UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO
DEPARTAMENTO DE COMPUTAÇÃO E MATEMÁTICA

PEDRO HENRIQUE D'ALMEIDA GIBERTI RISSATO

Reconhecimento de praxia não verbal em imagens da face humana utilizando Aprendizado de Máquina e Rede Neural

Ribeirão Preto-SP

2022

PEDRO HENRIQUE D'ALMEIDA GIBERTI RISSATO

Reconhecimento de praxia não verbal em imagens da face humana utilizando Aprendizado de Máquina e Rede Neural

Versão Corrigida

Versão original encontra-se na FFCLRP/USP.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) da Universidade de São Paulo (USP), como parte das exigências para a obtenção do título de Mestre em Ciências.

Área de Concentração: Computação Aplicada.

Orientadora: Profa. Dra. Alessandra Alaniz Macedo

Ribeirão Preto-SP

2022

Pedro Henrique D'Almeida Giberti Rissato

Reconhecimento de praxia não verbal em imagens da face humana utilizando Aprendizado de Máquina e Rede Neural. Ribeirão Preto-SP, 2022.

151p. : il.; 30 cm.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras
de Ribeirão Preto da USP, como parte das exigências para
a obtenção do título de Mestre em Ciências,
Área: Computação Aplicada.

Orientadora: Profa. Dra. Alessandra Alaniz Macedo

1. Visão Computacional. 2. Praxia Não-Verbal. 3. Redes Neurais.

Pedro Henrique D'Almeida Giberti Rissato

Reconhecimento de praxia não verbal em imagens da face humana utilizando Aprendizado de Máquina e Rede Neural

Modelo canônico de trabalho monográfico acadêmico em conformidade com as normas ABNT.

Trabalho aprovado. Ribeirão Preto-SP, 09 de Março de 2022:

Orientadora:
Profa. Dra. Alessandra Alaniz Macedo

Professora
Dra. Carolina Yukari Veludo Watanabe

Professora
Dra. Maria da Graça Campos Pimentel

Professor
Dr. Luiz Otávio Murta Júnior

Ribeirão Preto-SP
2022

Este trabalho é dedicado à minha avó, Celina Lopes D'Almeida, por cuidar e me tornar a pessoa que sou hoje. À minha esposa Bruna Fidêncio Rahal Ferraz, pela sua bondade, dedicação em tudo o que faz e por acreditar em mim, mesmo quando eu não acreditava. Às minhas filhas, Maria Clara e Maria Fernanda, que este trabalho sirva de motivação, para que vocês alcancem tudo que almejarem.

Agradecimentos

Agradeço...

Aos meus amigos pessoais, que há muito tempo fazem parte dessa história; por todas as palavras de alegria nos momentos felizes e pelas palavras de carinho nos momentos mais sombrios. Para alguém como eu, vocês são como minha família.

À minha querida amiga Fernanda Dellajustina e Breno Andrade por abrirem de bom coração sua casa e me receberem, por contribuírem ativamente com dados para essa pesquisa, e por serem pessoas maravilhosas! Este trabalho somente foi possível por vocês. Muito obrigado!

Aos meus colegas da pós graduação: Ronen Filho, Fernando Meloni, Tomaz Alexandre, Flávio Monteiro, Bianca Bortolai, Lina Garcés e Adriano Cantão, por todas as conversas, conhecimento, alegrias e tristezas que compartilhamos nesse período. Nossas vidas se cruzaram nessa caminhada e sempre que precisarem, um amigo estará por aqui.

À secretaria do Departamento de Computação e Matemática, em especial à Lúcia Akemi que, além de contribuir ativamente com dados para esta pesquisa, também cuidou dos meus assuntos acadêmicos com muito zelo e carinho. Por mais pessoas com você no funcionalismo público!

Aos Professores, Alinne Corrêa e Carlos Souza pela coleta de dados que embasaram diversos artigos e esta pesquisa. A qualidade e dedicação do trabalho de vocês fez toda a diferença.

Ao Prof. Renato Bulcão-Neto pela paciência, resiliência e companheirismo durante a execução deste e outros trabalhos. Obrigado por todos ensinamentos.

Aos membros do projeto "SofiaFala" por dedicarem parte das suas vidas à um projeto tão gratificante.

À todos meus professores pelos ensinamentos e pelo amor que possuem por essa linda profissão, em especial à minha Orientadora, Profa. Alessandra, pela sua dedicação à este e nossos outros trabalhos, por me ensinar à pensar de forma diferente e me introduzir no mundo da pesquisa acadêmica. Saiba que seus ensinamentos vão me acompanhar para sempre. Muito obrigado por sua paciência e companheirismo.

"Palavras são, na minha nada humilde opinião,
nossa inesgotável fonte de magia.
Capazes de causar grandes sofrimentos
e também de remediá-los."
(Harry Potter e as Relíquias da Morte, Parte 2, 2011)

Resumo

A capacidade de comunicar-se por meio da fala é essencial para qualquer ser humano. Contudo, pessoas com Transtorno de Fala (TF) decorridas de apraxia de fala na infância, desordem fonológica ou fonética necessitam de terapia fonoaudiológica. O profissional fonoaudiológico propõe uma série de exercícios para fortalecer os músculos orofaciais. Nesse contexto, os movimentos e sons não articulatórios como, por exemplo, sopro, estalo de língua ou beijo, exercitam e fortalecem boca, lábios, língua e bochechas que apoiam e sustentam a fala. Nesse sentido, o objetivo deste estudo consistiu em propor um método para o reconhecimento de beijo, estalo de língua e sopro na face humana utilizando pontos de marcação, denominados de *landmarks*. O método consiste em reconhecer o rosto humano, extrair a distância Euclidiana entre a análise combinatória de 20 *landmarks* da boca humana, para construir um vetor de distâncias. Esse vetor de distâncias foi utilizado para induzir modelos com os algoritmos de Árvore de Decisão, k-vizinhos mais próximos, Random Forest, Support Vector Machine e treinar uma rede neural do tipo Multilayer Perceptron. Por meio do método desenvolvido, o modelo induzido com Random Forest apresentou os melhores resultados e foi capaz de classificar entre as classes: (i) beijo e estalo; (ii) estalo e sopro e (iii) beijo e sopro, com uma acurácia de 93%, 93% e 65%, respectivamente. A separação entre os movimentos foi satisfatória e o modelo generalizado pode ser utilizado como apoio ao tratamento fonoaudiológico de pacientes com Transtornos de Fala.

