índices apresentados para o conjunto de imagens do IAPAR foram maiores que em todas as outras técnicas, conforme mostrado na figura 47.

Essa técnica foi aplicada apenas às imagens coloridas, tendo em vista que a arquitetura VGG16 espera uma imagem colorida como entrada. Os valores utilizados para a porcentagem de variância retida pelo algoritmo PCA e porcentagem de conjunto de teste, foram 80% e 20% respectivamente, conforme explicado na seção 5.1. Esses parâmetros foram obtidos a partir de testes, como os mostrados nas figuras 44, 45 e 46, onde pode-se observar que o número de autovetores para o algoritmo PCA, assim como o tempo de computação, possuem uma relação direta com a porcentagem de variância retida pelo algoritmo PCA.

As figuras 47 e 48 também exemplificam como a porcentagem de conjunto de teste foi definida. No caso da técnica descrita nesta subseção, pode-se observar que os índices de acerto apresentaram valor máximo, ou muito próximo ao máximo para todas as porcentagens de divisão testadas.

Essa foi a única técnica, dentre as técnicas testadas, que apresentou índice de acerto

100% para o conjunto de imagens do IAPAR. O que mostra que essa é a técnica com maior capacidade de generalização, e sugere que, no projeto de um sistema de classificação de bovinos que exija confiabilidade, essa é a melhor técnica, dentre as técnicas testadas nesse trabalho.

A figura 48 mostra que o tempo de computação exigido é maior para esta técnica, quando comparada a técnica apresentada na seção 5.2.1 por exemplo. Contudo, como o tempo apresentado no gráfico das figuras 33 e 48 engloba o tempo necessário para extração de características e treinamento, a utilização de uma CNN, como extratora de características, em um sistema de classificação de bovinos em tempo real, talvez seja uma opção razoável, uma vez que o tempo crítico nesse caso seria apenas o tempo de classificação. Como esse trabalho não entrou no nível de detalhamento necessário para essa conclusão, este é um ponto que deixa-se em aberto para pesquisas futuras.

A figura 47 mostra que as taxas de 100% de acerto são alcançadas no caso de estudo 1, de maneira que, mais uma vez, a opção ótima é este caso de estudo. Tendo em vista que a aplicação de algoritmos de PDI não trás nenhum benefício significativo.

O conjunto de imagens da USP, assim como nas demais técnicas, apresentou indíces de acerto melhores que os índices apresentados pelo conjunto de imagens do IAPAR, e por isso os gráficos foram omitidos.

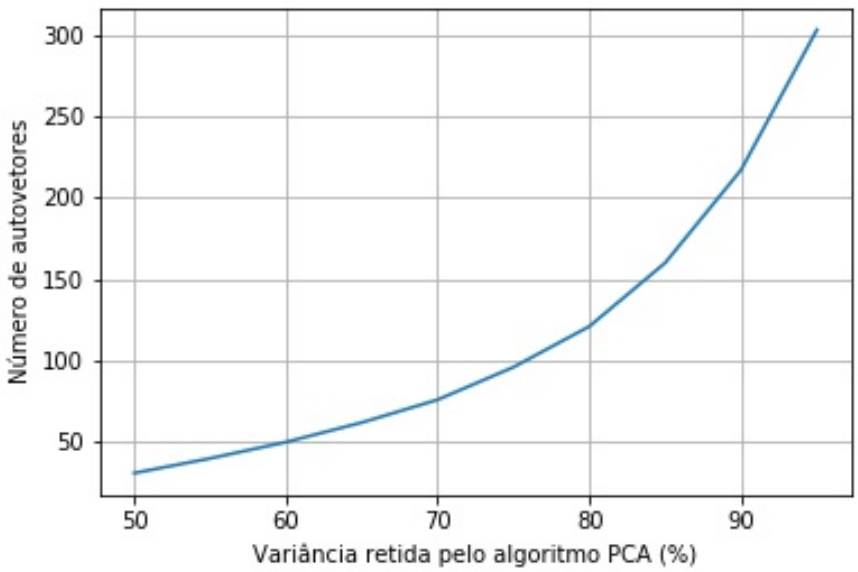

Figura 44: Número de autovalores em função da variância retida no algoritmo PCA

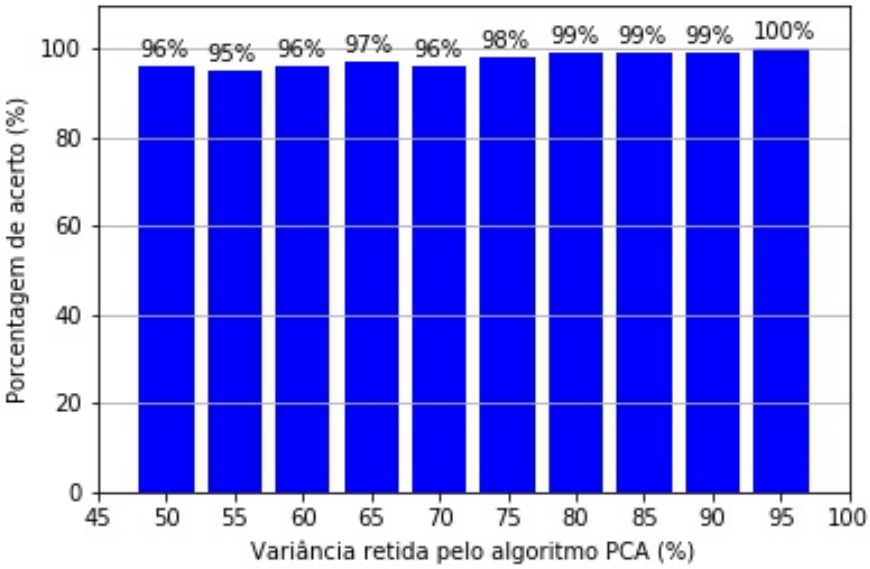

Figura 45: Porcentagem e acerto em função da variância retida com 20% do dataset para teste

