

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS**

ALEXANDRE BRITO

**RECONHECIMENTO FACIAL BOVINO: UMA ALTERNATIVA AOS
MÉTODOS TRADICIONAIS DE RASTREIO**

**CAXIAS DO SUL
2021**

Alexandre Brito

**RECONHECIMENTO FACIAL BOVINO: UMA ALTERNATIVA AOS
MÉTODOS TRADICIONAIS DE RASTREIO**

Trabalho de Conclusão de Curso apresentado
como parte dos requisitos para a obtenção do
título de Engenheiro de Controle e Automa-
ção da Universidade de Caxias do Sul.

Orientador:
Prof. Dr. Guilherme Holsbach Costa

**Caxias do Sul
2021**

Ficha catalográfica.

BRITO, Alexandre

Reconhecimento facial bovino: uma alternativa aos métodos tradicionais de rastreio/Alexandre Brito; Orientador: Prof. Dr. Guilherme Holsbach Costa

45 folhas, 30 cm.

Trabalho de conclusão do curso de Engenharia de Controle e Automação — Universidade de Caxias do Sul, Área do conhecimento de ciências exatas e engenharias, 2021

Inclui referências bibliográficas.

1. Visão computacional. Reconhecimento facial. Agricultura 4.0. Conjunto de dados. I. REFERENCIAR ORIENTADOR II. Universidade de Caxias do Sul. III. Reconhecimento facial bovino: uma alternativa aos métodos tradicionais de rastreio.

Alexandre Brito

**RECONHECIMENTO FACIAL BOVINO: UMA ALTERNATIVA AOS
MÉTODOS TRADICIONAIS DE RASTREIO**

Trabalho de Conclusão de Curso apresentado
como parte dos requisitos para a obtenção do
título de Engenheiro de Controle e Automa-
ção da Universidade de Caxias do Sul.

Aprovado em ____/____/_____

Banca Examinadora

Prof. Dr. Guilherme Holsbach Costa
Universidade de Caxias do Sul - UCS

Prof. Dr. André Gustavo Adami
Universidade de Caxias do Sul - UCS

M.e Rodrigo Biasuz
Universidade de Caxias do Sul - UCS

A meus pais, família e amigos.

AGRADECIMENTOS

Eu gostaria de agradecer a meus professores e orientadores que me guiaram e acompanharam nesta jornada. A troca de conhecimentos que tivemos foi fundamental para eu conseguir chegar aonde cheguei.

A instituição que forneceu toda a estrutura e ferramentas para moldar o meu conhecimento e incentivar a tornar um profissional e acadêmico cada vez melhor.

Também gostaria de agradecer, de coração, a toda minha família, que me apoiaram e deram forças do início ao fim. A meus queridos amigos, vocês foram fundamentais em me dar energias e tornar essa jornada mais tranquila.

Agradecer especialmente a meus pais, Lourival e Rosane, que eu sei que sem eles eu jamais seria a pessoa que sou hoje e que jamais teria conseguido chegar até aqui. Vocês são o pilar de toda caminhada e minha principal motivação.

Por fim, agradecer a mim mesmo, por jamais ter desistido de buscar meu objetivo e sonho. Sei que a jornada até aqui não foi fácil e que meu esforço foi bem recompensado.

“A vida não é sobre o quão forte você bate. É sobre o quão duro você pode ser atingido e seguir em frente; Quanto você pode aguentar e continuar seguindo em frente.”

Rocky Balboa

RESUMO

A rastreabilidade bovina é de extrema importância tanto para o produtor quanto para o consumidor, uma vez que com ela se identificam detalhes da vida de cada animal sendo possível reagir a diferentes cenários. Atualmente, a rastreabilidade é usualmente feita com brincos ou com tecnologia RFID, que são sistemas eficazes porém com suas respectivas particularidades e desvantagens. Nos últimos anos, tem-se buscado alternativas a esses sistemas utilizando-se de diversas técnicas e tecnologias. Uma das alternativas é a utilização de visão computacional e processamento de imagens. Porém, uma realidade que impede que se amplie o número de estudos nessa área é a carência por uma base de dados de acesso aberto. Considerando isso, neste trabalho é proposta a construção de uma base de dados com imagens de faces de gados que possa ser utilizada para desenvolver modelos de identificação e detecção facial. A metodologia para criação da base é apresentada, bem como sua validação, justificativas para sua forma de criação e seus desafios.

Palavras-chave: Visão computacional. Reconhecimento facial. Agricultura 4.0. Conjunto de dados.

ABSTRACT

Bovine traceability is extremely important for both the producer and the consumer, since it identifies details of the life of each animal and makes possible to react to different scenarios. Currently, traceability is usually done with earrings or with RFID technology, which are effective systems, but with their particularities and disadvantages. In recent years, alternatives to these systems have been sought using various techniques and technologies. One of the alternatives is the use of computer vision and image processing. However, a reality that prevents the number of studies in this area from expanding is the lack of an open access database. Considering this, in this work is the proposed a database with images of the faces of cattle that can be used to develop models of facial identification and detection. The methodology for creating the database is presented, as well as its validation, justifications for its form of creation and its challenges.

Keywords: Computer vision. Facial recognition. Agriculture 4.0. Dataset.

LISTA DE FIGURAS

Figura 1 – Árvore demonstrando a divisão das técnicas de detecção facial	16
Figura 2 – Exemplo de <i>data point</i>	18
Figura 3 – Vista lateral do brete	22
Figura 4 – Suporte de fixação da câmera, destacado em vermelho, sobre a porta de saída do brete	23
Figura 5 – Exemplo de imagem fora de enquadramento, desconsiderada na base de dados	24
Figura 6 – Demonstração do resultado da extração da região de interesse	25
Figura 7 – Exemplo de uma <i>eigenface</i> de um gado	27
Figura 8 – Demonstração da relação da variância acumulada com os componentes principais	31
Figura 9 – Matriz de confusão para um treinamento com 90% do conjunto em treino	36
Figura 10 – Matriz de confusão para um treinamento com 75% do conjunto em treino	37
Figura 11 – Exemplos de faces de gado	38
Figura 12 – Exemplo de um gado da base, neste caso o 51, em ambiente não controlado	40
Figura 13 – Resultado da identificação	40

LISTA DE TABELAS

Tabela 1 – Funcionamento da Matriz de Confusão	33
Tabela 2 – Médias após 10 repetições, com variadas divisões e limiar 95%	35
Tabela 3 – Médias após 20 repetições, com variadas divisões e limiar 95%	35
Tabela 4 – Acurácias para diferentes métodos de reconhecimento	39
Tabela 5 – Acurácias para diferentes limiares	39

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	DETECÇÃO FACIAL VERSUS RECONHECIMENTO FACIAL	15
2.2	CONJUNTO DE DADOS E SUA IMPORTÂNCIA	17
2.3	COMO SE AVALIA UM CONJUNTO DE DADOS	19
3	MATERIAIS E MÉTODOS	22
3.1	AQUISIÇÃO E PROCESSAMENTO DAS IMAGENS	22
3.2	ANÁLISE DE COMPONENTES PRINCIPAIS	26
3.2.1	Notação e nomenclatura	27
3.2.2	A determinação do espaço vetorial	28
3.2.3	Determinação do sub-espaço de faces	30
3.2.4	Classificação de uma imagem	31
3.3	DIVISÃO E TREINAMENTO DO CONJUNTO	32
4	RESULTADOS E DISCUSSÕES	33
4.1	BASE RESULTANTE	33
4.2	MÉTRICAS UTILIZADAS PARA AVALIAÇÃO DO DESEMPENHO	33
4.3	AVALIAÇÃO DO DESEMPENHO DA BASE	34
4.4	COMPARAÇÃO DOS RESULTADOS	36
5	CONCLUSÕES	41
	REFERÊNCIAS	42

1 INTRODUÇÃO

A pecuária é um ramo da agricultura que lida com a domesticação, cuidado e reprodução de animais como porcos, bois, ovelhas e afins. A atividade é extremamente essencial para a vida humana desde sua descoberta durante a Revolução Neolítica, alterando completamente o estilo de vida humano, passando de caçador, coletor e nômade, para um estilo agricultor e sedentário fixo, permitindo o aumento da população (ANCIENT, 2010). No Brasil, a pecuária de corte e leite é um dos principais pilares econômicos, com um rebanho maior que 210 milhões de cabeças de gado bovino registradas, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), sendo maior que a população brasileira (IBGE, 2019)

Com rebanhos cada vez maiores, valiosos e lucrativos, com o passar do tempo, diversas formas de se rastrear os animais foram desenvolvidas. De acordo com a ISO 9000:2015 (ABNT, 2015) a rastreabilidade é definida como a habilidade de descrever a história, aplicação, processos ou eventos e localização, de um produto, a uma determinada organização. Resumidamente, pode-se dizer que rastrear é manter registros, ou dados, relativos à origem e ao destino de um produto, neste caso, o gado bovino. Com isso, se consegue um controle de todas ocorrências relevantes ao longo da vida do animal. Isso garante a segurança, tanto do produtor quanto do consumidor, uma vez que pode se identificar detalhes da vida de cada animal e permite reagir a diferentes cenários. Crises de doenças, como a febre aftosa, podem afetar severamente o comércio, por exemplo (MELDAU, 2006).

O Sistema Brasileiro de Identificação Individual de Bovinos e Búfalos (SISBOV), promulgado pela Instrução Normativa nº 1, de 10 de janeiro de 2002, é o sistema oficial de rastreabilidade no Brasil. Ele trata de um conjunto de ações, medidas e procedimentos adotados para caracterizar a origem, o estado sanitário, a produção e a produtividade da pecuária nacional. Sua adesão é voluntária, com algumas exceções.

Atualmente, a rastreabilidade pode ser feita com um chip colocado no animal de duas formas, externamente, acoplado a um brinco, ou internamente, inserido no rúmen do animal. Esse chip gera um identificador único, contendo todas as informações da vida do animal, desde sua origem, alimentação e até vacinação (GOV, 2020). A leitura destes dados geralmente é feita via RFID, usando um bastão ou com um painel na entrada do curral.(ALLFLEX, 2020). Embora seja um sistema robusto, pode-se considerar invasivo e estressante demais ao animal e com alguns riscos, como infecções, especialmente se usado internamente.

Nos últimos anos, tem-se buscado alternativas a esses sistemas utilizando-se de diversas técnicas e tecnologias. Uma das alternativas é a utilização de visão computacional e processamento de imagens. Kumar et al. (2017a) propôs uma solução se aproveitando dos padrões únicos encontrados nos focinhos dos bois, similar a uma impressão digital,

implementando diferentes técnicas de extração e reconhecimento de padrões. Uma aproximação diferente é a de Zin et al. (2018), que busca identificar os animais, dentro da baía, através do padrão de sua pelagem, também considerando que cada animal vai possuir um padrão único a ser extraído utilizando-se de segmentações de imagem. Esses trabalhos alcançaram bons resultados para o contexto da aplicação, na identificação em cenários variados. Contudo, não se pode ignorar que a aplicação prática deles é limitada. Para utilizar, exclusivamente, o focinho dos animais se necessita de uma imagem bem enquadrada e de boa qualidade para conseguir fazer um recorte que seja utilizável por um *software*, algo que pode ser bastante desafiador em um ambiente real e não controlado. O reconhecimento através da pelagem acaba sendo ineficaz para determinadas raças de gado, especialmente se considerarmos as mais comuns em solo brasileiro, tornando a técnica bastante limitada.

Bergamini et al. (2018) buscaram utilizar o conhecimento prévio, e validado, do reconhecimento facial humano e transportar isso para o uso em animais. Uma potencial fraqueza do sistema proposto é que ele utiliza primariamente um sistema baseado em redes neurais e *deep learning*, técnica que é fortemente baseada em um grande *dataset* de imagens para ter sucesso. Pensando em um princípio mais básico, Kumar, Tiwari e Singh (2015) e Cai e Li (2013) propuseram a extração de vetores de características da face do gado, utilizando técnicas similares, atingindo resultados igualmente satisfatórios. Tais vetores de características podem ser considerados como um pré-processamento para a alimentação de uma rede neural, sendo um ponto de início para um sistema de rastreamento, apesar de ainda precisar de uma quantidade considerável de imagens, a fim de validar o modelo. Entretanto, até onde abrange a revisão de literatura deste trabalho, não há uma base de imagens disponível (de acesso aberto) para esse tipo de aplicação.