Palavras-chave: Reconhecimento de padrões. *Landmarks*. Face humana. Visão Computacional.

Abstract

The ability to communicate through speech is essential for any human being. However, people with Speech Disorder (SD) due to childhood speech apraxia, phonological disorder or phonetics need speech therapy. The speech therapist proposes a series of exercises to strengthen the orofacial muscles. In this context, non-articulatory movements and sounds (such as blow, tongue snap, or kiss) strengthen the mouth, lips, tongue, and cheeks to support and sustain speech. In this sense, our goal was to propose a method to recognize kisses, tongue snaps and blows on the human face using landmarks. This method consists of the following steps: recognize the human face, extract the Euclidean distance between the combinatorial analysis of twenty landmarks from the human mouth, and create a vector of distances. This distance vector induces models with the Decision Tree, k-nearest neighbours, Random Forest, Support Vector Machine algorithms. It also trains a Multilayer Perceptron neural network. By using the proposed method, the model induced with Random Forest presented the best results and was able to classify between the classes: (i) kiss and snap; (ii) snap and blow and (iii) kiss and blow, with an accuracy of 93%, 93% and 65%, respectively. The distinction between the movements was satisfactory, and the generalized model can be used to support the speech therapy treatment of patients with Speech Disorders.

Keywords: Pattern recognition. Landmarks. Human face. Computer Vision.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

300-W	*300 Faces In-the-Wild Challenge*
ADAM	*Adaptive Momentum Estimation*
AFW	*The Annotated Faces in the Wild*
AM	Aprendizado de Máquina
AUC	*Area Under Curve*
AVC	Acidente Vascular Cerebral
CART	*Classification and Regression Tree*
CFAN	*Coarse-to-Fine Auto-Encoder Networks*
CK+	*The Extended Cohn-Kanade Dataset*
CK	*Cohn-Kanade Dataset*
CNN	*Convolutional Neural Networks*
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DNN	*Deep Neural Network*
DRMF	*Direct Robust Matrix Factorization Method*
ELM	*Extreme Learning Machine*
ERT	*Ensemble of Regression Trees*
FGnet	*Face and Gesture Recognition*
FN	Falso Negativo
FP	Falso Positivo
HOG	*Histogram of Oriented Gradients*
k-NN	*K-nearest neighbors*
LBP	*Local Binary Pattern*
LFPW	*Labeled Face Parts in the Wild*
LSTM	*Long Short-Term Memory*

MFCC	*Mel-Frequency Cepstrum Coefficients*
MLP	*Multilayer Perceptron*
MS	Mapeamento Sistemático
MTCNN	*Multi-task Cascaded Convolutional Neural Network*
PCA	*Principal Component Analysis*
PLN	Processamento de Linguagem Natural
QP	Questões de Pesquisa
ReLU	*Rectified Linear Unit*
RN	Rede Neural
ROIS	Regiões de Interesse
RVPNV	Reconhecimento Visual de Praxia Não Verbal
SD	Síndrome de Down
SDM	*Supervised Descent Method*
SIFT	*Scale-Invariant Feature Transform*
SMO	*Sequential Minimal Optimization*
SMOTE	*Synthetic Minority Over-sampling Technique*
SVM	*Support Vector Machine*
TEA	Transtorno Espectro Autista
TF	Transtornos de Fala
VC	Visão Computacional
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
WEKA	*Waikato Environment for Knowledge Analysis*

Sumário

1

Introdução

1.1 Contextualização

Pessoas com Transtorno Espectro Autista (TEA), com deficiência intelectual, Síndrome de Down (SD), sequelados de Acidente Vascular Cerebral (AVC), idosos, com disartrias e apraxias na infância podem desenvolver Transtornos de Fala (TF). A exemplo, pessoas com SD possuem uma modificação genética causada por uma cópia extra do cromossomo 21, conhecido como HSA21 (PATTERSON, 2009). Embora alterações fenotípicas, aquelas que impactam a aparência, possam diferir entre indivíduos, as alterações intelectuais estão presentes, variando de leves a moderadas (PATTERSON, 2009). Dessas alterações decorrem atraso no desenvolvimento dos sistemas motor, linguístico, cognitivo e funções adaptativas, quando comparado com crianças da mesma idade mental. É imperativo que tratamentos terapêuticos e medicamentos sejam ministrados com o intuito de promover o desenvolvimento de pessoas com SD, por exemplo. Os avanços para esse desenvolvimento mais expressivos são realizados na primeira infância da pessoa com a SD (STAGNI, 2015)[1].

Dentre as alterações físicas das pessoas com SD, destacam-se a hipotonia muscular[2] e dos ligamentos dos músculos da face (MUSTACCHI, 2009) como desencadeadores de uma série de déficits fonoaudiológicos como sintático, pragmático e semântico da linguagem, além de desordens auditivas que também contribuem para o atraso no desenvolvimento da linguagem (BARBOSA, 2015).

No paciente com SD, dada a acentuada perda do tônus muscular, os músculos orofaciais devem ser fortalecidos com práticas semelhantes às utilizadas na fisioterapia corporal. Neste contexto, o profissional fonoaudiológico define exercícios repetitivos para

[1] Esta pesquisa deriva do trabalho realizado pelo grupo de pesquisa denominado "SofiaFala", apoiado pelo CNPq por meio do processo n. 442533/2016-0, que objetiva o desenvolvimento de aplicativos para apoiar o desenvolvimento da fala em crianças com SD. Sítio do projeto: https://dcm.ffclrp.usp.br/sofiafala/sobre.php.