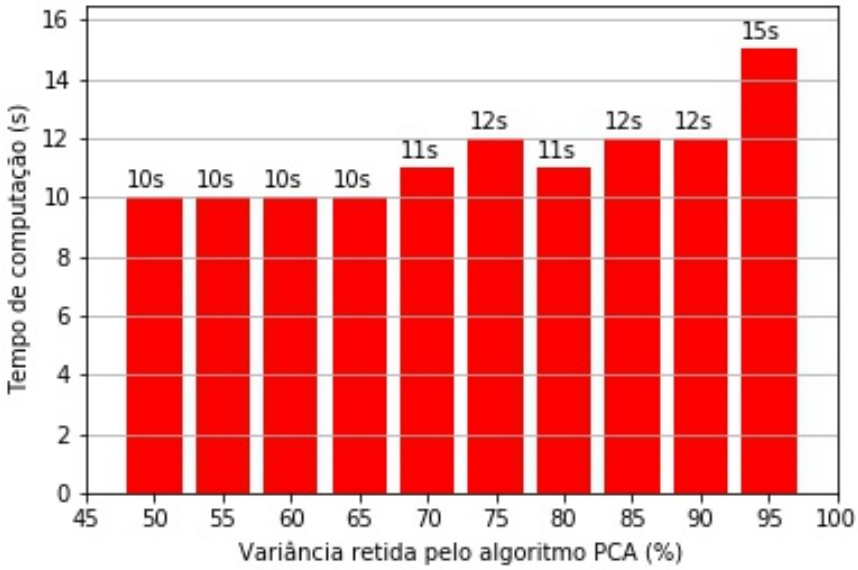

Figura 46: Tempo de computação em função da variância retida com 20% do dataset para teste

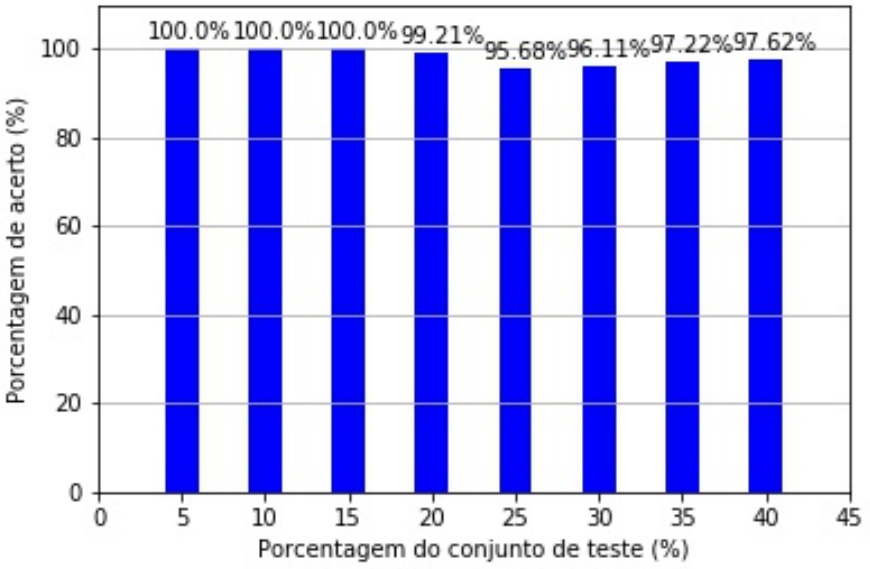

Figura 47: Porcentagem de acerto em função da porcentagem do dataset para teste com 85% da variância retida

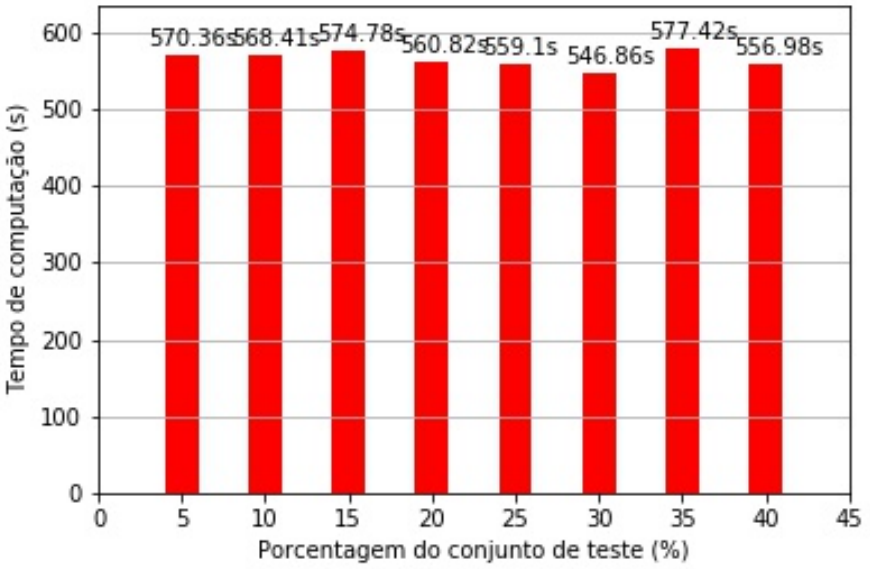

Figura 48: Tempo de computação em função da porcentagem do dataset para teste com 85% da variância retida

5.3 RESULTADOS OBTIDOS NA PLACA NVIDIA JETSON TX1

Após o desenvolvimento dos códigos e seus testes em laptop, estes foram transferidos para a Jetson e executados. Logo durante a transferência dos arquivos foi notada a primeira dificuldade, o espaço de armazenamento restante de apenas 2GB não seria o suficiente para criar os quatro datasets. Portanto, a única opção viável foi rodar o código de pré processamento para um conjunto de imagens, executar em todos os tipos de algoritimo extrator de características (PCA com SVM, SIFT e SURF) e após gerar os gráficos e tabelas com resultados apagar o dataset e refazer o processo para o próximo conjunto de imagens.