Utilizando-se dessa premissa, este trabalho busca construir um *dataset* de faces de gados que possa ser utilizado para desenvolver modelos de identificação e detecção facial. Para tanto, será avaliada uma metodologia para a construção desse *dataset*, bem como uma forma de validação para a base proposta. Serão discutidas as principais abordagens utilizadas para o reconhecimento facial humano e alguns métodos utilizados como base de validação para conjunto de dados, suas vantagens e desvantagens.

1.1 OBJETIVOS

O objetivo geral deste trabalho é construir de um *dataset* bovino voltado para reconhecimento facial dos animais através técnicas de processamento de imagem. Para o cumprimento deste objetivo geral, tem-se os seguintes objetivos específicos:

- Colher vídeos de diferentes gados em diferentes fases;
- Selecionar os melhores *frames* de cada sujeito e identificá-los;

- Realizar um estudo de técnicas necessárias para realizar a validação;
- Definir as melhores técnicas a serem utilizadas para validação;
- Desenvolver um *software* capaz de realizar a validação.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 DETECÇÃO FACIAL VERSUS RECONHECIMENTO FACIAL

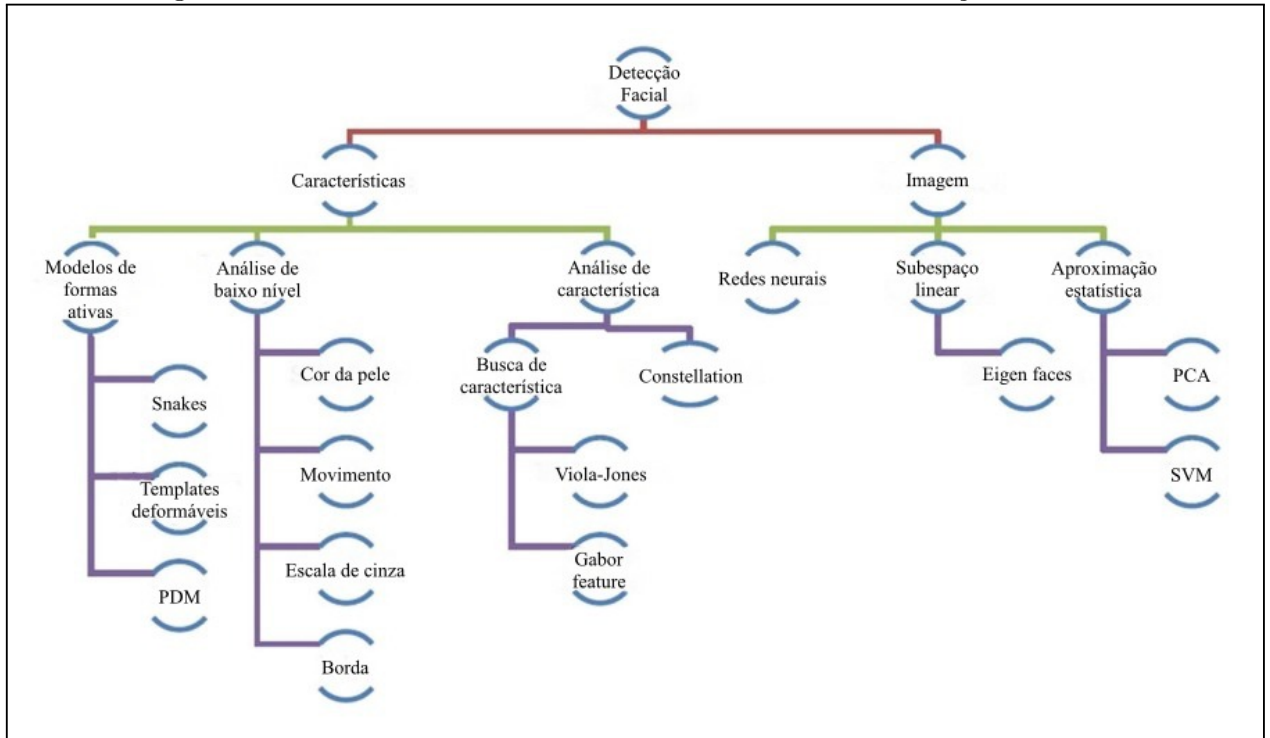
A diferença entre detecção e reconhecimento facial é, essencialmente, de aplicação, sendo portanto intimamente relacionadas. A detecção facial trata de localizar rostos, caso existam, e estimar seus tamanhos em uma imagem. Usualmente esse é o primeiro passo em qualquer operação a se fazer com faces, seja rastreamento, determinação de pose ou identificação, pois assim se consegue extrair a região da imagem que possui o rosto, o que facilita o processamento. Já o reconhecimento facial trata de identificar quem é o indivíduo ao qual aquele rosto pertence, cruzando parâmetros daquele rosto com os de um banco de dados.

Ambos conceitos podem ser implementados por técnicas muito semelhantes. De modo geral, as técnicas de detecção facial podem ser divididas em dois grandes grupos, sendo um deles baseado em características e o outro chamado de baseado em imagens, como demonstrado na Figura 1. O grupo baseado em características, como o próprio nome já indica, busca analisar características extraídas de regiões isoladas em uma imagem (geralmente características morfológicas, dos olhos, boca, nariz, etc.) e compará-las com o que se conhece sobre as características faciais. Um dos algoritmos mais famosos de detecção facial que utiliza essa aproximação é o algoritmo de Viola-Jones¹, proposto em 2001. O que torna esse algoritmo tão eficaz é sua robustez e velocidade, que possibilita o seu uso em tempo real (VIOLA; JONES, 2001). Já o grupo baseado em imagem, que também pode ser chamado de método baseado em área, adota características da imagem como um todo, geralmente estatísticas, e comumente recai no uso de classificadores treinados para encontrar rostos na imagem como, por exemplo, a Análise de Componentes Principais (PCA — *Principal Component Analysis*), que é utilizada para gerar subespaços lineares reduzindo, assim, o peso computacional do processo, o que permite detectar faces em tempo real. Seu uso voltado para este fim é hoje conhecido como *Eigenfaces*, popularizado por Turk e Pentland (1991). Neste ramo também tem-se o SVM (*Support Vector Machines*), técnica de aprendizagem desenvolvida por Boser, Guyon e Vapnik (1992) que foi utilizada por Osuna, Freund e Girosit (1997) como uma alternativa para se linearizar a detecção facial. De uma forma simplificada, é possível dizer que um classificador linear é um classificador que separa os dados em classes através de hiperplanos (uma reta em um espaço bidimensional, um plano no espaço tridimensional ou um hiperplano em um espaço de dimensão elevada) (KUBAT, 2015).

É interessante destacar que muitas técnicas utilizadas para detecção de rostos são adaptadas de técnicas utilizadas para detecção de objetos, usando um classificador focado

¹Pesquisando-se pelo termo *face detection* na base de dados do Google Acadêmico obtêm-se o artigo de Viola e Jones como o mais citado, superando 15 mil citações.

Figura 1 – Árvore demonstrando a divisão das técnicas de detecção facial



Fonte: Adaptado de Kumar, Kaur e Kumar (2018)

apenas em um objeto específico. Algoritmos, como Viola-Jones, nascem como detectores de objetos e, posteriormente, evoluem para serem específicos em uma função.

As técnicas de reconhecimento facial também podem ser subdivididas em holísticas, híbridas e por características (M. PARISA BEHAM, 2013). Essencialmente, é a mesma coisa que se observa na detecção facial, apenas com nomes diferentes e a adição de uma terceira forma. As formas híbridas utilizam as técnicas holísticas combinadas com características locais como, por exemplo, expandindo o conceito de *Eigenfaces* para boca, olhos e nariz. Já as técnicas baseadas em características são similares às de detecção, visto que o objetivo desses algoritmos é obter uma espécie de descritor da face para compará-lo com um padrão conhecido obtido anteriormente, ou seja, a principal diferença está na fase de classificação (M. PARISA BEHAM, 2013). Aproximações holísticas utilizam informações globais das faces de modo a realizar o reconhecimento. Essas informações são representadas fundamentalmente por um pequeno número de características que são derivadas diretamente de *pixels* de imagens de faces. Esse pequeno número, por sua vez, captura a variância entre diferentes faces e, então, pode ser utilizado para reconhecimento. Entre as principais técnicas, em especial para grandes conjuntos, se destacam a *Eigenfaces* e a *Fisherfaces* (M. PARISA BEHAM, 2013). A técnica *Fisherfaces* é bem similar às *Eigenfaces*, sendo sua principal diferença o fato de ela utilizar a técnica de Discriminante Linear de Fisher (FLD — Fisher's linear discriminant), de onde vem seu nome. FLD é um exemplo de uma técnica de classe específica, que significa que ele busca moldar a dispersão

dos dados de modo a torná-la mais confiável para classificação (BELHUMEUR; HESPANHA; KRIEGMAN, 1997). Comparativamente, PCA e FLD são bem parecidos e ambas as técnicas foram utilizadas para projeção e redução de dados, por exemplo. A diferença principal é o que ambas técnicas conseguem. A FLD consegue uma dispersão melhor entre classes, simplificando a classificação, enquanto a PCA consegue uma dispersão geral melhor. A *Fisherfaces* utiliza uma combinação de PCA com FLD, em uma tentativa de obter o melhor das duas técnicas (BELHUMEUR; HESPANHA; KRIEGMAN, 1997).

Pode-se notar, pela literatura, que muitas das técnicas utilizadas para o reconhecimento facial são as mesmas utilizadas para a detecção. Muito disso se trata pela natureza das duas ações. Para se detectar um rosto, é preciso classificar se aquele objeto é uma face, seja identificando características básicas como olhos, nariz e boca, ou mesmo o formato do rosto, enquanto, para se reconhecer a pessoa, os mesmos dados que servem para dizer o que é um rosto são necessários para se dizer de quem é aquele rosto, porém comparados com métricas específicas que auxiliem nessa determinação. Obviamente existem técnicas específicas para uma determinada aplicação, como é o caso do algoritmo de Viola-Jones, que não pode ser usado para reconhecimento, por exemplo. Isso acontece pela própria natureza do algoritmo, que nasce com um propósito já pré-determinado, enquanto outros, como LBP, por exemplo, foram adaptados por conta de sua finalidade.

Uma técnica versátil, que pode ser utilizada para ambas as tarefas, é a das *Eigenfaces*. Segundo seus autores, a projeção dos dados em um subespaço linear permite seu uso para identificação e para detecção (TURK; PENTLAND, 1991). Por exemplo, se apenas busca-se detectar uma face, se analisa se aquela imagem está próxima do subespaço ou não, independentemente de estar próxima das classes. Por utilizar como base a PCA, seu treinamento é de baixo custo computacional e consegue ser utilizada em tempo real, especialmente com a tecnologia de hoje. Além disso, já é uma técnica bem validada, estudada e madura (os trabalhos originais já foram citados mais de 25000 vezes, de acordo com o Google Acadêmico). *Fisherfaces*, apesar ser uma técnica que consegue, segundo seus autores, resultados melhores que as *Eigenfaces*, possui uma implementação mais complexa, uma vez que utiliza uma combinação de duas técnicas distintas. Em virtude disto, a técnica *Eigenfaces* será o método primário considerado neste trabalho.

2.2 CONJUNTO DE DADOS E SUA IMPORTÂNCIA

Um conjunto de dados, usualmente tratado pelo termo em inglês — *dataset*, é uma coleção de dados relacionados. Para o problema de reconhecimento facial, por exemplo, um conjunto de dados não se constitui apenas de um conjunto de imagens. Similar ao que acontece em outras aplicações, ele contém todas as demais informações pertinentes a essa imagem no contexto do problema tais como, identificação do indivíduo, data de aquisição, gênero, idade, entre outros. Sendo assim possível utilizar essas informações

para alimentar modelos e algoritmos de modo a prever uma saída com acurácia, baseado na entrada recebida.

Quanto mais dados existirem e maior for a sua representatividade em relação a todo o universo de possibilidades, mais os modelos serão capazes de prever o resultado com dados desconhecidos. Logo, para se construir um conjunto de dados são necessárias várias amostras em diversos cenários. Um exemplo de *dataset* clássico, muito utilizado na literatura, é o *Iris flower dataset*, proposto por Fisher (1936)². Ele fez um extenso estudo sobre o uso de múltiplas medidas para um problema de taxonomia das flores Iris, coletando dados como largura e comprimento das pétalas de três espécies e tabelando todos esses dados conforme ilustrado no Anexo A. Esse conjunto é relativamente simples, porém, por este motivo, o conjunto acaba sendo bastante didático e uma boa forma para se entender conceitos básicos sobre a formação de um conjunto e suas utilidades. Por exemplo, neste *dataset*, podemos observar facilmente os atributos, que seriam as medidas, os valores categóricos, que são as três categorias de flores, e os chamados *data points*, que são o conjunto de uma ou mais medidas de um único sujeito dentro do conjunto, como mostrado na Figura 2.