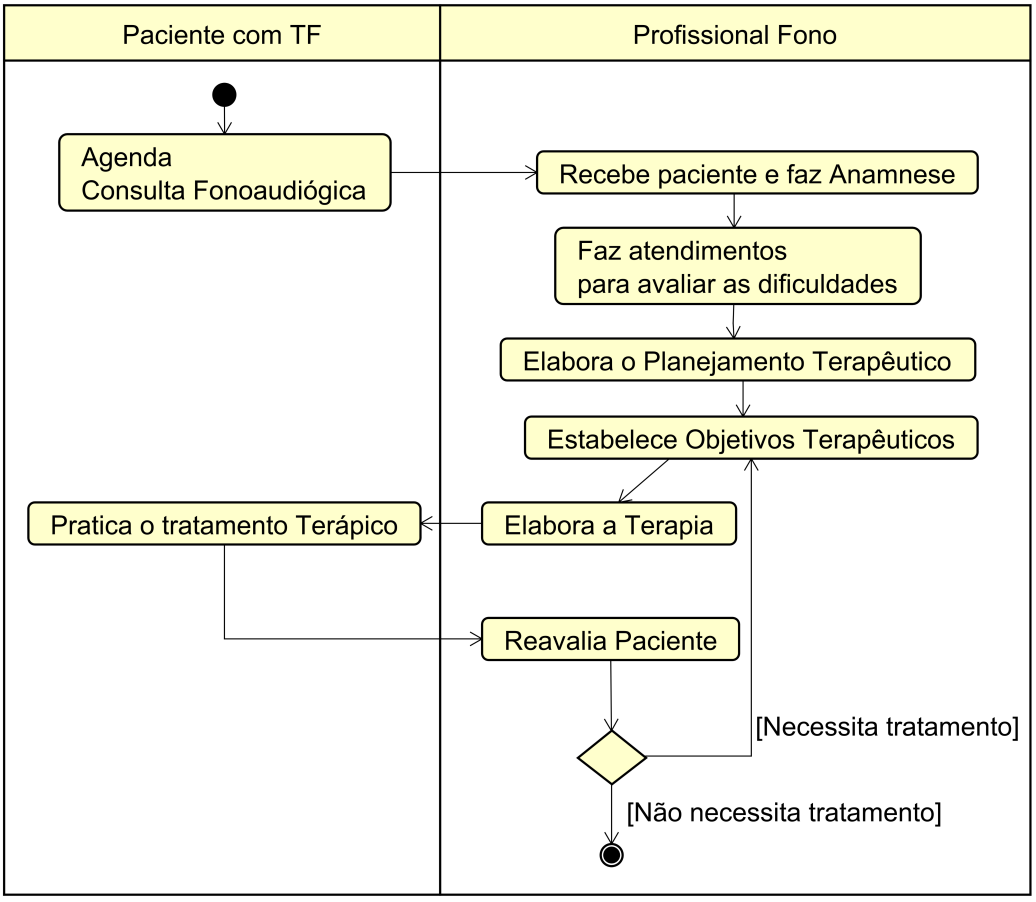
[2] Trata-se da diminuição da força/tônus muscular, o que pode causar fraqueza ou flacidez nos músculos que auxiliam na fala.

promover a rigidez muscular necessária para a produção de palavras que culminem na oratória desejada, respeitadas as limitações verbais do próprio indivíduo em crescimento (GIACCHINI, 2013). Esses exercícios de fortalecimento de estruturas orofaciais e do sistema estomatognático[3] são realizados por meio de praxias não verbais, mediante a produção de sons denominados não articulatórios, os quais não produzem sílabas ou palavras, mas são base de desenvolvimento muscular da linguagem falada. Movimentos como beijo, estalo de língua e sopro, são capazes de exercitar os músculos orofaciais (BEARZOTTI, 2007).

Na Figura 1, pode-se observar o procedimento adotado pelo profissional fonoaudiológico para estabelecer o tratamento adequado específico para pacientes com Transtornos de Fala. De acordo com o diagrama de atividades da Figura 1, o paciente com TF entra em contato com o profissional fonoaudiológico. Este profissional realiza, no primeiro atendimento, a anamnese, cujo objetivo principal é fazer o diagnóstico do paciente. Após o primeiro encontro, são agendados alguns atendimentos, quando é avaliado o modelo de comunicação do paciente, isto é, como ele(a) compreende a comunicação e a prática das praxias não verbais (informação verbal)[4].

[3] Sistema de estrutura bucal com a participação da mandíbula.
[4] Protocolo de anamnese fornecido pela fonoaudióloga Ma. Bianca Bortolai Sicchieri.

Figura 1 – Diagrama dos procedimentos para a realização do processo de tratamento por profissional fonoaudiológico



Fonte: Autoria própria

Posteriormente, o profissional realiza o planejamento terapêutico, com o estabelecimento dos objetivos que se pretendem alcançar, bem como a elaboração da terapia. Esta terapia é aplicada ao paciente com TF que realiza os movimentos indicados pelo profissional por períodos de tempo. Constantemente, o profissional reavalia o paciente para verificar se este está de alta ou necessita continuar o tratamento

As sessões fonoaudiológicas advindas da terapia são realizadas em parte no consultório e em grande maioria fora desse ambiente, na residência do paciente, por exemplo, sem a constante supervisão do profissional. Essa ausência de validação especialista na execução do treino fora do ambiente do consultório e a influência de distrações do ambiente podem causar distúrbios na prática assertiva dos movimentos necessários ao desenvolvimento orofacial. Com as tecnologias computacionais assistivas, existem meios de coletar dados objetivamente de modo a prover uma forma de auxiliar o paciente a sempre performar o movimento de interesse na forma prescrita pelo profissional fonoaudiólogo.

A hipótese da pesquisa apresentada nesta dissertação é que para validar a realização de movimentos não articulatórios realizados por pacientes com TF possa se utilizar

métodos de Visão Computacional (VC)[5] de modo a prover a detecção de padrões, por meio da análise das imagens capturados dos movimentos. A ideia é aplicar a indução de modelos capazes de classificar novos exemplos entre os padrões de (i) beijo e estalo, (ii) estalo e sopro e (iii) beijo e sopro que destinam-se a exercitar os músculos orofaciais responsáveis pela fala.