Uma segunda dificuldade encontrada durante o processo de execução dos softwares na placa foi em relação à versão do python utilizada. Como todo o código foi feito e rodado primeiramente usando python 3, ao tentar executá-los na Jetson muitos erros foram acusados, pois nela estava sendo utilizado python 2.7. Existiam duas maneiras das quais este problema poderia ser contornado, atualizando o código para ser rodado com python 2.7 ou instalando a versão 3. Como as modificações era simples de serem feitas, foi optada a primeira opção. Por fim, após rodar, notou-se que o OpenCv 3 precisava de uma biblioteca não instalada na Jetson, que contém funções de SIFT e SURF. Esta biblioteca foi relativamente difícil de instalar pois precisava de uma atualização da DIGITS que por consequência tinha como necessidade atualizar o *numpy*. Como a DIGITS não poderia ser atualizada pois as novas versões ainda eram incompatíveis com CAFFE, foi necessário desinstalar o *OpenCv 3* e reinstalá-lo fora do pacote JetPack 3.2. Pois desta maneira foi assegurado que a bibliteca *OpenCv contrib* fosse incluída. Abaixo estão os resultados e comentários dos testes feitos para o dataset Muzzle_IAPAR.

5.3.1 PCA E SVM

O primeiro passo após ter sido feito o pré processamento foi gerar os kernels lineares, polinomial, RBF e Sigmoid. O kernel linear obteve um ótimo desempenho, com uma porcentagem de acerto de 93,65%, muito superior à taxa de acerto do RBF(5.56%) e do Sigmoid(0.79%). Já o kernel polinomial alcançou uma taxa de acerto de 95.24% com um grau 1.0. As tabelas abaixo mostram um comparativo entre os três tipos de kernel, sua porcentagem de acerto e tempo de computação, além dos resultados do kernel polinomial para diferentes graus.

A Figura 49 mostra uma comparação entre o número de autovetores e a variância retida pelo PCA. Desta forma, é possível analisar que a partir de 75% da variância retida, o número

Tabela 1: Tabela de Resultados dos kernels Linear, RBF e Sigmoid

Kernel	Porcentagem de acerto (%)	Tempo de computação (s)
Linear	95,24	1,55
RBF	2,38	8,81
Sigmoid	0,0	20,43

Tabela 2: Tabela de Resultados do kernel Polinomial para diferentes graus

Grau	Porcentagem de acerto (%)	Tempo de computação (s)
0,1	35,71	0,15
0,5	10,32	0,05
0,99	80,95	0,16
1,0	95,24	0,14
1,1	84,13	0,16
1,5	94,44	0,18
1,99	93,65	0,17
2,0	93,65	0,17
2,1	93,65	0,18
2,5	92,86	0,17
2,99	91,27	0,18
3,0	92,86	0,15
3,01	91,27	0,19
3,5	87,3	0,19
3,99	83,33	0,19
4,0	83,33	0,19
4,01	84,13	0,2

de autovetores aumentou de forma gradativa. Isto reflete diretamente na porcentagem de acerto, que até 75% estava em 19% e a partir de então alcançou o nível de 93.65%, como é visto na Figura 50. Já o tempo de computação se manteve relativamente constante durante a faixa de variância retida de 40% à 100%, como é visto na Figura 51.

Usando uma variância retida de 85%, a qual proporcionou os melhore resultados, foi executado o próximo algorítimo. O dataset foi separado em classes de imagens e labels, treinado e classificado usando SVM. Após sua execução, os gráficos das Figuras 52 e 53 foram plotados, e, como pode ser visto, alcançou uma faixa de acerto aceitável em um tempo relativamente baixo de computação, quando comparado a outras técnicas testadas nesse trabalho.

Para finalizar a execução de PCA com Support Vector Machine, foram executados os treinamentos e classificações para os 10 diferentes estudos de caso. E, de forma interessante, foi notado que para um estudo de caso sem qualquer tipo de pré processamento digital das imagens, foi obtida a maior porcentagem de acerto como mostra a Figura 54. Além disso, o tempo de computação foi diferente em todos os estudos de caso. Atingindo seu ápice no terceiro tipo de estudo, como é visto na Figura 55

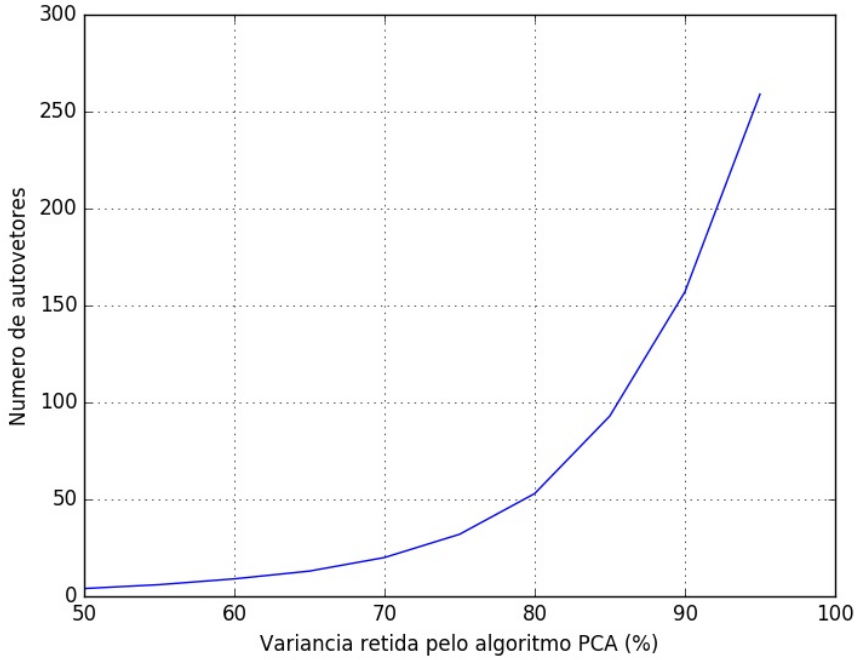

Figura 49: Número de autovetores gerados pelo algoritmo PCA em função da variância retida