Figura 2 – Exemplo de *data point*

<i>Iris setosa</i>			
Sepal length	Sepal width	Petal length	Petal width
5.1	3.5	1.4	0.2

Fonte: (FISHER, 1936)

Datasets também podem ser utilizados para confirmar que a solução do problema se comporta como deve, uma vez que já se sabe qual o resultado esperado para determinada entrada. Isso é válido tanto para aplicações que utilizam inteligência artificial, quanto para aplicações mais simples que utilizam algoritmos mais tradicionais. Os *datasets* também facilitam a refinar e ajustar os parâmetros da aplicação de forma a conseguir melhores acurácia e precisão.

Para o caso de imagens, o usual é construir o *dataset* utilizando diversas fotos do objeto de interesse, de modo a se conseguir um grande volume de imagens, com intuito de aumentar a quantidade de amostras disponíveis³. Cada imagem do conjunto precisa conter alguma forma de identificação, também conhecida como rótulo ou anotação, também

²O *dataset* pode ser encontrado em: <https://archive.ics.uci.edu/ml/datasets/iris>

³Amostra, enquanto conceito estatístico, é qualquer subconjunto de elementos de uma população (MONTGOMERY; RUNGER, 2016). Em se tratando de um conjunto de imagens, por exemplo, a totalidade de imagens que compõem esse conjunto se denomina população, enquanto a uma única imagem ou a um subconjunto delas chama-se de amostra.

tratado pelo seu termo em inglês — *label*, que vai ser, o resultado a ser previsto pelo modelo e auxiliar na tomada de decisão dele. Não existe um padrão para isso, variando de conjunto para conjunto e dependendo da situação. Por exemplo, no caso de gados, uma forma de se criar um rótulo pode ser pelo número de identificação do brinco do animal e um contador de quantas imagens daquele animal existem no *dataset*.

De modo geral, a composição de um conjunto de dados pode ser considerado um dos primeiros passos para o desenvolvimento da maioria das aplicações em visão computacional e reconhecimento de padrões, junto com a formulação do problema (WEBB; COPSEY, 2011). É através dele que saberemos como os algoritmos devem se comportar e quais os resultados esperados. Também é possível se otimizar algoritmos e ajustá-los às necessidades específicas de cada situação. Desta forma ele se torna peça-chave no desenvolvimento desses sistemas.

2.3 COMO SE AVALIA UM CONJUNTO DE DADOS

A avaliação de um *dataset* começa pela a avaliação da qualidade dos dados. Rosli, Tempero e Luxton-Reilly (2018) buscaram formalizar uma forma de se avaliar a qualidade dos dados de um *dataset*, de modo a facilitar que pesquisadores cheguem a conclusões e interpretações sobre o mesmo. Os autores descrevem quatro passos para se avaliar a qualidade de um *dataset* e, também, elencaram os problemas de qualidade mais comuns para dados. Esses problemas mais comuns são: dados duplicados, onde se possui dois ou mais registros que possuem os mesmos valores para a mesma métrica e para a mesma entidade⁴; dados inconsistentes, onde dois ou mais registros possuem valores diferentes para a mesma métrica e para a mesma entidade; dados faltando, um registro que não possua valor para alguma métrica; e dados incorretos, onde um registro possui valores impossíveis para alguma métrica. Identificar estes problemas pode ajudar na construção de *datasets* mais consistentes e mais precisos.

Conseguindo-se garantir a qualidade do conjunto, pode-se pensar em uma forma de colocá-lo à prova, utilizando algum algoritmo como base para avaliação, que é um algoritmo básico para gerar métricas de base para aplicações mais complexas. Existem diferentes formas de se fazer isto, não existindo portanto um padrão específico. A forma a ser adotada depende de diversos fatores, porém o mais comum encontrado na literatura para problemas de visão computacional é utilizar o conjunto de dados avaliado para treinar um algoritmo para produzir um modelo e observar o comportamento deste modelo. A vantagem dessa aproximação é que ela retorna métricas que tornam a análise mais completa. Obviamente, não se utiliza todo o conjunto de dados para isso, se cria um protocolo para realizar a divisão do conjunto em treinamento e validação (ou ainda em três conjuntos, em alguns casos). Esse protocolo visa a divisão mais eficaz do conjunto,

⁴Entidade é um objeto que pode ser medido por seus atributos ou propriedades.

de forma a facilitar o trabalho de treinamento para o algoritmo base que vai realizar a avaliação do conjunto e também ajuda a torná-lo melhor treinado, no sentido que ele executa suas previsões da melhor maneira possível.

Uma forma interessante de se dividir um conjunto para reconhecimento facial é apresentado por Gao et al. (2007), que propôs um protocolo dividindo o conjunto em: treinamento, galeria e o que ele chama de sonda. O conjunto de treinamento é formado por 1200 imagens de 300 indivíduos selecionados aleatoriamente dentro dos 1040 que ele possui, com cada um contribuindo com 4 imagens aleatórias. Já o conjunto galeria é um conjunto de indivíduos conhecidos que será comparada com uma imagem do conjunto de sonda e é formada por uma imagem de cada indivíduo no *dataset*, totalizando 1040 imagens. O conjunto de sonda é, efetivamente, o que poderia ser chamado de conjunto de teste. Ele é formado por diversas imagens de indivíduos, provenientes da própria base, a serem reconhecidos e em diversos cenários, excluindo-se as imagens utilizadas no conjunto de treinamento e galeria. No caso, o autor produz um conjunto sonda para cada cenário diferente de expressão, iluminação, acessório (como óculos), fundo, distância e tempo, neste caso, tempo refere-se a meio ano entre as sessões de foto do sujeito Gao et al. (2007) acredita que essas divisões podem ser usadas para verificar os pontos fortes e fracos dos algoritmos ou modelos utilizados. Apesar disso, essa divisão é bastante complexa e praticável apenas em conjuntos com inúmeras imagens, Gao et al. (2007) possui quase 31000 imagens, algo que nem sempre acontece.

Alternativamente, pode-se utilizar um método mais simples, chamado de *holdout*. Neste método, se divide aleatoriamente o conjunto simplesmente em dois subconjuntos, treinamento e teste, em uma dada proporção, por exemplo 75% e 25%. O problema com este método é que ele realiza o treinamento e o teste apenas uma vez o que pode gerar métricas incorretas e induzir o pensamento de que o modelo está prevendo os resultados corretamente (KOHAVI, 1995). Uma forma de se solucionar este problema é utilizar o método chamado *random subsampling*. Similar ao método anterior, também se divide aleatoriamente em dois subconjuntos em uma dada proporção, porém, ao invés de realizar apenas uma sessão de treinamento e teste, se faz n sessões e, em cada uma delas, um novo par de subconjuntos é gerado. Para cada sessão, as métricas são salvas e posteriormente se extrai uma média delas (KUBAT, 2015). Pode-se dizer que o *random subsampling* é uma extensão do *holdout*, uma vez que é, basicamente, o *holdout* executando várias vezes seguidas.

Após decidir como segmentar o conjunto, pode-se partir para a discussão de qual o algoritmo base a se utilizar e como interpretar os seus resultados, em formato de métricas, e saber qual desempenho pode-se atingir com aquele conjunto e com aqueles parâmetros utilizados na divisão. Um exemplo de algoritmo base utilizado no ramo da visão computacional é a PCA, pura ou utilizando a técnica *Eigenfaces*, que utiliza componentes principais como um de seus pilares. A PCA é um algoritmo extremamente versátil e tam-

bém já foi extensamente estudado e aplicado. Também pesando a seu favor, tem-se a sua relativa facilidade de implementação e principalmente o fato de que ele, de certo modo, remove variáveis correlacionadas, que são dados que não vão influenciar no processo de decisão do algoritmo, tornando o tempo de treinamento dele mais rápido. Com a análise das métricas obtidas, podemos identificar eventuais fraquezas do conjunto, como a baixa variância dos dados, por exemplo, ou mesmo do método de divisão utilizado e ajustar, tanto o conjunto, quanto o algoritmo, para que a resposta obtida seja o mais próximo do ideal e o mais confiável possível.

3 MATERIAIS E MÉTODOS

Este capítulo trata do método de construção do conjunto de dados, bem como algumas técnicas específicas que serão utilizadas e como utilizá-las para realizar a validação do conjunto.

3.1 AQUISIÇÃO E PROCESSAMENTO DAS IMAGENS

De modo a se conseguir imagens mais padronizadas e uniformes entre si, é proposta a aquisição de vídeos de animais em um brete, também conhecido por tronco de contenção, ilustrado na Figura 3. O brete é uma jaula projetada para a contenção ou imobilização do animal¹. A câmera, no caso um telefone celular, é fixada no alto da porta de saída do brete, conforme Figura 4, visando, desta forma, aumentar a probabilidade de se conseguir imagens do animal bem enquadradas, obtidas a partir de uma vista frontal, em condições reais de aplicação. Nenhum animal é manejado com a finalidade de que se faça a aquisição das imagens ou mesmo tem sua rotina no manejo alterada em função da aquisição. Todas as gravações são realizadas durante o período e a rotina regular do manejo do rebanho. Com isso, são capturados vídeos de cada animal, durante a permanência no brete para avaliação dos brincos, inseminação ou outro tipo de manejo.

Figura 3 – Vista lateral do brete



Fonte: O autor (2020)

¹Esse tipo de dispositivo é baseado na *máquina do abraço*, de Temple Grandin, autista, Psicóloga e Doutora em Zootecnia que revolucionou o manejo e doma de animais com base em experiências sensoriais próprias. As práticas de Grandin são atualmente conhecidas por Doma Racional. Grandin, Temple; Johnson, Catherine (December 26, 2004). "Animals in Translation". The New York Times. ISSN 0362-4331. Retrieved April 14, 2019.

Figura 4 – Suporte de fixação da câmera, destacado em vermelho, sobre a porta de saída do brete



Fonte: O autor (2020)

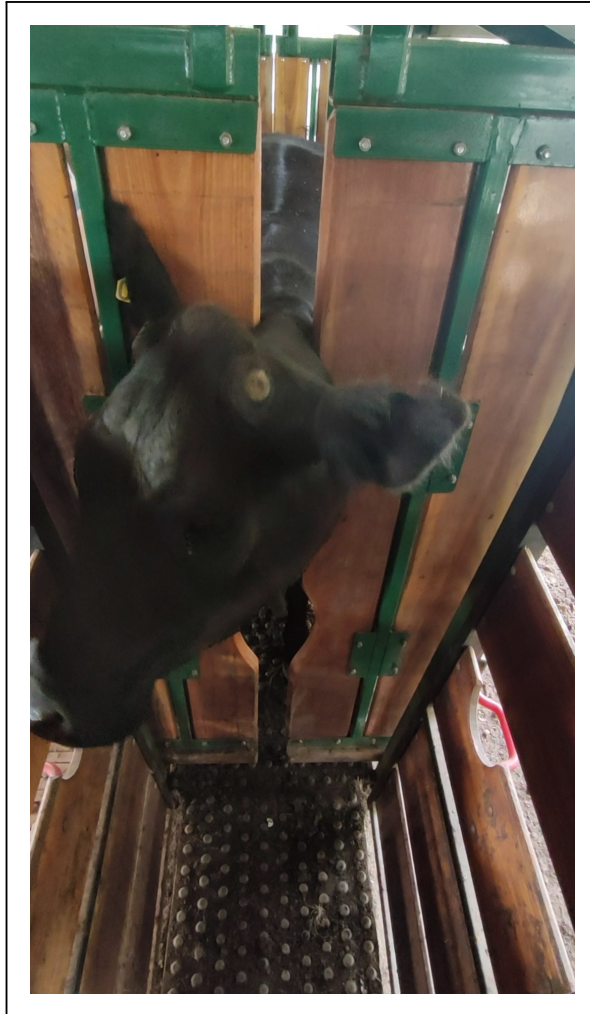
Os vídeos são gravados em 1080×1920 *pixels* e com uma taxa de 30 quadros por segundo, que é a resolução máxima do equipamento à disposição. O uso de vídeos para captura de imagens facilita a busca por imagens bem enquadradas e com exposição adequada, uma vez que não é necessário esperar o momento mais conveniente para capturar uma foto (considere-se que o animal, por muitas vezes, se agita dentro do brete, o que dificultaria a aquisição de uma foto). Cada vídeo é nomeado (rotulado) de acordo com a identificação do animal, correspondendo ao número do brinco dele, e, posteriormente, todos vídeos são processados por um *script* feito em Python² de modo a se extrair todos os quadros e salvar as imagens, numerando cada uma de acordo com a identificação dos animais. As imagens são salvas em um diretório único que, por sua vez, é dividido em sub-diretórios para cada indivíduo, mantendo, desta forma, os arquivos organizados facilitando o manuseio.