No tangente à análise de padrões na Visão Computacional, a detecção de face humana enquadra-se nas categorias de detecção ou reconhecimento de objetos. Após a detecção/reconhecimento da face, segundo o trabalho em Kumar, Kaur e Kumar (2018), existem duas grandes abordagens para extrair dados de imagens da face, (i) baseada em atributos e (ii) baseada na imagem. A análise baseada na imagem, consiste em empregar redes neurais, subespaço linear ou abordagem estatística para detectar partes da face como olhos, boca, nariz, etc, e inclusive a própria face.

Por outro lado, na análise baseada em atributos, o foco é extrair atributos da imagem e compará-los com atributos conhecidos do rosto humano. Dentre as técnicas disponíveis, encontram-se *Active Shape Model*, análise de baixo nível e análise de atributo. Na técnica baseada em *Active Shape Model*, marcadores são aplicados na face reconhecida para extração de dados. Esses marcadores são denominados de *landmarks*.

A extração de dados por meio dos *landmarks* exerce papel de destaque no campo da Visão Computacional como processo intermediário para uma vasta gama de processos subsequentes de análises desde reconhecimento biométrico até análise de estados mentais (OUANAN, 2016). Conforme exposto em Cui, Huang e Liu (2018), os *landmarks* são invariantes traslação, rotação e escala, o que reforça sua aplicabilidade para domínios que exijam um refinamento mais apurado na detecção de padrões com nuances, tais como beijo, estalo de língua e sopro.

1.2 Motivação

O "SofiaFala" objetiva o desenvolvimento de um sistema inteligente e interativo como tecnologia de apoio ao treinamento de pessoas com TF. No citado projeto, estão sendo desenvolvidos dois métodos de análise e apoio à fala: um envolvendo áudio e outro vídeo.

Em Souza, Souza, Watanabe, Mandrá e Macedo (2019) foi desenvolvido um método utilizando coeficientes extraídos de exemplos de áudios, utilizando a técnica *Mel-Frequency Cepstrum Coefficients* (MFCC) dentre uma janela temporal de 20ms em conjunto com a transformada de LaPlace em escala logaritmica aplicada ao quadrado dos coeficientes adotados para induzir um modelo baseado em Support Vector Machine (SVM)

[5] É a área da computação que estuda a capacidade de máquinas analisarem imagens e dados espaciais destas.

obtendo uma acurácia final de 75%, 55% e 38% em cenários como controle, adicionando ruídos como sons de TV e, por fim, sons de chuva, respectivamente. No mesmo trabalho, os autores realizam a avaliação visual dos quadros dos vídeos extraídos que originaram os áudios, apresentando-os para indução de uma Rede Neural do tipo Long Short-Term Memory (LSTM), denominada Inception V3, a qual atingiu a acurácia de 51%, para reconhecimento do movimento de sopro, 66% para o movimento de estalo de língua e 81% para o movimento de beijo.

No trabalho Meloni, Sicchieri, Mandrá, Bulcão-Neto e Macedo (2021), os autores apresentam um método de análise e reconhecimento de sons não articulatórios como beijo, estalo de língua e sopro, utilizando a técnica MFCC, também aplicadas em um espaço temporal com recortes em duas faixas de tempos pré-estabelecidas para reconhecer subpadrões dentre os padrões objetivados, afim de produzir uma classificação mais acurada. Ao final, a acurácia balanceada foi de 95%, 92% e 82% para reconhecimento de beijo, estalo de língua e sopro, respectivamente.

Conforme disposto nos trabalhos Souza, Souza, Watanabe, Mandrá e Macedo (2019), Meloni, Sicchieri, Mandrá, Bulcão-Neto e Macedo (2021), assim como o método apresentado nesta dissertação, o desenvolvimento de um modelo híbrido envolvendo áudio e vídeo simultaneamente pode ser capaz de aprimorar a predição de novos exemplos.

1.3 Objetivo

O objetivo desta pesquisa é propor um método para reconhecimento de praxias não verbais como beijo, estalo de língua e sopro na face humana, por meio da análise de distâncias Euclidianas de vinte marcadores (*landmarks*) aplicados na boca.

Para alcançar o objetivo principal, os seguintes objetivos secundários foram realizados:

- Criação de um conjunto de imagens dos movimentos de interesse a partir de coleta e avaliação por um profissional fonoaudiológico, sendo tais imagens, assim, representativas do movimento ideal a ser executado pelo paciente com TF.
- Mapeamento Sistemático (MS) com o intuito de encontrar as técnicas, algoritmos, aplicações e conjunto de dados utilizados na aplicação de análise envolvendo *landmarks* e boca humana.
- Definição de um método para induzir um modelo a reconhecer os padrões de beijo, estalo de língua e sopro.
- Aplicação do modelo induzido em conjunto de dados de teste.

- Aplicação das métricas para avaliação dos resultados apresentados.

1.4 Resultado e Limitações

Como resultado, foi possível realizar a geração de modelos induzidos capazes de, sob a ótica das métricas avaliadas, diferenciar com um desempenho considerável, entre os movimentos de beijo, estalo de língua e sopro. Para o obtenção desses modelos, diversos cenários foram construídos com o intuito de demonstrar a evolução desta pesquisa desde a concepção da ideia com o primeiro conjunto de dados até os modelos finais obtidos.

Infelizmente, não foi possível desenvolver um modelo único capaz de distinguir os três movimentos indistintamente, com uma acurácia aceitável (igual ou maior a 90%), em especial entre os movimentos de beijo e sopro. Uma possível explicação para que os modelos não sejam capazes de separar entre aqueles movimentos concomitantemente, por possuírem distribuição e sobreposição de distâncias similares na amplitude de tais movimentos. Para investigar essas suposições, foram propostos trabalhos de continuação do trabalho apresentação nessa dissertação.

1.5 Organização do Documento

O documento está organizado da seguinte forma: o Capítulo 2 apresenta os trabalhos relacionados a esta pesquisa; o Capítulo 3 apresenta a Fundamentação Teórica; o Capítulo 4 apresenta os processos para criação dos conjuntos de dados; o Capítulo 5 o delineamento metodológico para a criação do Método de Reconhecimento Visual de Praxia Não Verbal; no Capítulo 6 são apresentados os resultados obtidos e, por fim, no Capítulo 7 é apresentada a conclusão deste trabalho.