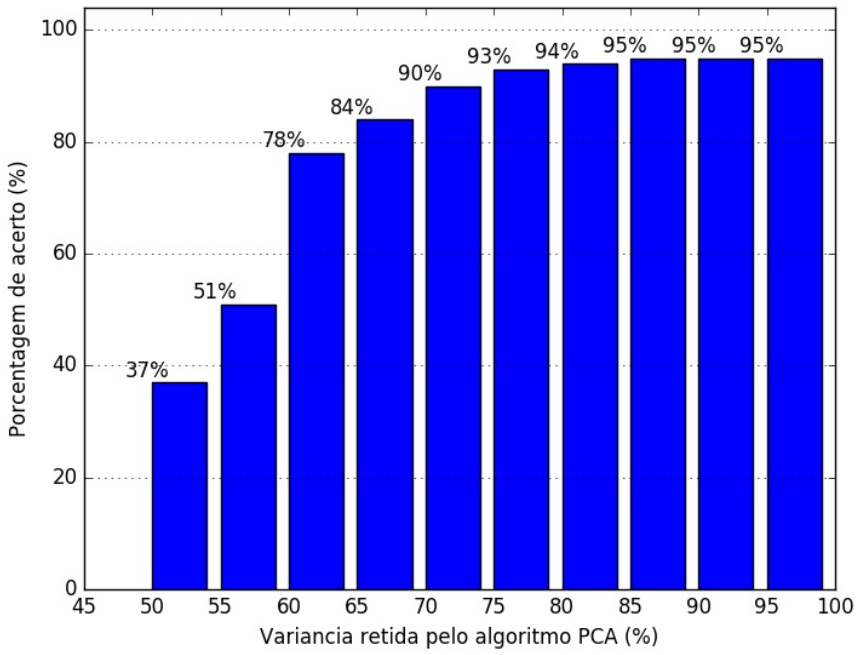

Figura 50: Porcentagem e acerto em função da variância retida com 20% do dataset para teste

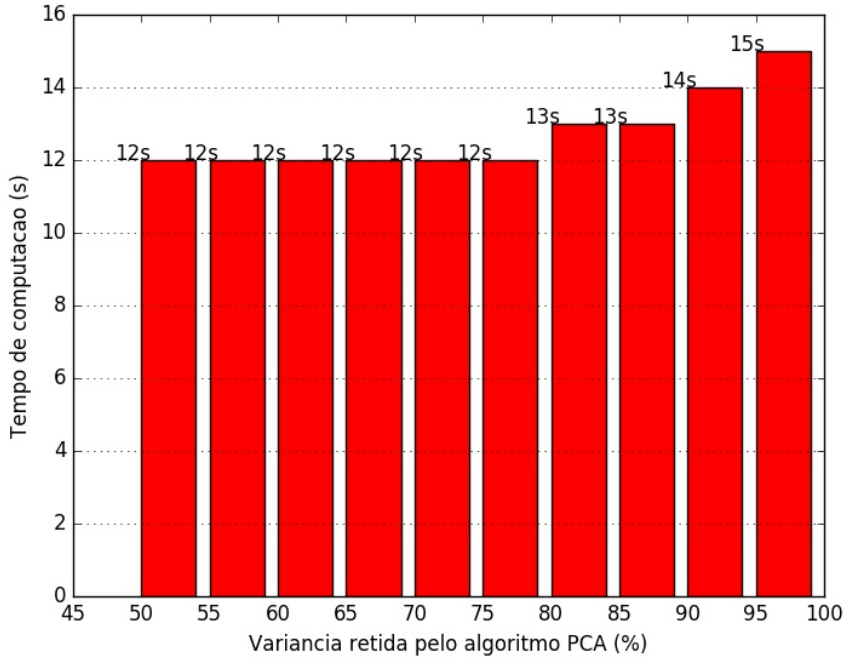

Figura 51: Tempo de computação em função da variância retida com 20% do dataset para teste

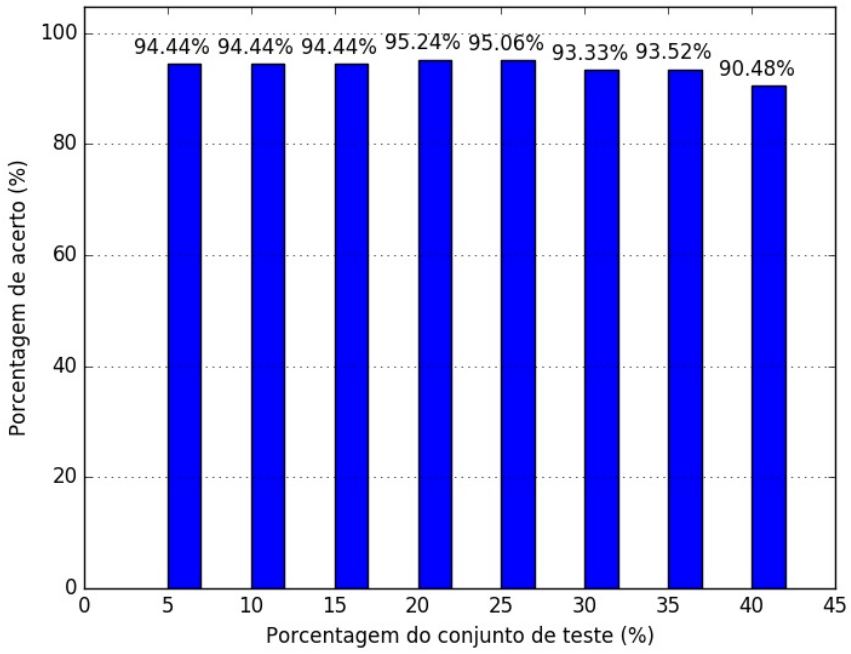

Figura 52: Porcentagem de acerto em função da porcentagem do dataset para teste com 85% da variância retida