Considerando que, de forma a não alterar a rotina dos animais, não é utilizada iluminação dedicada durante a aquisição das imagens e que os animais por vezes se movimentam

²Python é uma linguagem de programação interpretada lançada em 1991 por Guido van Rossum. É uma linguagem versátil e muito utilizada para aplicações de visão computacional, facilitando o desenvolvimento.

rapidamente, algumas imagens são adquiridas com velocidade do obturador muito baixa para o contexto de movimento, gerando aquisições com borramento. Por essa razão, é feita uma verificação visual de todas as imagens capturadas de modo a remover imagens borradas ou desenquadradas, a exemplo do que pode ser visto na Figura 5, caso de um *frame* descartado. Também é feita, via *software*, uma avaliação da diferença quadrática média (erro quadrático médio) entre as imagens³, de modo a se eliminar *frames* subsequentes muito semelhantes entre si. O erro quadrático médio é relativamente simples de se implementar e possui um baixo custo computacional, o que mantém a agilidade neste processo. Dessa forma, busca-se que apenas as melhores imagens de cada animal sejam incluídas no conjunto de dados e que essas imagens não sejam repetidas (ou muito similares), na medida do possível.

Figura 5 – Exemplo de imagem fora de enquadramento, desconsiderada na base de dados

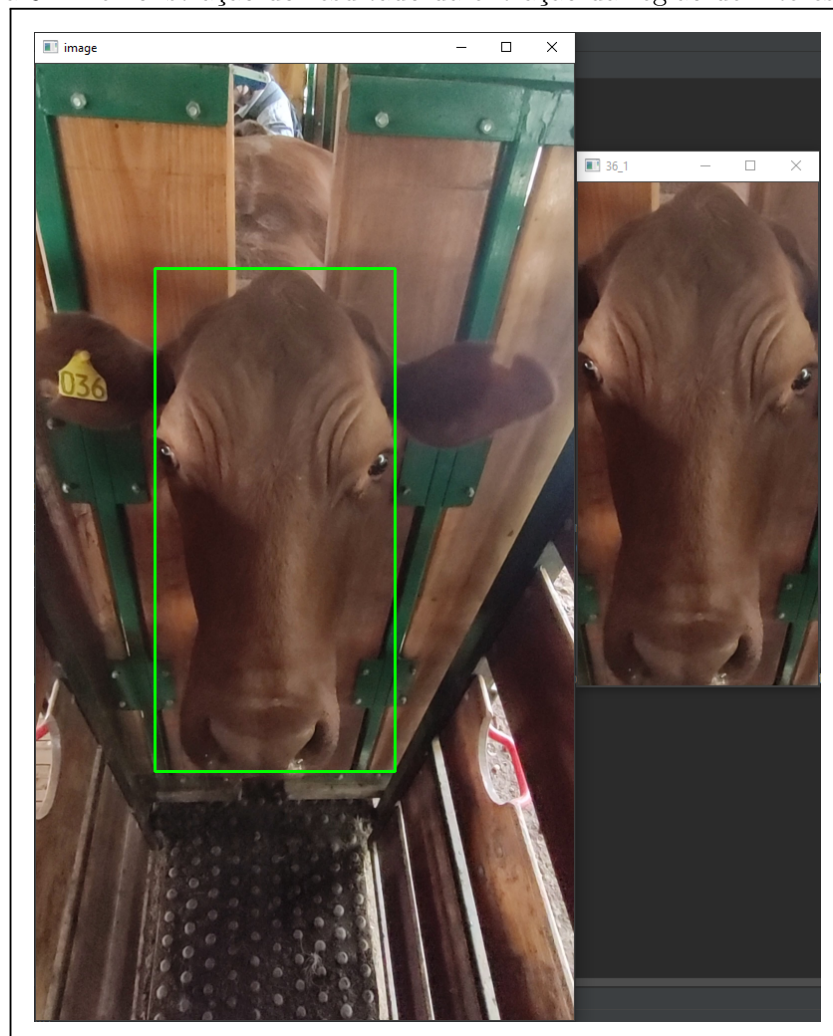


Fonte: O autor (2020)

³O erro quadrático médio (MSE – *mean square error*) é usualmente tido como uma figura de mérito inadequada para comparação de padrões em imagem, entretanto, para aplicações em que as imagens são muito parecidas, o MSE é uma métrica adequada e largamente aplicada (COSTA; BERMUDEZ, 2007; ELAD; FEUER, 1999).

Após, são definidos manualmente dois pares de coordenadas que correspondam ao retângulo que delimita a região de interesse. Esses pontos de coordenadas são armazenados em um arquivo separado, identificando claramente os pontos, a qual animal e a qual aquisição (*frame*) pertencem. Essa etapa de verificação e extração de coordenadas é feita com auxílio de um *script* feito em Python, que, basicamente, varre o diretório que contém as imagens e permite, com o mouse, desenhar um retângulo da região de interesse. Depois de desenhado, o *script* analisa a região e, caso necessário, faz um ajuste na altura do recorte, de modo a manter sempre uma mesma proporção em todos recortes. Essa proporção foi encontrada utilizando uma média do tamanho que a face do animal ocupa nas diferentes aquisições da base, resultando em uma proporção de $1,66L \times L$, em que L é a largura (base) do retângulo. Após isso, é mostrada uma prévia do recorte desta região da imagem, conforme ilustrado na Figura 6. O *script* salva automaticamente as coordenadas do retângulo e também o recorte da imagem, já nomeados de acordo com a imagem original.

Figura 6 – Demonstração do resultado da extração da região de interesse



Fonte: O autor (2020)

Para este trabalho, foram coletados vídeos de 31 animais diferentes, das raças Brangus e Braford, ao longo de 6 meses, o que totalizou 26261 imagens brutas (sem tratamento ou região de interesse extraída). Destes, alguns foram descartados, por terem baixo aproveitamento de imagens por conta de vários motivos. As primeiras aquisições de imagens foram feitas com vídeos relativamente curtos, em torno de 10 a 20 segundos. Porém, notou-se que esse tempo agrava o problema de baixo aproveitamento das imagens do animal. Aquisições subsequentes foram feitas com duração média de 1 minuto. Apesar de isso gerar um número maior de *frames* descartados também se aumenta a quantidade de *frames* disponíveis e a probabilidade de se conseguir boas quantidades de imagens aproveitáveis.

A validação do conjunto de dados é realizada utilizando-se o método PCA, visto que esse é um método tradicionalmente utilizado no reconhecimento de faces humanas e já utilizado na avaliação de outras bases, como a de Gao et al. (2007) e, também, é a base para outras técnicas de reconhecimento facial humano como a técnica *Eigenfaces*, de Turk e Pentland (1991).

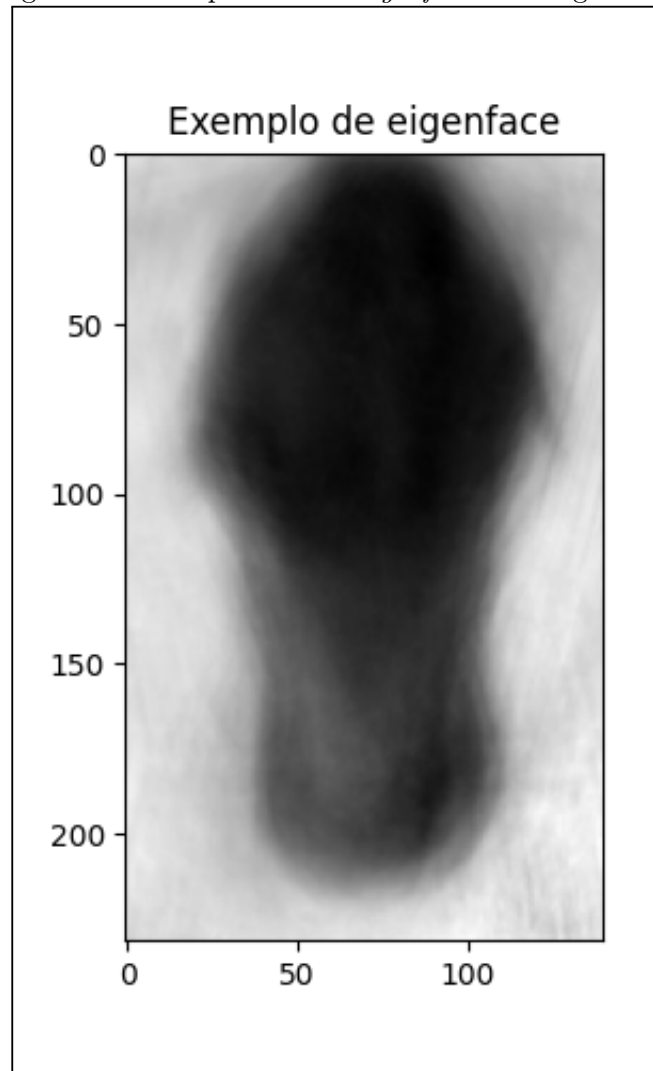
3.2 ANÁLISE DE COMPONENTES PRINCIPAIS

A análise de componentes principais, ou PCA, é uma técnica de redução dimensional baseada em estatística que aumenta a interpretabilidade, enquanto reduz a perda de informações. Por ser um conceito relativamente simples e versátil, a PCA é amplamente utilizada para análise de dados em diversas áreas, inclusive processamento de imagens.(ABDI; WILLIAMS, 2010).

A PCA foi introduzida em 1901 por Karl Pearson e, posteriormente, desenvolvido de maneira independente por Harold Hotelling em 1939. A PCA é comumente utilizada quando se possui um *dataset* com muitas variáveis inter-relacionadas, de modo a reduzir sua dimensão enquanto procura manter o máximo de variação presente no *dataset* (JOLLIFFE, 2002). Seu uso para reconhecimento facial, foi proposto por Kirby e Sirovich em 1986, expandido e popularizado por Pentland e Turk em 1991 no que se conhece hoje como a técnica *Eigenfaces*. A versatilidade dessa técnica permite que ela seja utilizada tanto para problemas de reconhecimento facial quanto para problemas de detecção facial (TURK; PENTLAND, 1991). *Eigenfaces*, conforme definido pelo seus autores, é o nome dado para um autovetor, resultante do processo de PCA, exibido como uma imagem, conforme o exemplo ilustrado na Figura 7. Ou seja, a técnica do *Eigenfaces* pode ser, basicamente, dividida em duas partes: a primeira, logicamente, é a PCA; a segunda é a interpretação e utilização do resultado.

Considerando um problema de reconhecimento facial, uma imagem $I_{(x,y)}$ possui uma dimensão $N \times M$ e, para ela ser analisada, precisa ser reorganizada como um vetor bidimensional de intensidades, e considerando um valor médio de imagem como sendo 100×100

Figura 7 – Exemplo de uma *eigenface* de um gado



Fonte: O autor (2020)

pixels isso resulta em um vetor de 10000 posições. Imaginando agora que temos uma coleção de imagens, todas do mesmo tamanho citado, resulta em uma coleção de pontos no espaço muito grande. Imagens de faces, que contém uma configuração similar entre elas, ou seja, possuem um certo padrão, podem ser descritas por um sub-espaço menor, uma vez que não estarão distribuídas aleatoriamente em um espaço grande (TURK; PENTLAND, 1991). Esta redução é feita através do uso de PCA.

3.2.1 Notação e nomenclatura

Deste ponto em diante, a notação matemática será estabelecida como se segue. Matrizes serão representadas por letras maiúsculas em negrito. Vetores serão representados por letras minúsculas em negrito. Variáveis escalares serão representadas por letras minúsculas em itálico. Escalares que denotam limites e quantidades pré-definidas (constantes

dentro de uma solução) serão representadas por letras maiúsculas em itálico. Por exemplo: uma matriz \mathbf{A} com L linhas e C colunas possui como c -ésima coluna o vetor \mathbf{a}_c , em que $1 \leq c \leq C$.

Uma imagem pode ser representada de duas formas: a forma convencional (matricial) e por sua representação lexicográfica (vetorial). A organização dos pixels de uma imagem na forma lexicográfica consiste em organizar as colunas da matriz que compõe a imagem coluna abaixo de coluna, formando um vetor.