2

Trabalhos Relacionados

Até o presente momento, a literatura correlata apresenta escassos trabalhos relacionados ao reconhecimento de padrões envolvendo beijo, estalo de língua ou sopro na face, independente da técnica utilizada.

Dentre os estudos encontrados, em Garrido, Zollhöfer, Wu, Bradley, Pérez, Beeler e Theobalt (2016) é apresentada uma abordagem para reconstrução 3D corretiva de lábios para vídeo monocular para animação facial, incluindo a reconstrução do movimento de beijo. Os autores utilizam uma *radial basis function network* em um conjunto de dados construído para tal finalidade e a compara com a verdade real dos lábios em 3D dos indivíduos modelados. No trabalho, os autores utilizam 66 *landmarks* na face para mapear a face e o contorno dos lábios. Na comparação com um regressor de afinidade, o método apresentado obteve uma taxa de erro de 13% (desvio padrão de 0.04) contra 14% (desvio padrão de 0.05) do regressor de afinidade.

Em Zhang (2016) foi desenvolvido um dispositivo de simulação do beijo humano através da rede denominado de *Kissinger*. Na Figura 2 é possível visualizar o dispositivo que conecta em aparelhos iPhone e permite que os usuários se beijem remotamente. Existem sensores hápticos que recebem os dados do dispositivo criado e transmite a sensação ao outro, e também um módulo que libera feromônios durante sua utilização. Sensores de força medem a força dos lábios no dispositivo, que é transmitida bidirecionalmente via internet para ambos os dispositivos, controlados por um aplicativo próprio.

No campo do Processamento da Linguagem Natural em conjunto com a Interação Humano-Computador (HCI), o trabalho em Igarashi e Hughes (2001) analisa um tipo de estalo de língua, mais curto e rápido, como meio de controle não verbal direto para aplicações interativas. O trabalho propõe a extração de uma janela de áudio baseando-se na *Fast Fourier Transformation* no tempo de 12ms. A técnica, que utiliza somente o som, detecta picos no áudio, como por exemplo o som não verbal produzido pelo estalo rápido da língua e o modelo detecta os picos na janela temporal delimitada para produzir o comando.

Figura 2 – Dispositivo denominado *Kissinger* que transmite dados entre os dispositivos correlatos para simular um beijo humano

Fonte: Zhang (2016)

Tratando-se do Reconhecimento da Fala, no trabalho em Reyes, Zhang, Ghosh, Shah, Wu, Parnami, Bercik, Starner, Abowd e Edwards (2016) o sopro é analisado para controlar um *smartwatch* por meio do dispositivo criado denominado *FluetCase*. O dispositivo altera o fluxo de ar de um sopro de forma que gere tons diferentes para determinadas formas de sopro. Esses tons são captados pelo microfone do relógio e um aplicativo desenvolvido no trabalho, os converte em ações a serem executadas no *smartwatch*. Uma validação 10-Fold Cross foi realizada com um conjunto de dados de 2179 amostras e o dispositivo obteve uma acurácia final geral de 79.7% (desvio padrão de 9.7%).

Apenas o trabalho em Souza, Souza, Watanabe, Mandrá e Macedo (2019) fez análise com os movimentos de estalo de língua, beijo e sopro envolvendo análise de imagens. Portanto, um estudo mais aprofundado e amplo, denominado Mapeamento Sistemático (MS) foi executado para localizar estudos relacionados que envolvessem reconhecimento de padrões na face humana, que envolvessem a boca e *landmarks*.

Um Mapeamento Sistemático tem por objetivo reunir todos os estudos primários que envolvem um determinado tema, com o intuito de saber o que é pesquisado sobre o tópico escolhido e suas variações, bem como tentar identificar lacunas que possam ser exploradas (NAKAGAWA, 2017). Um MS pode ser utilizado para o levantamento de

trabalhos relacionados a uma determinada proposta de pesquisa.

Seguindo as diretrizes expostas por Nakagawa, Scannavino, Fabbri e Ferrari (2017), o trabalho em Rissato, Bulcao-Neto e Macedo (2021) identifica os estudos primários que tratam sobre a técnica de *landmark* na boca humana para o reconhecimento de padrões. Com esse intuito, foram realizadas a buscas em seis fontes de pesquisa [1], utilizando-se da *string* de busca: "*facial AND mouth AND landmark AND detection*". O resultado dessas buscas retornou um total de 344 trabalhos acadêmicos entre os anos de 2015 e 2021, que após critérios de inclusão e exclusão, foram reduzidos a 115 artigos científicos.

Os seguintes critérios de **inclusão** foram definidos:

1. O estudo apresenta o reconhecimento de padrões de imagens da face humana comparando uma ou mais técnicas de *landmark* com outra(s) técnica(s).
2. O estudo apresenta o reconhecimento de padrões de imagens da face humana propondo uma nova técnica de *landmark*.
3. O estudo relata o reconhecimento de padrões de imagens da face humana envolvendo a utilização da técnica de *landmark*.

Os seguintes critérios de **exclusão** foram definidos:

1. Não é aplicada/mencionada a aplicação da técnica de *landmark* envolvendo a boca.
2. Não é um estudo primário.
3. O estudo é uma versão mais antiga de outro estudo já considerado.
4. O estudo foi publicado apenas como resumo.
5. O estudo for mais antigo que 07 (sete) anos.
6. O estudo não aborda o uso da técnica de *landmark* no reconhecimento de padrões em imagens da face.
7. O estudo não possui um resumo.
8. O texto completo do estudo não está disponível na Web ou no Portal de Periódicos da CAPES.
9. O texto do estudo não está escrito na língua inglesa.
10. Ser capítulo de livro.

[1] As bases utilizadas no Mapeameto Sistemático (MS) citado foram: *ACM Digital Library*, *Engineering Village*, *IEEE Digital Library*, *ScienceDirect*, *Scopus* e *Web of Science*.