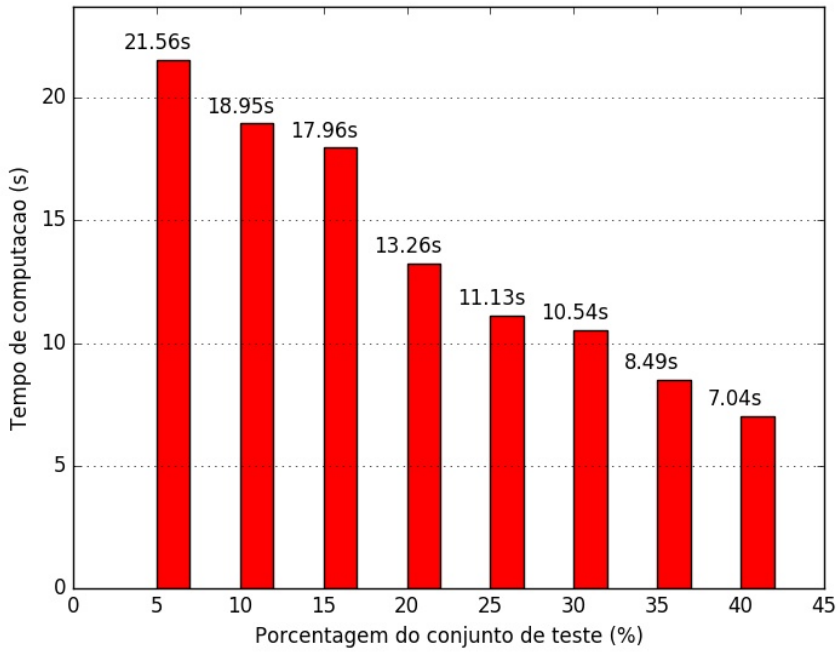

Figura 53: Tempo de computação em função da porcentagem do dataset para teste com 85% da variância retida

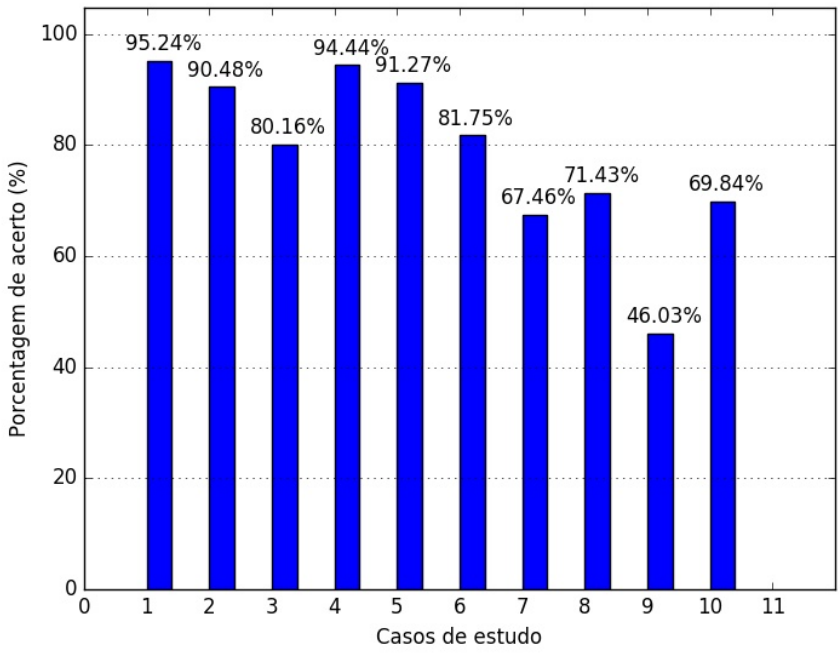

Figura 54: Porcentagem e acerto para cada caso de estudo com 85% de variância retida e 20% do dataset para teste

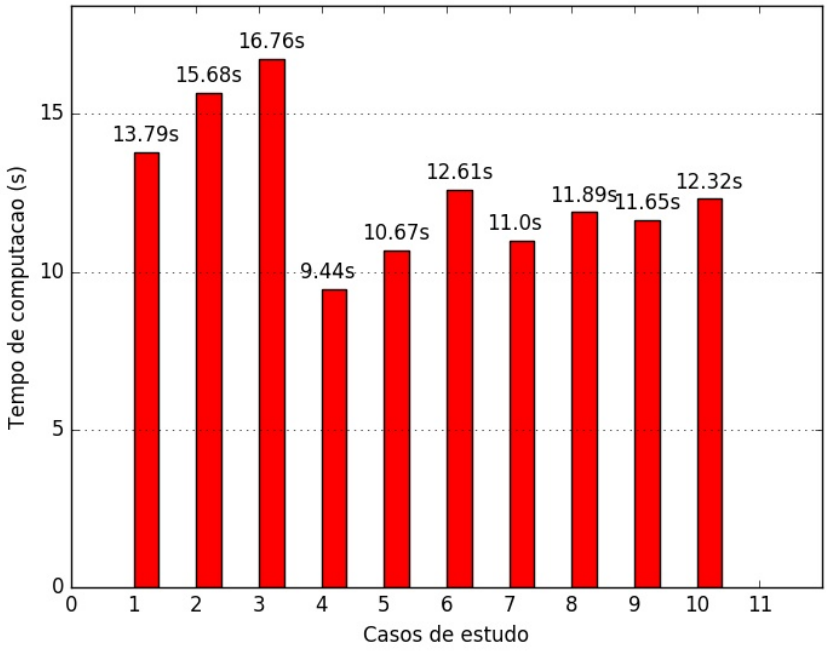

Figura 55: Tempo de computação para cada caso de estudo com 85% de variância retida e 20% do dataset para teste

5.4 DESCRIÇÃO DA METODOLOGIA USADA NA PLACA NVIDIA JETSON TX1

Com o avanço no campo de visão computacional, tão quanto o uso de aprendizado de máquina, aprendizado profundo e redes neurais em processamento de imagens, o número de dados utilizados para treinamento de datasets, suas inferências e módulos aumentou ao ponto que uma CPU convencional começou a não ter capacidade suficiente para realizar tais processamentos.