3.2.2 A determinação do espaço vetorial

Seja uma imagem de face (de gado), em sua representação lexicográfica, modelada como um vetor aleatório \mathbf{x} com dimensões $N \times 1$ (ou seja, uma imagem de N pixels vetorizada). A média desse vetor aleatório é então definida por

$$\boldsymbol{\mu}_{\mathbf{x}} = E[\mathbf{x}] \quad (3.1)$$

em que $E[\cdot]$ denota a esperança matemática. Seja \mathbf{X} uma matriz de dimensões $N \times M$ (assumidamente $M < N$) cujas colunas \mathbf{x}_m correspondem às M imagens de faces, com N pixels (linhas) cada uma, organizadas lexicograficamente:

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_3 \quad \dots \quad \mathbf{x}_M] \quad (3.2)$$

A média do conjunto de imagens que compõem \mathbf{X} pode ser interpretada como uma estimativa da média em (3.1) e é definida por

$$\hat{\boldsymbol{\mu}}_{\mathbf{x}} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m \quad (3.3)$$

Cada imagem do conjunto difere da média tal como $\hat{\boldsymbol{\phi}}_m = \mathbf{x}_m - \hat{\boldsymbol{\mu}}_{\mathbf{x}}$.

A redução de dimensão desse conjunto pode ser realizada por meio da PCA. Para isso, de forma resumida, deve-se encontrar um conjunto de M autovetores e autovalores associados da matriz de auto-covariância do conjunto de imagens \mathbf{X} .

A matriz de auto-covariância é uma matriz quadrada, $\mathbf{C}_{\mathbf{xx}}$, que contém as variâncias e covariâncias do conjunto de dados (imagens de faces, no caso). Os elementos na diagonal principal de $\mathbf{C}_{\mathbf{xx}}$ são as variâncias e os elementos fora dessa diagonal são as covariâncias (MONTGOMERY; RUNGER, 2016). Essa matriz é definida por

$$\mathbf{C}_{\mathbf{xx}} = E [(\mathbf{x}_m - \boldsymbol{\mu}_{\mathbf{x}})(\mathbf{x}_m - \boldsymbol{\mu}_{\mathbf{x}})^T] = E [\boldsymbol{\phi}\boldsymbol{\phi}^T] \quad (3.4)$$

que pode ser estimada pela matriz de auto-covariância do conjunto de dados, conforme

$$\hat{\mathbf{C}}_{\mathbf{xx}} = \frac{1}{M} \sum_{m=1}^M \hat{\phi}_m \hat{\phi}_m^T \quad (3.5)$$

Essa equação equivale-se a

$$\hat{\mathbf{C}}_{\mathbf{xx}} = \hat{\Phi} \hat{\Phi}^T \quad (3.6)$$

em que $\hat{\Phi} = \begin{bmatrix} \hat{\phi}_1 & \hat{\phi}_2 & \cdots & \hat{\phi}_M \end{bmatrix}$. Considerando-se que $\hat{\mathbf{C}}_{\mathbf{xx}}$ possui dimensão $(N \times N)$, a determinação dos N autovetores e autovalores é geralmente excessivamente custosa do ponto de vista computacional, visto que o número de pixels em uma imagem é geralmente elevado (TURK; PENTLAND, 1991). Considerando que $\hat{\mathbf{C}}_{\mathbf{xx}}$ é composta por apenas $M < N$ imagens, o posto dessa matriz se restringe a M . Sendo assim, dos N autovetores, apenas M são associados a autovalores não nulos. Esses autovetores, por sua vez, podem ser obtidos a partir dos M autovetores \mathbf{v}_m da matriz (TURK; PENTLAND, 1991)

$$\hat{\mathbf{C}}_{\mathbf{yy}} = \hat{\Phi}^T \hat{\Phi} \quad (3.7)$$

com dimensões $(M \times M)$. Note-se que, sendo

$$\hat{\mathbf{C}}_{\mathbf{xx}} = \hat{\Phi} \hat{\Phi}^T = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \quad (3.8)$$

em que \mathbf{U} e $\mathbf{\Lambda}$ são respectivamente as matrizes de autovetores e autovalores de $\hat{\mathbf{C}}_{\mathbf{xx}}$, segue que, pós-multiplicando-se ambos os lados de (3.8) por \mathbf{U} , chega-se em

$$\hat{\Phi} \hat{\Phi}^T \mathbf{U} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \mathbf{U} \quad (3.9)$$

Sendo que os autovetores compõem uma matriz ortonormal, $\mathbf{U}^T = \mathbf{U}^{-1}$, e portanto

$$\hat{\Phi} \hat{\Phi}^T \mathbf{U} = \mathbf{U} \mathbf{\Lambda} \quad (3.10)$$

Separando-se o cálculo de (3.10) em uma equação por autovetor, chega-se em

$$\hat{\Phi} \hat{\Phi}^T \mathbf{u}_m = \lambda_m \mathbf{u}_m \quad (3.11)$$

Pré-multiplicando-se, agora ambos os lados de (3.11) por $\hat{\Phi}^T$, tem-se

$$\hat{\Phi}^T \hat{\Phi} (\hat{\Phi}^T \mathbf{u}_m) = \lambda_m (\hat{\Phi}^T \mathbf{u}_m) \quad (3.12)$$

Por inspeção, é possível notar que, similarmente a (3.11), a Equação (3.12) equivale a dizer que

$$\mathbf{v}_m = \hat{\Phi}^T \mathbf{u}_m \quad (3.13)$$

corresponde à equação de autovetores e autovalores de $\hat{\Phi}^T \hat{\Phi}$.

A determinação dos M autovetores \mathbf{u}_m pode ser feita, com menor custo computacional, a partir dos M autovetores \mathbf{v}_m , pode ser obtida pré-multiplicando-se ambos os lados de (3.13) por $\hat{\Phi}$, o que leva a

$$\mathbf{u}_m = \hat{\Phi} \mathbf{v}_m \quad (3.14)$$

3.2.3 Determinação do sub-espço de faces

Considerando as semelhanças dos padrões pertencentes a uma mesma classe, é razoável esperar que eles se encontrem em uma mesma região do espaço. Sendo assim, um número reduzido de autovetores é provavelmente suficiente para delimitar o sub-espço em que se concentram os indivíduos de tal classe. Por exemplo, Sirovich e Kirby (1987) em seus estudos encontraram que em um universo de $M = 115$ imagens de homens brancos, 40 *Eigenfaces* eram suficientes para uma boa descrição do conjunto. Uma forma de se chegar em um valor de componentes adequado é utilizar a variância total explicada. Ela fornece a medida (percentual) da contribuição de cada autovetor na representação dos dados do conjunto. A variância explicada por cada i -ésimo autovetor é calculada a partir do respectivo autovalor, conforme

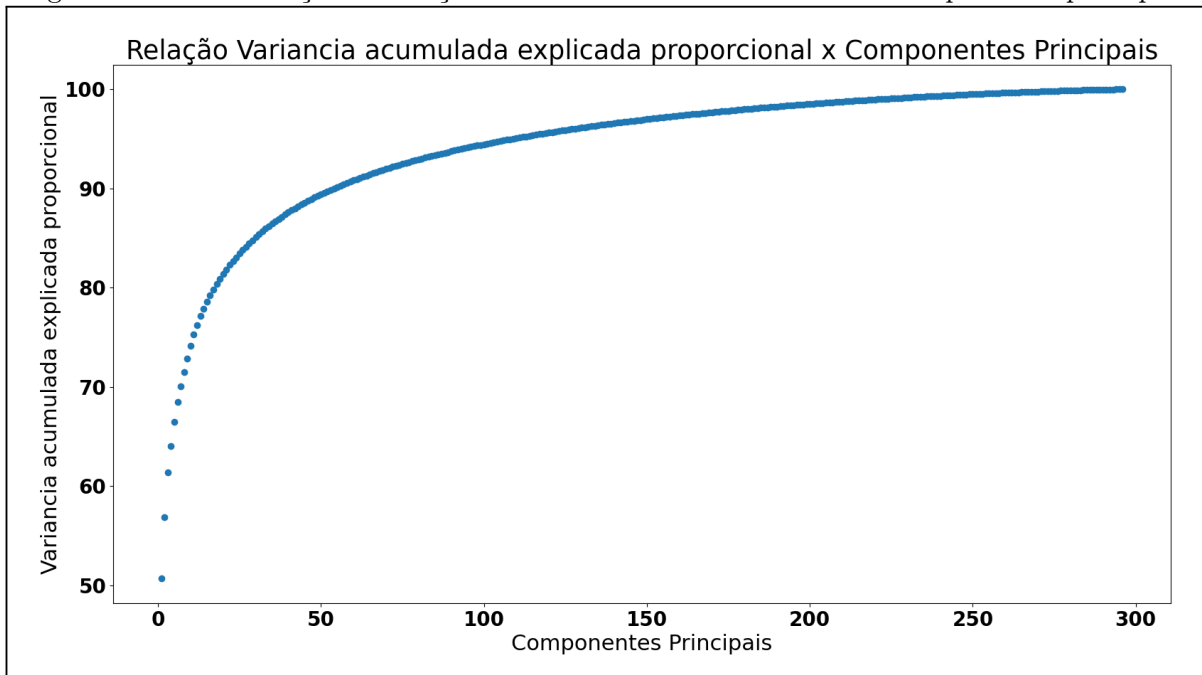
$$Var(\lambda_i) = \frac{\lambda_i}{\sum_{m=1}^M \lambda_m} 100 \quad (3.15)$$

Ou seja, divide-se cada autovalor λ_i pela soma de todos autovalores, multiplicando por 100 para obter o valor em porcentagem. O interessante desse cálculo é que ele evidencia o fato de não se necessitar todos autovalores para descrever o conjunto. A variância total explicada vai ser a soma acumulada das variâncias explicadas por cada autovetor (organizados pela ordem decrescente de seus autovalores) e pode ser visualizada graficamente, de modo a facilitar o entendimento, conforme exemplo na Figura 8. No eixo das abscissas tem-se a quantidade de componentes principais disponíveis, enquanto no eixo das ordenadas tem-se a variância acumulada proporcional. Nota-se que quanto mais componentes são considerados, mais se tende a 100% de explicação. Note-se ainda que mais de 90% da variância dos dados é explicada com menos de 30 % dos autovetores. Logo, com 150 componentes principais, de um universo de 300, já se consegue caracterizar quase 100% do conjunto, o que resulta em uma redução do custo de processamento. Jolliffe (2002) comenta que valores entre 70% e 90% de explicação dos dados já são o suficiente para obter bons resultados.

Sabendo a quantidade total de autovalores que será usada, pode-se agora determinar os pesos que representam a projeção de uma imagem \mathbf{i} no sub-espço M' -dimensional de acordo com

$$w_k = \mathbf{u}_k^T (\mathbf{i} - \hat{\psi}) \quad k = 1, \dots, M' \quad (3.16)$$

Figura 8 – Demonstração da relação da variância acumulada com os componentes principais



Fonte: O autor (2020)

em que \mathbf{u}_k corresponde ao autovetor de $\hat{\mathbf{C}}_{\mathbf{x}\mathbf{x}}$ associado ao k -ésimo maior autovalor. Os pesos w_k são então organizados em um vetor $\boldsymbol{\omega} = [w_1, w_2, \dots, w_{M'}]^T$ que descreve a contribuição de cada *Eigenface* para representar a imagem. O cálculo desse vetor, para todas as imagens do conjunto de treinamento, encerra a fase de treinamento do algoritmo.

3.2.4 Classificação de uma imagem

O processo para se classificar uma imagem, utilizando os resultados das seções anteriores, é bastante direto. Sendo \mathbf{i} a imagem a ser classificada, ela deve ser projetada dentro do sub-espaco M' , chamado de (sub-)espaco de faces, como discutido sobre a Equação (3.16), obtendo-se um vetor $\boldsymbol{\omega}$. Feito isso, esse vetor pode ser comparado com um vetor pré-definido para uma c -ésima classe (indivíduo), $\boldsymbol{\omega}_c$. O vetor $\boldsymbol{\omega}_c$ pode ser definido, por exemplo, pela projeção da média das imagens do conjunto de treinamento referentes àquela classe.