Conforme descrito por Nakagawa, Scannavino, Fabbri e Ferrari (2017), foram definidas as seguintes Questões de Pesquisa (QP), que a MS tem a finalidade de responder:

QP 1: Quais são os estudos primários que aplicam a técnica de *landmark* com o intuito de reconhecer padrões na face humana?

QP 2: Considerando os estudos selecionados, quais são as técnicas mais utilizadas para extrair ou selecionar atributos?

QP 3: Dos estudos selecionados, quais são os algoritmos que são utilizados para reconhecer os padrões almejados?

QP 4: Quais conjuntos de dados foram utilizados nos estudos?

QP 5: Quais eram os problemas (que tipo de padrão) que os estudos levantados estavam tentando reconhecer?

Não obstante a importância de todas as questões apresentadas, se destaca a QP 3, pois aborda todos algoritmos que apareceram nos trabalhos relacionados acerca do tópico pesquisado. Os algoritmos mais utilizados e a quantidade de usos nos trabalhos do MS são apresentados na Tabela 1.

Tabela 1 – Algoritmos mais utilizados encontrados após a realização de Mapeamento Sistemático contendo uma *string* de busca volta para trabalhos que envolvessem *landmarks* na face humana

Algoritmo	Quantidade
Support Vector Machine (SVM)	25
Convolutional Neural Networks (CNN)	17
Viola-Jones	10
K-nearest neighbors (k-NN)	7
Supervised Descent Method (SDM)	7
AdaBoost	6
Decision Tree	5
Random Forest	5
ResNet	5
Deep Neural Network (DNN)	4
Ensemble of Regression Trees (ERT)	4
Long Short-Term Memory (LSTM)	4
Multi-task Cascaded Convolutional Neural Network (MTCNN)	4
Coarse-to-Fine Auto-Encoder Networks (CFAN)	3
Direct Robust Matrix Factorization Method (DRMF)	3
Multilayer Perceptron (MLP)	3
Naïve Bayes	3
SegNet	3

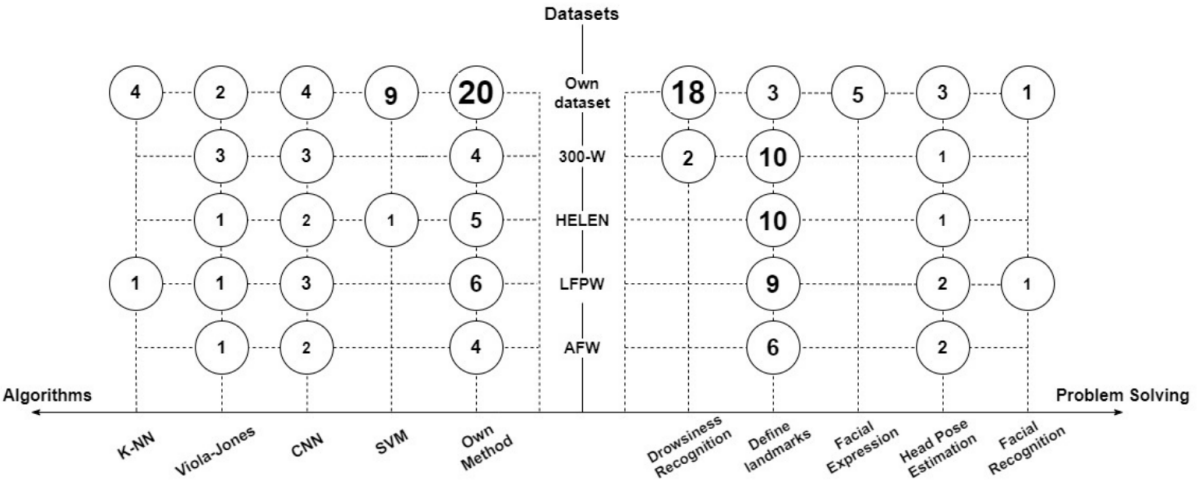
Fonte: Autoria própria

Os algoritmos da Tabela 1, em conjunto com 53 trabalhos que propuseram algoritmos próprios para resolução das questões de pesquisa inerentes, representam 62% do total (146) de algoritmos utilizados. Como visualizado, com esse tipo de resultado retornado pelo MS, é possível destacar alguns algoritmos para classificar as praxias não verbais de interesse.

A Figura 3 demonstra a correlação entre cinco os domínios, conjuntos de dados e algorítimos mais utilizados obtidos através do MS (RISSATO, 2021).

Os autores de Rissato, Bulcao-Neto e Macedo (2021), concluem, em suma, que para no tocante ao domínio de reconhecimento de sonolência ao voltante, dos 29 estudos na área 18 construíram seu próprio conjunto de dados, considerando a falta de especificidade de dados para o problema. Há uma distribuição homogênea entre os algorítimos de Viola-Jones e redes neurais convolucionais, onde aquele foi o precursor do reconhecimento da face humana em imagens e, essas estão se tornando mais comuns dado o aumento do poder computacional. Por fim, dentre 20 estudos na área de reconhecimento de expressões faciais, foram utilizados 17 conjuntos de dados distintos, demonstrando que apesar da relevância do campo, não há consenso sobre os conjuntos para criar modelos generalizados para a área.

Figura 3 – Gráfico de bolhas contendo a agregação das respostas obtidas das questões de pesquisa QP3, QP4 e QP5. Do lado esquerdo são correlacionados os eixos de conjuntos de dados e algorítimos mais utilizados, ao lado direito do eixo central, são comparados os conjuntos de dados com os domínios mais apontados



Fonte: Rissato, Bulcao-Neto e Macedo (2021)

É válido destacar que o MS observado não obteve nenhum trabalho que explorou os padrões de beijo, estalo de língua ou sopro na face humana, separados ou concomitantemente. O próprio MS buscou trabalhos que envolvessem somente a boca humana e a aplicação da técnica de *landmarks*, logo, todos os trabalhos encontrados são afetos ao domínio especificado, entretanto, é válido destacar trabalhos que mais se aproximam ao escopo dessa dissertação.