A NVidia Jetson TX1 é um computador em módulo utilizado com foco em visão computacional. Para isso, conta com processamento paralelo acelerado por uma GPU com arquitetura Maxwell de 256 CUDA cores e com ambiente Linux. Ela ainda possui um kit de desenvolvimento integrado com saídas USB, HDMI, entrada Ethernet e uma câmera de alta resolução (NVIDIA, 2015). Seu único ponto fraco é a baixa memória de disco, apenas 16GB. Pelo seu tamanho, mobilidade e eficiência energética, e disponibilidade, a Jetson foi a principal opção para embarcar os softwares de processamento dos datasets gerados.

O processo de desenvolvimento na placa foi dividido em duas etapas. Primeiramente foi necessário estudá-la, se familiarizar com suas funcionalidades e atualiza-la, já que muitas bibliotecas estavam com versões antigas. Em seguida, foi feito o trabalho de instalação de JetPacak 3.2, junto de seus pacotes CUDA, CAFFE e opencv 3.2. Algumas decisões foram tomadas em relação ao CAFFE, onde a versão 0.15 foi escolhida pelo fato de que as versões 0.16 e 0.17 não serem compatíveis com a DIGITS 6. Por fim, foi instalado o PIP e a DIGITS versão 6.

Após o desenvolvimento dos algoritmos descritos anteriormente e dos testes realizados em máquina, estes foram transsferidos para a Jetson e rodados com auxílio de ssh para sua validação e análise dos datasets gerados. Além disso, foram instaladas as bibliotecas que ainda faltavam bem como alterado o código para funcionamento na Jetson, pois nela foi utilizado python versão 2.7 no lugar de python 3 na qual foram desenvolvidos os softwares.

Foram utilizados dois lugares de trabalho para o desenvolvimento e testes na placa. No início, o laboratório localizado no terceiro andar do Bloco B da UTFPR foi o local de trabalho escolhido. Isto se deve ao fato de que foi necessário a ajuda do aluno Moritz Vogt para instruir os primeiros passos da utilização e manuseio da placa. A Figura 56 mostra a estrutura montada para os testes no laboratório. Após as primeiras semanas, a placa pode ser retirada do laboratório e trabalhada localmente. Isto ajudou com o fato de que os algorítimos demoravam horas para ser rodados. A Figura 57 mostra como foi montada a estrutura para trabalho local e pode ser visto a placa em execução. Também é possível observar, na Figura 58 um exemplo de execução

embarcada da placa com o algoritmo de PCA com SVM.