Existem diferentes formas de se determinar a qual classe a imagem pertence. A mais simples (e adotada neste trabalho) é determinar qual classe minimiza a distância euclidiana entre os vetores $\boldsymbol{\omega}$ e $\boldsymbol{\omega}_c$, conforme:

$$\epsilon_c^2 = \|(\boldsymbol{\omega} - \boldsymbol{\omega}_c)\|^2 \quad (3.17)$$

3.3 DIVISÃO E TREINAMENTO DO CONJUNTO

O treinamento é realizado dividindo-se a base de imagens em proporções pré-determinadas e aplicando sobre elas o algoritmo das *Eigenfaces*. O algoritmo elaborado neste trabalho foi feito de forma que se possa alterar esta proporção passando um parâmetro para uma função, o que permite que se possa observar qual efeito uma proporção maior ou menor produz no resultado final da classificação. A base é dividida em um conjunto de treinamento e um de teste, seguindo o padrão do método de *random subsampling*, ou seja, o procedimento descrito a seguir é repetido R vezes, sendo R um número inteiro.

A seleção das imagens a serem utilizadas para o treinamento é feita de forma aleatória, utilizando um selecionador de números aleatórios que existe de maneira nativa em *Python*. É importante ressaltar que, devido a natureza das imagens coletadas, algumas classes possuem mais imagens que outras. Em virtude disso, decidiu-se ignorar as classes que possuem menos que 10 imagens. Para realizar a divisão, varre-se o diretório onde estão as classes. Cada classe é um diretório que contém as imagens relativas a ela e, nesse caso, o nome dela é relativo ao número do brinco do animal. As imagens são então convertidas para escala de cinza. Por estar sendo utilizado *Python*, se faz uso de seus dicionários, que armazenam variáveis em par chave-valor, assim consegue-se atrelar cada classe a suas respectivas imagens com um acesso rápido e prático. A decisão da porcentagem de amostras que irão compor o conjunto de treino e de teste é, na maior parte, arbitrária, não existindo uma maneira objetiva de se determinar (SKANSI, 2018).

Após a divisão e obtenção dos dicionários de treinamento e teste, se processa a parte de treinamento de modo a equalizar o tamanho das imagens. Isso é feito buscando-se o tamanho da menor imagem no subconjunto, desta forma minimiza-se eventuais distorções que possam ocorrer quando se aumenta uma imagem muito pequena e harmoniza o subconjunto. Além disso, a PCA opera, essencialmente, em vetores, ou seja, cada imagem, que é uma matriz, precisa ser transformada em um vetor unidimensional. As imagens selecionadas para serem utilizadas no treinamento serão armazenadas em um diretório específico, dessa forma é possível utilizar as mesmas imagens para visualizar o impacto de outros parâmetros, como um número diferente de autovetores, por exemplo.

4 RESULTADOS E DISCUSSÕES

De posse das imagens das regiões de interesse, obtidas na Seção 3.1, e da Análise de Componentes Principais, discutida na Seção 3.2, foi elaborado um algoritmo base com o intuito de verificar a usabilidade da base construída. Também busca-se verificar como um algoritmo projetado para funcionar com características e faces humanas se comporta para o problema de reconhecimento facial bovino.

4.1 BASE RESULTANTE

A base resultante possui 31 classes (animais) com 783 imagens de faces extraídas no total. As classes possuem quantidades variadas de faces. Cada face possui um arquivo texto relacionado a ela contendo as coordenadas da região de interesse na imagem original e, também, cada par (face e arquivo texto) está nomeado de acordo com qual imagem original ela corresponde para aquela classe.

As classes são nomeadas de acordo com o brinco do animal que originou as imagens, podendo ser utilizada como saída do algoritmo. A base estará disponível publicamente, no momento, no Google Drive¹ e possui tamanho total de 12 *Giga Bytes*.

4.2 MÉTRICAS UTILIZADAS PARA AVALIAÇÃO DO DESEMPENHO

Para avaliação dos resultados que se seguem são consideradas as seguintes figuras de mérito: Precisão, *Recall* ou Sensibilidade, *F-Score* e Matriz de Confusão. A matriz de confusão é uma matriz quadrada de tamanho $C \times C$, onde C é a quantidade de classes do sistema. Nesta matriz se organizam os resultados em positivos verdadeiros, falsos positivos, negativos verdadeiros e falsos negativos de uma maneira bem visual e de fácil compreensão. A Tabela 1 ilustra a construção da matriz para um caso binário (SKANSI, 2018).

Tabela 1 – Funcionamento da Matriz de Confusão

	Classe prevista A	Classe prevista B
Classe real A	Número de positivos verdadeiros	Número de falsos positivos
Classe real B	Número de falsos negativos	Número de negativos verdadeiros

Fonte: Adaptado de SKANSI (2018)

A precisão nos diz como o modelo é eficaz em evitar alarmes falsos, ou seja, qual fração das predições positivas foram de fato positivas (KUBAT, 2015). A precisão de um

¹Acesso para a base construída no Google Drive: <https://drive.google.com/drive/folders/1QAk42DgjRMaLt-Eh0XXR-mgr2uOutuoH?usp=sharing>

classificador pode ser calculada através da seguinte fórmula

$$P = \frac{\textit{PositivosVerdadeiros}}{\textit{PositivosVerdadeiros} + \textit{FalsosPositivos}} \quad (4.1)$$

O *recall*, ou sensibilidade, nos indica a razão de verdadeiros positivos do modelo (KUBAT, 2015). Também é chamada de taxa de verdadeiros positivos e pode-se entender que ela indica a probabilidade de um positivo real ser identificado positivamente pelo modelo. Calcula-se da seguinte forma

$$R = \frac{\textit{PositivosVerdadeiros}}{\textit{PositivosVerdadeiros} + \textit{FalsosNegativos}} \quad (4.2)$$

Pode-se notar que o *recall* e a precisão tem a diferença apenas no denominador. Isso acontece por que a precisão é a frequência de verdadeiros positivos entre todas as predições positivas, enquanto o *recall* trata dos mesmos verdadeiros positivos, mas comparados com os exemplos positivos no conjunto (KUBAT, 2015).

O *F-Score* é uma medida que combina o *recall* e a precisão e foi definida por Chinchor (1992). Seu valor é calculado da seguinte forma

$$FS = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R} \quad (4.3)$$

em que β é um peso dado à importância do *recall* sobre a precisão (CHINCHOR, 1992). Para este trabalho, se considerará um peso $\beta = 1$, ou seja, igual importância para ambas as medidas, como normalmente se é feito quando não se tem certeza de qual é mais importante que o outro (KUBAT, 2015). O *F-Score* carrega consigo uma ênfase maior no impacto dos falsos negativos e falsos positivos, evidenciado pela forma que o *recall* e a precisão são calculados, diferentemente da acurácia, que dá ênfase aos verdadeiros positivos e negativos.

4.3 AVALIAÇÃO DO DESEMPENHO DA BASE

De modo a se analisar a performance do conjunto de uma forma mais ampla, se realizou diversos treinamentos em cenários distintos alterando-se alguns parâmetros e observando o comportamento. Para este trabalho, realizou-se simulações com 40, 50, 75 e 90% do conjunto para treinamento. Realizou-se então, para cada uma das simulações, $R = 10$ repetições com essa proporção, seguindo o método de *random subsampling*, usando um limiar da variância explicada para encontrar o valor de M' de 95%. Os resultados estão demonstrados na Tabela 2.

Nota-se que mesmo com uma divisão não exatamente ideal de 40%, uma vez que o conjunto de treino é menor que o de testes, o modelo conseguiu bons números e que,

Tabela 2 – Médias após 10 repetições, com variadas divisões e limiar 95%

Tam. Treino	Precisão	Recall	F-Score
40%	91.35%	92.95%	0.916
50%	94.17%	95.64%	0.945
75%	94.60%	96.26%	0.950
90%	96.56%	97.20%	0.963

Fonte: O autor (2020)

conforme aumenta as amostras de treinamento, os resultados melhoram consistentemente.

Ao se aumentar as repetições de treino, no caso para $R = 20$, percebe-se que os ganhos de performance são mínimos, como mostra a Tabela 3. Uma explicação para isso é que, como o conjunto é relativamente pequeno com pouco mais de 700 imagens, utilizando 10 repetições já se consegue utilizar praticamente todas as imagens para treinar e testar o conjunto. Isso indica que para o caso deste conjunto de dados o ideal é se utilizar 10 repetições, no máximo, de modo a se ter um balanço entre velocidade de computação e performance do modelo.

Tabela 3 – Médias após 20 repetições, com variadas divisões e limiar 95%

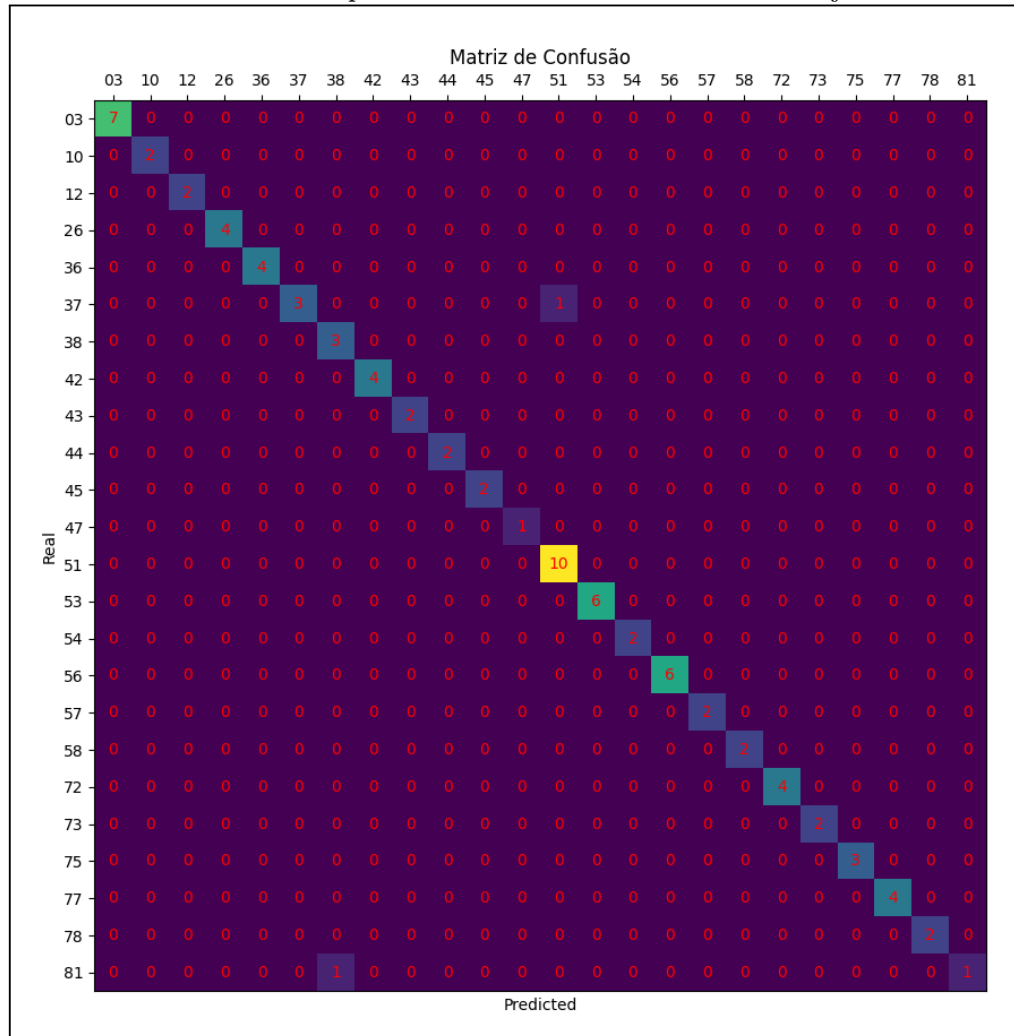
Tam. Treino	Precisão	Recall	F-Score
40%	91.63%	93.43%	0.920
50%	93.35%	94.83%	0.936
75%	95.05%	96.39%	0.951
90%	96.61%	97.70%	0.966

Fonte: O autor (2020)

Pegando-se o melhor resultado, no caso divisão de 90% do conjunto para treino e repetições $R = 10$, e observando-se graficamente a matriz de confusão na Figura 9 nota-se algumas particularidades que as métricas demais figuras de mérito não explicitam. Como o conjunto é não uniforme, ou seja, as classes são irregulares e possuem quantidades de animais diferentes entre si, uma divisão agressiva como essa pode não ser ideal para uma aplicação real. Pela matriz de confusão pode-se perceber que existem classes com apenas 2 animais no conjunto de teste, como a classe 10, por exemplo.