No trabalho García, Álvarez e Orozco (2017) foi apresentado um método de reconhecimento de emoções utilizando a distância Euclidiana entre os *landmarks* das regiões dos olhos e da boca. Foi utilizada a técnica de distribuição do erro relativo como método de validação da acurácia. A acurácia média final e desvio padrão foram respectivamente 94,53% $\pm$ 2,47%.

Um modelo baseado em k-NN foi proposto em Anas, Ramadijanti e Basuki (2018), para reconhecer se uma pessoa gostou ou não de um item de moda. Foi utilizada a biblioteca dlib para reconhecimento da face humana e extração de 42 landmarks dos olhos, sobrancelhas e boca. Foram analisadas 63 imagens contendo 32 expressões do gostar de item e 31 de não gostar. O melhor modelo apresentado possui um erro médio de 15,83% (acurácia de 84,17%).

Conforme apresentado em Cui, Huang e Liu (2018), foi desenvolvido um método para reconhecer o sorriso na face humana, utilizando a distância Euclidiana como atributo para treinar um algoritmo denominado *Extreme Learning Machine* (ELM). Foi obtida a

acurácia de 93,40% em um dos conjuntos de testes.

Em Salmam, Madani e Kissi (2016) foi proposto um método de reconhecimento de expressões faciais para o treinamento de uma árvore de decisão[2] chamada *Classification and Regression Tree* (CART) que utiliza como atributos as medidas das distâncias Euclidiana, *Manhattan* e Minkowski.

Em Beh e Goh (2019) foi apresentado um método para detecção de micro-expressões faciais. A biblioteca dlib foi utilizada para reconhecer a face humana, extrair 24 landmarks dentre bocas, olhos, sobrancelhas e nariz para calcular 12 distâncias Euclidianas entre tais marcações. Razões foram calculadas entre as distâncias dos olhos e boca para reduzir o erro natural causado pela perspectiva do posicionamento da câmera. Alterações nos frames subsequentes são comparadas com o primeiro frame e considerando um *threshold* e sensibilidade inferidas, as micro-expressões são detectadas com uma acurácia média de 64,77%. Essa acurácia supera a detecção por humanos e também de outros modelos correlatos.

2.1 Considerações Finais

Nesse capítulo foram apresentados achados que mais se aproximam da temática desse trabalho, considerando a escassez de trabalhos na literatura. No MS foram estudados os trabalhos que de alguma forma aplicaram *landmarks* no reconhecimento de padrões que envolvessem a boca humana. Destacam-se os trabalhos em que padrões na boca foram mais minuciosamente explorados ou que possuíssem uma correlação com a temática dessa dissertação. A exemplo do domínio de reconhecimento de sonolência ao volante, apresentado no MS, este trabalho também propôs a extensão do conjunto de dados criado em Souza, Souza, Watanabe, Mandrá e Macedo (2019), bem como a criação do novo conjunto de testes explorada no Capítulo 4 considerando a especificidade do domínio explorado.

[2] É um método de aprendizado que baseia-se em decisões considerando os valores dos atributos, e, o resultado final é representado em forma de árvore, onde parte-se do nó raiz para os nós folhas, considerando as decisões aprendidas. Pode ser representada também como um conjunto de *if-else* (MITCHELL, 1997).

3

Fundamentos Teóricos

Este capítulo apresenta conceitos e tecnologias utilizados no desenvolvimento deste trabalho. São apresentados conjunto de dados comumente utilizados para detecção de padrões em imagens. Temas sobre processamento e extração de atributos em imagens são abordados como etapas preliminares para detectar padrões em imagens.

3.1 *Landmarks* na Face Humana

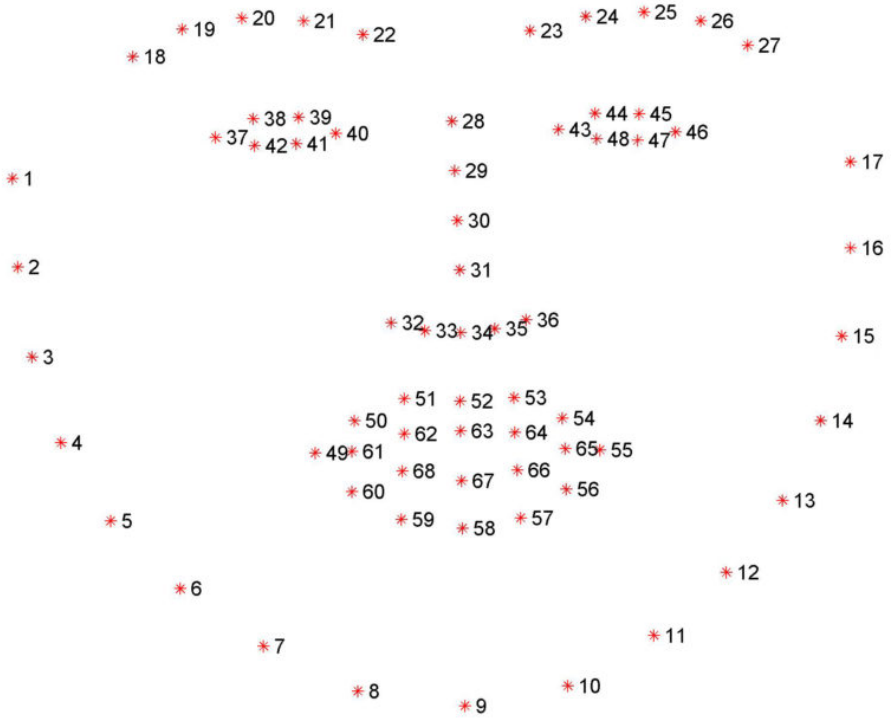
Os *landmarks* são pontos de marcações de um modo geral que objetivam destacar uma regiões de interesse. Na medicina, os *landmarks* foram aplicados na face humana para realizar medidas de distâncias entre essas marcações com o intuito de definir a estrutura do crânio humano (FARKAS; DEUTSCH, 1996).