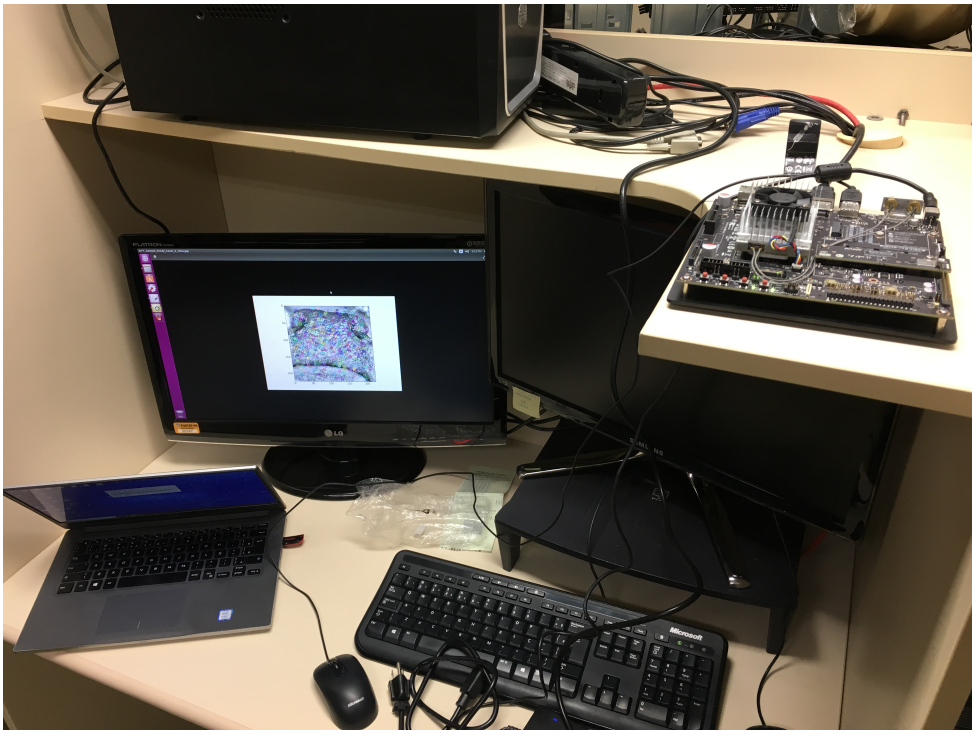

Figura 56: Estrutura montada no laboratório da UTFPR

Figura 57: Estrutura montada localmente para trabalhar com a Jetson

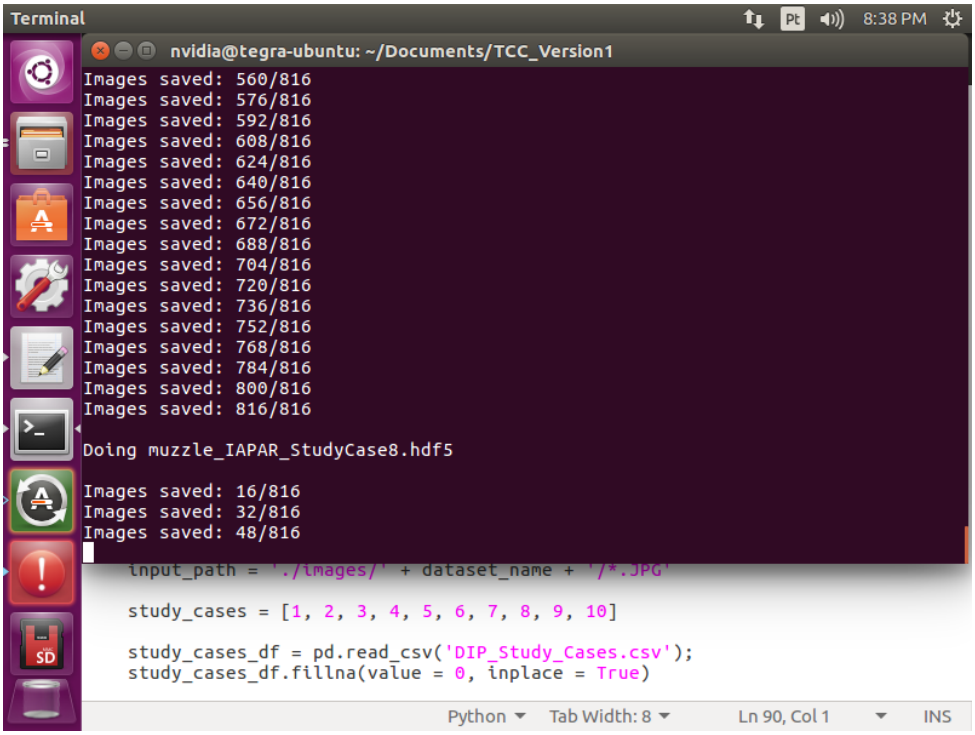

Figura 58: Exemplo de funcionamento embarcado da placa

6 CONCLUSÃO

Nesse trabalho foram testadas quatro combinações diferentes de algoritmos de extração de características e classificação em um notebook, e três combinações diferentes em um sistemas embarcado na placa Nvidia Jetson TX1.

A partir dos testes realizados, com diferentes variâncias retidas pelo algoritmo PCA, diferentes quantidades de imagens referência para os algoritmos SIFT e SURF, diferentes porcentagens de divisão entre conjunto de treinamento e com a aplicação de diferentes combinações de algoritmos de PDI, algumas tendências puderam ser observadas.

Em todas as técnicas, observou-se que a aplicação de algoritmos de PDI não trouxe ganhos significativos para os índices de acerto do algoritmo de classificação. De maneira que pode-se concluir que, para os conjuntos de imagens fornecidos pelo IAPAR e pela USP, e para as técnicas utilizadas, a aplicação de algoritmos de PDI pode ser dispensada. O único pré processamento que se mostrou realmente vantajoso em algumas técnicas, foi a conversão das imagens para escala de cinza.

Em se tratando das combinações de algoritmos, pôde-se observar, no caso do notebook, que a melhor técnica foi aquela que utilizou uma CNN como extratora de característica. Essa técnica foi capaz de atingir índices de acerto de 100% no conjunto de imagens do IAPAR e da USP, mostrando-se a técnica com maior capacidade de generalização.

No caso do sistema embarcado, a técnica com CNN não foi testada, contudo a técnica com PCA e SVM foi a que apresentou melhores índices de acerto. A partir disso, pode-se abstrair que a utilização de uma CNN como extratora de características juntamente com os algoritmos PCA e SVM seria a melhor escolha, uma vez a técnica com PCA e SVM é um subconjunto da técnica com CNN, PCA e SVM.

Foram realizados testes utilizando-se dois tipos diferentes de extração da ROI. No primeiro caso a imagem do focinho inteiro do bovino foi utilizada e no segundo, apenas a imagem do espelho nasal. As imagens do focinho inteiro do bovino, apresentaram em todas as técnicas utilizadas, índices de acerto iguais ou superiores aos índices apresentados pelas

imagens do espelho nasal, de maneira que pudemos observar que a imagem do focinho inteiro é mais apropriada para um sistema de classificação de bovinos.

A extração da ROI das imagens utilizadas nesse trabalho foram feitas manualmente, utilizando-se o software GIMP, de maneira que deixa-se aqui, como sugestão para uma pesquisa futura, o testes com algoritmos de Machine Learning e Deep Learning, capazes de aprender a reconhecer a região de interesse na imagem de um bovino e capazes de realizarem a segmentação da mesmo de uma maneira confiável. Assim, a saída do algoritmo de segmentação poderia ser utilizada como entrada para alguma das técnicas apresentadas nesse trabalho, e um sistema completo de classificação de bovinos em tempo real poderia ser realizado.

Finalmente, pudemos observar que, a construção de um sistema confiável de classificação de bovinos, para conjuntos de imagens com distribuições homogêneas, como o conjunto da USP, ou não homogêneas, como o conjunto do IAPAR, é realizável a partir de técnicas conhecidas de Machine Learning e Deep Learning. Tal sistema pode trazer celeridade, confiabilidade e reduzir os custo no processo de transporte e rastreamento de bovinos, por fazer uso de uma característica única e imutável do bovino. Por isso, deixamos como sugestão para futuras pesquisas, sistemas confiáveis de segmentação de imagens em tempo real, que possam ser integrados com sistemas de classificação como o desenvolvido nesse trabalho.

REFERÊNCIAS

AWAD, A. I. **From classical methods to animal biometrics: A review on cattle identification and tracking**. [S.l.]: Elsevier, 2016.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. **SURF: Speeded Up Robust Features**. 2008. Disponível em: <http://www.vision.ee.ethz.ch/ surf/eccv06.pdf>. Acesso em: 13 de julho de 2018.