Comparando-se a matriz da Figura 9 com a matriz da Figura 10, que considera um conjunto de treinamento utilizando 75% das imagens, apesar de ainda se tratar de um conjunto desbalanceado, com algumas classes contendo um número muito reduzido de imagens, já se mostra mais promissor. As figuras de mérito consideradas, para este caso, apresentam resultados relativamente próximos do caso de 90%. Imaginando-se um cenário em que se tenham milhares de cabeças de gado, por exemplo, esse tipo de cenário pode ser mais útil para se visualizar como se ajustar melhor os parâmetros do modelo, de modo a fazer ele ter uma resposta melhor, ou ainda, em quais pontos realizar melhorias no próprio conjunto de dados para mitigar esses problemas.

Figura 9 – Matriz de confusão para um treinamento com 90% do conjunto em treino



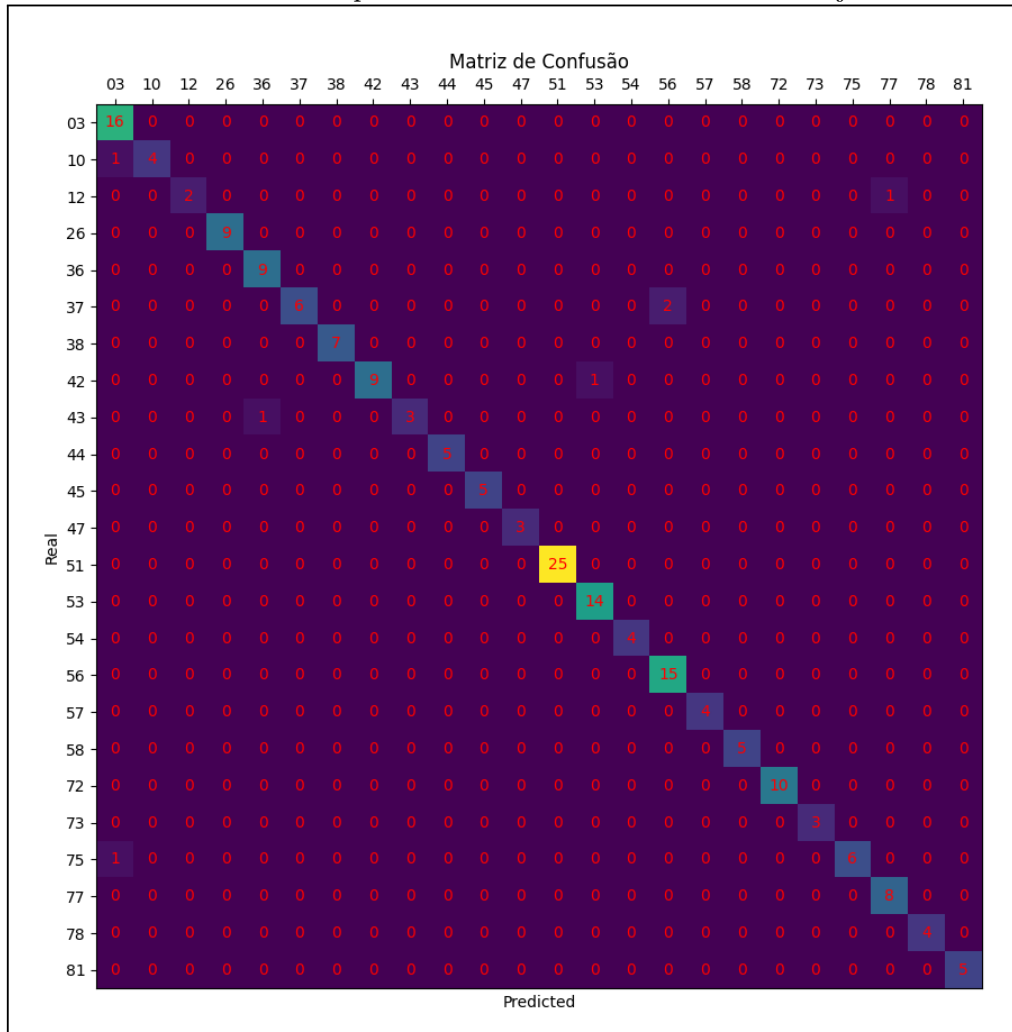
Fonte: O autor (2020)

4.4 COMPARAÇÃO DOS RESULTADOS

Apesar de muitos trabalhos nesta área utilizarem aproximações por vezes muito específicas ou se utilizando fortemente de redes neurais, KUMAR et al. (2017b) utilizaram uma solução similar com a proposta deste trabalho. Ainda que sejam trabalhos com objetivos distintos, KUMAR et al. (2017b) também utilizaram imagens frontais, com algumas variações de pose². Eles ressaltam a importância de se ter variações de pose e iluminação de um ambiente não controlado, tratando isso como forma benéfica de se aumentar a variância dos dados dentro da própria classe. Como a proposta dos autores é um sistema completo de identificação, eles também propõem técnicas de pré-processamento das imagens, como um filtro passa-baixas baseado em pirâmide Gaussiana, para redução de ruídos na imagem, e uma técnica conhecida como equalização adaptativa de histograma

²Pose, neste caso, se trata da posição do rosto do animal. Ou seja, o ângulo de inclinação da cabeça, para cima ou para baixo e para os lados.

Figura 10 – Matriz de confusão para um treinamento com 75% do conjunto em treino



Fonte: O autor (2020)

limitada por contraste, conhecida pela sigla em inglês CLAHE — *Contrast Limited Adaptive Histogram Equalization*, que é utilizada para melhorar o contraste da imagem. Não fica exatamente claro em sua publicação se ele capta as imagens no brete, porém, pelos exemplos que ele mostra, assume-se que não, como pode ser visto na Figura 11.

O conjunto de dados montado por KUMAR et al. (2017b) possui 500 animais, de diferentes raças, e 10 imagens de cada animal totalizando 5000 imagens de faces. Para analisar a performance, os autores utilizaram 40% das imagens para treinamento e 60% para teste e realizaram validação cruzada cinco vezes. Os resultados obtidos pelos autores para diferentes algoritmos de reconhecimento facial estão demonstrados na Tabela 4. Para estes resultados, os autores consideram diferentes níveis gaussianos de filtragem, o que segundo eles reduz o número de *Eigenfaces* escolhidas para cada nível.

É importante notar na Tabela 4, diferente do que foi utilizado neste trabalho, KUMAR et al. (2017b) utiliza acurácia como forma de apresentar seus resultados e eficácia do seu sistema. A acurácia avalia a frequência de avaliações corretas feitas pelo classificador

Figura 11 – Exemplos de faces de gado



Fonte: (KUMAR et al., 2017b)

(KUBAT, 2015). Os autores também, em nenhum momento, indicam exatamente quantas *Eigenfaces* utilizaram para o treinamento do algoritmo de PCA. Considerando todas essas informações, o algoritmo proposto neste trabalho foi parametrizado de forma similar, para que, na medida do possível, se possa fazer uma comparação justa dos desempenhos obtidos pelo PCA em ambas as bases. Sendo assim, considerou-se: i) 40% do conjunto para teste e 60% para treino; ii) 5 repetições de *random subsampling*; e iii) Limiar da variância explicada, para encontrar a quantidade de *Eigenfaces*, mínimo de 60% e máximo de 90%, como forma de emular os 4 níveis gaussianos. A Tabela 5, demonstra a acurácia obtida para cada cenário.

Como é possível observar, a base proposta consegue se comportar em níveis melhores que o sistema de KUMAR et al. (2017b). Uma explicação possível para isso, é que a base proposta, de certa forma, mitiga de forma natural alguns problemas possíveis de serem encontrados em um ambiente não controlado, como por exemplo, enquadramento do rosto do animal. Por conta da natureza e funcionalidade do brete, os animais já ficam relativamente mais imóveis e em uma posição mais ideal e padronizada. Em contrapartida,

Tabela 4 – Acurácias para diferentes métodos de reconhecimento

Nível Gaussiano	PCA	LDA	ICA
1	74.39%	75.57%	79.75%
2	79.81%	80.64%	82.95%
3	81.89%	84.19%	84.90%
4	83.86%	85.95%	86.95%

Fonte: Adaptado de (KUMAR et al., 2017b)

Tabela 5 – Acurácias para diferentes limiares

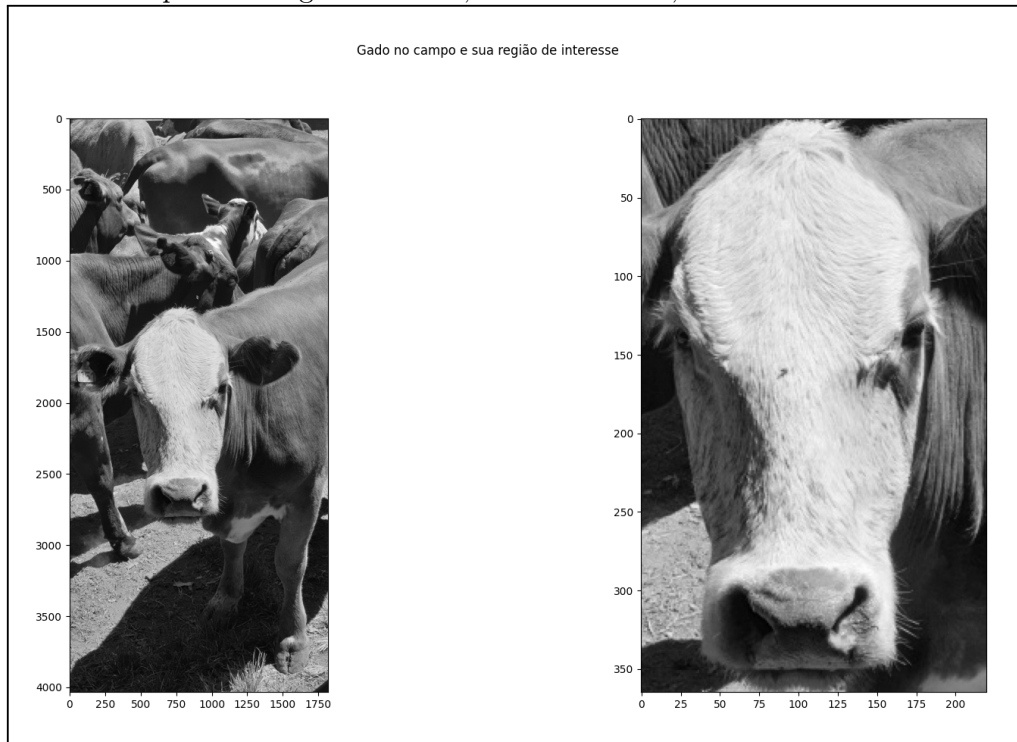
Limiar	Acurácia
60%	97.87%
70%	99.13%
80%	99.29%
90%	99.37%

Fonte: O autor(2020)

isto reduz, relativamente, a variância dos dados do conjunto, com a principal variável se tornando a iluminação, que não foi tratada em pré-processamento na imagem antes de ela ser utilizada pelo algoritmo. Como não se tem acesso integral a base produzida por KUMAR et al. (2017b), apenas pode-se especular sobre a variância do conjunto coletado por eles, levando em conta o texto escrito e os dados apresentados no artigo.

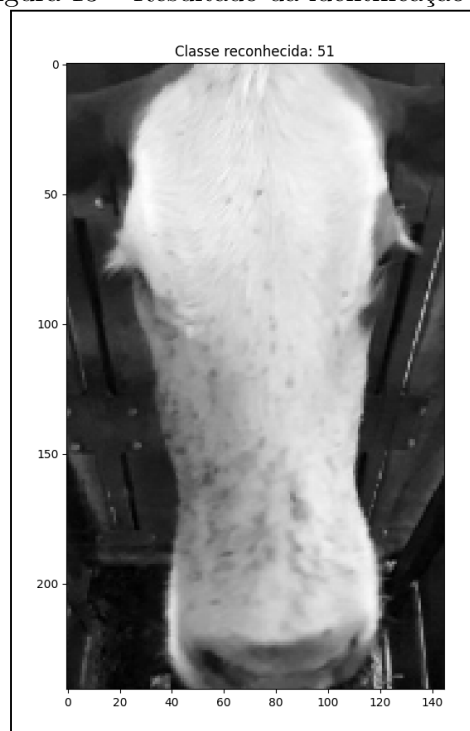
As Figuras 12 e 13 ilustram um exemplo de identificação para uma condição de aquisição substancialmente distinta àquela utilizada na construção da base de dados. A imagem, após passar pelos mesmos procedimentos para a extração da região de interesse, foi projetada no espaço de faces obtido como resultado do treinamento do algoritmo de PCA e, subsequentemente, identificada. Como é possível observar, mesmo em ambientes diferentes, o algoritmo treinado conseguiu obter sucesso em identificar o animal. Isto demonstra que a base consegue ser suficientemente rica a ponto de permitir generalizações dessa natureza.

Figura 12 – Exemplo de um gado da base, neste caso o 51, em ambiente não controlado



Fonte: O autor (2020)

Figura 13 – Resultado da identificação



Fonte: O autor (2020)

5 CONCLUSÕES

Nos últimos anos, a busca por métodos de visão computacional e reconhecimento de padrões aplicados à biometria animal tem sido bastante visada por diversos fatores, como evolução da tecnologia e demanda do mercado. As dificuldades de obtenção de imagens úteis para este tipo de validação é um dos principais problemas a serem solucionados. Animais são imprevisíveis quando se trata de seu movimento e reação a uma captura de imagem. Diferentes métodos foram propostos, com pouco consenso sobre um padrão. Mesmo com todos estes obstáculos, a biometria animal é uma tecnologia promissora e com um escopo de aplicação vasto.

Por conta de fatores como a distância até a fazenda parceira e a pandemia que assola o mundo desde 2020, a obtenção de imagens de maneira mais controlada e padronizada foi um obstáculo bastante complicado de se contornar. Em virtude disso, o conjunto de imagens construído e apresentado neste trabalho, bem como a aplicação do mesmo, ainda podem ser aprimorados. A principal fraqueza notada na base proposta é, aparentemente, a baixa variância dos dados. Uma forma de se aprimorar isto foi o aumento da base de forma sintética, a partir de transformações geométricas, filtragens e geração de padrões de iluminação a cada imagem. Porém, por questão de tempo e escopo, essa possibilidade não foi testada.

Outro passo importante que pode ser dado, é o treinamento de um algoritmo de detecção. Ele poderia auxiliar na obtenção de mais imagens de faces em poses variadas e aumentar a variância dos dados e, também, é parte vital de um sistema automático de reconhecimento comercial. Porém, neste trabalho foi possível mostrar a viabilidade do conjunto e uma metodologia de captura que pode solucionar alguns dos problemas encontrados por outros autores.

O conjunto de imagens obtido está publicado, contendo todas as imagens originais, os recortes de faces obtidos através da seleção manual e arquivos de texto indicando as coordenadas da região de interesse desses recortes. O objetivo é fornecer diferentes estilos de dados, como forma de se incentivar o uso de aproximações diferentes para reconhecimento e detecção facial dos animais.

A motivação de se construir uma base de dados pública se dá com o intuito de incentivar a pesquisa e a comparação de resultados, de modo a se buscar uma solução que se adéque a demandas do mercado de uma maneira não invasiva e com boa relação custo-benefício. O compartilhamento de conhecimento é essencial para a evolução tecnológica, científica e, também, da sociedade. Desta forma, julga-se ser de suma importância que as universidades e governos incentivem cada vez mais pesquisas que buscam tecnologias e formas de tornar as vidas humana e animal mais confortáveis e saudáveis.

REFERÊNCIAS

- ABDI, H.; WILLIAMS, L. J. Principal component analysis. **Wiley interdisciplinary reviews: computational statistics**, [S.l.], v. 2, n. 4, p. 433–459, 2010.
- ABNT. Nbr iso 9000:2015: sistemas de gestão da qualidade - fundamentos e vocabulário. , [S.l.], 2015.
- ALLFLEX. **Identificação eletrônica**. Acessado em 05 out. 2020, <http://www.allflex.com.br/identificacao-eletronica>.
- ANCIENT. **Animal husbandry**. Acessado em 05 out. 2020, https://www.ancient.eu/Animal_Husbandry/.
- BELHUMEUR, P.; HESPANHA, J.; KRIEGMAN, D. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], v. 19, n. 7, p. 711–720, 1997.
- BERGAMINI, L. et al. Multi-views embedding for cattle re-identification. **14th International Conference on Signal-Image Technology InternetBased Systems (SITIS)**, [S.l.], p. 184–191, Novembro 2018.
- BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. , [S.l.], p. 144–152, 1992.
- CAI, C.; LI, J. Cattle face recognition using local binary pattern descriptor. , [S.l.], p. 1–4, 2013.
- CHINCHOR, N. Muc-4 evaluation metrics. In: CONFERENCE ON MESSAGE UNDERSTANDING, 4., 1992, USA. **Proceedings...** Association for Computational Linguistics, 1992. p. 22–29. (MUC4 '92).
- COSTA, G. H.; BERMUDEZ, J. C. M. Statistical analysis of the lms algorithm applied to super-resolution image reconstruction. **IEEE Transactions on Signal Processing**, [S.l.], v. 55, n. 5, p. 2084–2095, 2007.
- ELAD, M.; FEUER, A. Super-resolution reconstruction of continuous image sequences. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (CAT. 99CH36348), 1999., 1999. **Proceedings...** [S.l.: s.n.], 1999. v. 3, p. 459–463 vol.3.
- FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, [S.l.], v. 7, n. 2, p. 179–188, 1936.
- GAO, W. et al. The cas-peal large-scale chinese face database and baseline evaluations. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, [S.l.], v. 38, n. 1, p. 149–161, 2007.
- GOV. **Rastreabilidade animal**. Acessado em 05 out. 2020, <https://www.gov.br/agricultura/pt-br/assuntos/saude-animal-e-vegetal/saude-animal/rastreabilidade-animal>.

IBGE. Acessado em 05 out. 2020,

<https://sidra.ibge.gov.br/tabela/3939#resultado>.

JOLLIFFE, I. T. **Principal component analysis**. [S.l.]: Springer, 2002.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE - VOLUME 2, 14., 1995, San Francisco, CA, USA. **Proceedings...** Morgan Kaufmann Publishers Inc., 1995. p. 1137–1143.

KUBAT, M. **An introduction to machine learning**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2015.

KUMAR, A.; KAUR, A.; KUMAR, M. Face detection techniques: a review. **Artificial Intelligence Review**, [S.l.], v. 52, p. 927–948, Agosto 2018.

KUMAR, S. et al. Real-time recognition of cattle using animal biometrics. **Journal of Real-Time Image Processing**, [S.l.], v. 13, n. 3, p. 505–526, 2017.

KUMAR, S. et al. Recognition of cattle using face images. **Animal Biometrics**, [S.l.], p. 79–110, 2017.

KUMAR, S.; TIWARI, S.; SINGH, S. K. Face recognition for cattle. , [S.l.], p. 65–72, 2015.

M. PARISA BEHAM, S. M. M. R. A review of face recognition methods. **International Journal of Pattern Recognition and Artificial Intelligence**, [S.l.], v. 27, Júlio 2013.

MELDAU, D. C. Rastreabilidade bovina. **InfoEscola**, [S.l.], 2006.

MONTGOMERY, D. C.; RUNGER, G. C. **Estatística aplicada e probabilidade para engenheiros**. [S.l.]: GEN, 2016.

Osuna, E.; Freund, R.; Girosit, F. Training support vector machines: an application to face detection. , [S.l.], p. 130–136, 1997.

ROSLI, M. M.; TEMPERO, E.; LUXTON-REILLY, A. Evaluating the quality of datasets in software engineering. **Advanced Science Letters**, [S.l.], v. 24, n. 10, p. 179–188, October 2018.

SIROVICH, L.; KIRBY, M. Low-dimensional procedure for the characterization of human faces. **Journal of the Optical Society of America. A, Optics and image science**, [S.l.], v. 4, p. 519–24, 04 1987.

SKANSI, S. **Introduction to deep learning: from logical calculus to artificial intelligence**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2018.

TURK, M. A.; PENTLAND, A. P. Face recognition using eigenfaces. , [S.l.], p. 586–587, 1991.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. , [S.l.], 2001.

- WEBB, A. R.; COPSEY, K. D. **Statistical pattern recognition**. [S.l.]: Wiley, 2011.
- ZIN, T. T. et al. Image technology based cow identification system using deep learning. , [S.l.], v. 1, p. 236–247, 2018.

ANEXO A — TABELA DE VALORES DO *IRIS FLOWER DATASET*

<i>Iris setosa</i>				<i>Iris versicolor</i>				<i>Iris virginica</i>			
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
5.0	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3.0	5.8	2.2
5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3.0	6.6	2.1
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5.0	3.4	1.5	0.2	4.9	2.4	3.3	1.0	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
5.4	3.7	1.5	0.2	5.0	2.0	3.5	1.0	6.5	3.2	5.1	2.0
4.8	3.4	1.6	0.2	5.9	3.0	4.2	1.5	6.4	2.7	5.3	1.9
4.8	3.0	1.4	0.1	6.0	2.2	4.0	1.0	6.8	3.0	5.5	2.1
4.3	3.0	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5.0	2.0
5.8	4.0	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
5.4	3.9	1.3	0.4	5.6	3.0	4.5	1.5	6.5	3.0	5.5	1.8
5.1	3.5	1.4	0.3	5.8	2.7	4.1	1.0	7.7	3.8	6.7	2.2
5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3
5.1	3.8	1.5	0.3	5.6	2.5	3.9	1.1	6.0	2.2	5.0	1.5
5.4	3.4	1.7	0.2	5.9	3.2	4.8	1.8	6.9	3.2	5.7	2.3
5.1	3.7	1.5	0.4	6.1	2.8	4.0	1.3	5.6	2.8	4.9	2.0
4.6	3.6	1.0	0.2	6.3	2.5	4.9	1.5	7.7	2.8	6.7	2.0
5.1	3.3	1.7	0.5	6.1	2.8	4.7	1.2	6.3	2.7	4.9	1.8
4.8	3.4	1.9	0.2	6.4	2.9	4.3	1.3	6.7	3.3	5.7	2.1
5.0	3.0	1.6	0.2	6.6	3.0	4.4	1.4	7.2	3.2	6.0	1.8
5.0	3.4	1.6	0.4	6.8	2.8	4.8	1.4	6.2	2.8	4.8	1.8
5.2	3.5	1.5	0.2	6.7	3.0	5.0	1.7	6.1	3.0	4.9	1.8
5.2	3.4	1.4	0.2	6.0	2.9	4.5	1.5	6.4	2.8	5.6	2.1
4.7	3.2	1.6	0.2	5.7	2.6	3.5	1.0	7.2	3.0	5.8	1.6
4.8	3.1	1.6	0.2	5.5	2.4	3.8	1.1	7.4	2.8	6.1	1.9
5.4	3.4	1.5	0.4	5.5	2.4	3.7	1.0	7.9	3.8	6.4	2.0
5.2	4.1	1.5	0.1	5.8	2.7	3.9	1.2	6.4	2.8	5.6	2.2
5.5	4.2	1.4	0.2	6.0	2.7	5.1	1.6	6.3	2.8	5.1	1.5
4.9	3.1	1.5	0.2	5.4	3.0	4.5	1.5	6.1	2.6	5.6	1.4
5.0	3.2	1.2	0.2	6.0	3.4	4.5	1.6	7.7	3.0	6.1	2.3
5.5	3.5	1.3	0.2	6.7	3.1	4.7	1.5	6.3	3.4	5.6	2.4
4.9	3.6	1.4	0.1	6.3	2.3	4.4	1.3	6.4	3.1	5.5	1.8
4.4	3.0	1.3	0.2	5.6	3.0	4.1	1.3	6.0	3.0	4.8	1.8
5.1	3.4	1.5	0.2	5.5	2.5	4.0	1.3	6.9	3.1	5.4	2.1
5.0	3.5	1.3	0.3	5.5	2.6	4.4	1.2	6.7	3.1	5.6	2.4
4.5	2.3	1.3	0.3	6.1	3.0	4.6	1.4	6.9	3.1	5.1	2.3
4.4	3.2	1.3	0.2	5.8	2.6	4.0	1.2	5.8	2.7	5.1	1.9
5.0	3.5	1.6	0.6	5.0	2.3	3.3	1.0	6.8	3.2	5.9	2.3
5.1	3.8	1.9	0.4	5.6	2.7	4.2	1.3	6.7	3.3	5.7	2.5
4.8	3.0	1.4	0.3	5.7	3.0	4.2	1.2	6.7	3.0	5.2	2.3
5.1	3.8	1.6	0.2	5.7	2.9	4.2	1.3	6.3	2.5	5.0	1.9
4.6	3.2	1.4	0.2	6.2	2.9	4.3	1.3	6.5	3.0	5.2	2.0
5.3	3.7	1.5	0.2	5.1	2.5	3.0	1.1	6.2	3.4	5.4	2.3
5.0	3.3	1.4	0.2	5.7	2.8	4.1	1.3	5.9	3.0	5.1	1.8

Fonte: (FISHER, 1936)