BURGUER, W. **Principles of Digital Image Processing: Core Algorithms**. 2009.

CORNELISSE, D. **An intuitive guide to Convolutional Neural Networks**. 2018. Disponível em: <https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050>. Acesso em: 24 de julho de 2018.

DERTAT, A. **Applied Deep Learning - Part 4: Convolutional Neural Networks**. 2017. Disponível em: <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>. Acesso em: 24 de junho de 2018.

FORMIGOI, I. **Dados da exportação de carne bovina do Brasil em janeiro de 2018**. 2018. Disponível em: <http://www.farmnews.com.br/mercado/exportacao-de-carne-bovina-7/>. Acesso em: 17 de maio de 2018.

GABER, T. et al. **Biometric cattle identification approach based on Weber's Local Descriptor and AdaBoost classifier**. [S.l.]: Elsevier, 2016.

GHOLAMI, V. et al. **Modeling of groundwater level fluctuations using dendrochronology in alluvial aquifers**. [S.l.]: Journal of Hydrology, 2015.

GIMENEZ, C. M. **Identificação biométrica de bovinos utilizando imagens do espelho nasal**. Pirassununga: Universidade de São Paulo, 2015.

HEESCH, D. van. **Introduction to Principal Component Analysis (PCA)**. 2015. Disponível em: <https://docs.opencv.org/3.1.0/d1/dee/tutorial_introduction_to_pca.html>. Acesso em: 3 de julho de 2018.

KOERICH, A. L. **Aprendizagem de Máquina,** Programa de Pós-Graduação em Engenharia Elétrica, UFPR. Curitiba: Universidade Federal do Paraná, 2003.

KUMAR, S. et al. **Deep learning framework for recognition of cattle using muzzle point image pattern**. Jatái: Elsevier, 2018.

KUMAR, S.; SINGH, S. K. **Automatic identification of cattle using muzzle point pattern: a hybrid feature extraction and classification paradigm**. [S.l.]: Multimedia Tools and Applications - Springer, 2016.

KUMAR, S. et al. **Group Sparse Representation Approach for Recognition of Cattle on Muzzle Point Images**. [S.l.]: International Journal of Parallel Programming - Springer, 2017.

KUMAR, S.; SINGH, S. K.; SINGH, A. K. **Muzzle point pattern based techniques for individual cattle identification**. [S.l.]: IET Image Processing, 2016.

KUMAR, S. et al. **Real-time recognition of cattle using animal biometrics**. [S.l.]: Journal of Real-Time Image Processing - Springer, 2017.

LINDEBERG, T. **Scale Invariant Feature Transform**. 2015. Disponível em: <http://www.diva-portal.org/smash/get/diva2:480321/FULLTEXT02.pdf>. Acesso em: 13 de julho de 2018.

LORENA, A. C.; CARVALHO, A. C. P. L. F. de. **Uma Introdução ás Support Vector Machines**. Santo André: Universidade de São Paulo, 2007.

LOWE, D. **Distinctive Image Features from Scale-Invariant Keypoints**. 2004. Disponível em: <https://www.cs.ubc.ca/ lowe/papers/ijcv04.pdf>. Acesso em: 13 de julho de 2018.

MORDVINTSEV, A. **Understanding SVM**. 2013. Disponível em: <http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_ml/py_svm/py_svm_basics/py_svm_basics.html>. Acesso em: 5 de julho de 2018.

MOUJAHID, A. **A Practical Introduction to Deep Learning with Caffe and Python**. 2016. Disponível em: <http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/>. Acesso em: 24 de julho de 2018.

NCSU. **Introduction to Principal Component and Factor Analysis**. 2007. Disponível em: <ftp://statgen.ncsu.edu/pub/thorne/molevoclass/AtchleyOct19.pdf>. Acesso em: 3 de julho de 2018.

NVIDIA. **NVidia Jetson TX1**. 2015. Disponível em: <http://www.nvidia.com.br/object/jetson-tx1-module-br.html>. Acesso em: 13 de julho de 2018.

OPENCV. **Understanding k-Nearest Neighbour**. 2014. Disponível em: <https://docs.opencv.org/3.0-alpha/doc/py_tutorials/py_ml/py_knn/py_knn_understanding/py_knn_understanding.html>. Acesso em: 9 de julho de 2018.

OTSU, N. **A Threshold Selection Method from Gray-level Histograms**. 1979.

PISANO S. ZONG, B. M. H. M. D. R. E. J. E. D. **Contrast-Limited Adaptive Histogram Equalization: Speed and Effectiveness**. EUA: [s.n.], 1990.

REZA, A. M. **Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement,**Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee,Milwaukee, USA. EUA: [s.n.], 2003.

ROSEBROCK, A. **ImageNet: VGGNet, ResNet, Inception, and Xception with Keras**. 2017. Disponível em: <https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>. Acesso em: 15 de julho de 2018.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.

SCHUTERA, M. et al. Automated phenotype pattern recognition of zebrafish for high-throughput screening. **Bioengineered**, Taylor & Francis, v. 7, n. 4, p. 261–265, 2016. PMID: 27285638. Disponível em: <https://doi.org/10.1080/21655979.2016.1197710>.

SIMONYAN, K.; ZISSERMAN, A. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. [S.l.]: ICLR 2015, 2014.

SOBEL, I. **An Isotropic 3x3 Image Gradient Operator**. EUA: [s.n.], 2014.

TOROK, L. **Método de Otsu**. 2015. Disponível em: <http://www2.ic.uff.br/ aconci/OtsuTexto.pdf>. Acesso em: 17 de julho de 2018.

ZHANG, T. Y.; SUEN, C. Y. **A Fast Parallel Algorithm for Thinning Digital Patterns**. 1984. Disponível em: <http://www-prima.inrialpes.fr/perso/Tran/Draft/gateway.cfm.pdf>. Acesso em: 17 de julho de 2018.