Os *landmarks* podem ser reproduzidos digitalmente por meio da imagem da face humana, conforme ilustrado na Figura 4. Assim, os dados desses *landmarks* podem ser extraídos para análises. Os métodos de definições dos marcadores na face humana são diversificados. Em Gondhi, Kour, Effendi e Kaushik (2017) foi utilizado o algoritmo de detecção de cantos, bem como medidas médias da face humana. Outros trabalhos, como Yu, Yang, Huang e Metaxas (2013), adotam técnicas mais consolidadas como Histogram of Oriented Gradients (HOG) [1] para compor os *landmarks*, inclusive quando existe parte da face encoberta. Outros como Yang, Shu e Zhou (2016) utilizam Redes Neurais com um massivo grupo de imagens para treinamento.

No campo da Visão Computacional, essas marcações são utilizadas para os mais variados objetivos, como o reconhecimento do formato do rosto humano (VIOLA; JONES, 2001), o reconhecimento de expressões faciais (SALMAM, 2016) e o reconhecimento de emoções (SHIVASHANKAR; HIREMATH, 2017).

[1] Método específico para a extração de atributos descritores das imagens.

Figura 4 – Representação dos marcadores, denominados de *landmarks*, na face humana utilizados como base para a extração de dados



Fonte: (SAGONAS, 2013)

3.2 Coleções de Imagens

Datasets, também conhecidos como Conjunto de Dados, Conjunto de Exemplos ou *Corpus* de imagens, são exemplos de dados compostos por atributos e classes. Atributos são elementos que representam o exemplo amostrado, com o objetivo de descrevê-lo em sua essência. A classe é descrita como o objetivo a ser atingido ao se predizer os dados contidos nos atributos do exemplo (REZENDE, 2003). Os conjuntos de dados são utilizados para realizar a indução de um aprendizado supervisionado ou semi-supervisionado.

No campo da Visão Computacional (VC), os dados que compõe o conjunto são extraídos de imagens. De fato, as imagens que constituem o conjunto de dados são extremamente importantes para a detecção dos padrões desejados. Conforme o próprio campo da VC evolui, conjuntos de dados específicos para determinados tipos de padrões são criados para prover à comunidade científica coleções de referências. Por exemplo, Phillips, Wechsler, Huang e Rauss (1998) apresenta critérios para criação de conjuntos de dados que envolvam a análise de reconhecimento facial.

Para que se possa realizar estudos na área da VC, o Mapeamento Sistemático (MS), descrito na Seção 2, buscou elencar os conjuntos de dados mais utilizados quando a temática envolve padrões na face humana utilizando *landmarks*. A Tabela 2 quantifica

imagens e *landmarks* dos dez conjuntos de dados resultantes do MS. Esses conjuntos representam os dados extraídos de imagens da face humana mais utilizados entre os anos de 2015 a 2021. A validação do modelo generalizado induzido pode ser realizada com um ou mais desses conjuntos de dados mapeados.

Tabela 2 – Levantamento dos conjuntos de dados por meio do Mapeamento Sistemático que encontrou trabalhos que envolvam a boca humana e *landmarks*, bem como as quantidades de imagens, quantidades de *landmarks* e datas de criação desses conjuntos

Conjunto de Dados	Imagens	*Landmarks*	Criação
Labeled Face Parts in the Wild (LFPW)	1432	29	2011
HELEN dataset	2330	194	2012
300 Faces In-the-Wild Challenge (300-W)	600	68	2013
CMU Multi-PIE Face Database	750000	68	2008
The Annotated Faces in the Wild (AFW)	205	6	2012
The Extended Cohn-Kanade Dataset (CK+)	593[1]	68	2010
AR Face Database	4000	——	1998
Bosphorus 3D Face Database	4666	24	2008
Cohn-Kanade Dataset (CK)	1917[2]	——	2000
Face and Gesture Recognition (FGnet)	1002	——	2004

[1]Sequência de imagens que podem conter de 10 a 60 quadros cada
[2]Sequência de imagens que podem conter de 9 a 60 quadros cada
Fonte: Autoria própria

Em resumo, os conjuntos de dados da Tabela 2 consistem de imagens estáticas[2], compostos por:

- Labeled Face Parts in the Wild (BELHUMEUR, 2013): imagens apenas de rostos humanos adquiridas de sítios com simples consulta; *landmarks* adicionados manualmente.

- HELEN (LE, 2012): imagens de resolução maior de 500 pixels de largura do rosto humano em diversas condições de luz, poses e expressões faciais criado através de site com simples consulta; *landmarks* adicionados manualmente.

- 300 Faces In-the-Wild Challenge (SAGONAS, 2016): 600 imagens de rostos humanos ao "ar livre" e em locais fechados contendo de um a sete rostos em diversas variações de posições, expressões e condições de luz, obtidas de sítio de simples consulta; *landmarks* dispostos automaticamente por meio de ferramenta desenvolvida pelos autores.

[2] Com exceção dos conjuntos The Extended Cohn-Kanade Dataset (CK+) e Cohn-Kanade Dataset (CK) que apesar de serem imagens estáticas, foram extraídas de vídeos.

- CMU Multi-PIE Face Database[3]: imagens de faces humanas de 337 pessoas, adquiridas em laboratório através de 15 câmeras e 19 pontos de iluminação em diferentes posições e diversas expressões faciais; não há informação sobre como os *landmarks* foram adicionados.

- The Annotated Faces in the Wild (ZHU; RAMANAN, 2012): imagens obtidas randomicamente de sítio de simples consulta contendo diversas variações de fundos, posições do rosto, expressões e aparência; não há informação sobre como os *landmarks* foram adicionados.

- Cohn-Kanade Dataset (KANADE, 2000): imagens de faces humanas de 210 pessoas, gravadas a partir de duas câmeras (uma frontal e outra em um ângulo de 30 graus) com 23 expressões faciais diferentes.

- AR Face Database[4]: imagens obtidas em laboratório de 126 pessoas (63 homens e 53 mulheres), sem restrição quanto ao uso de maquiagem, estilo do cabelo, roupas; a forma de aquisição das imagens foi controlada através de condições de iluminação, distância do rosto humano e foco da câmera.

- Bosphorus 3D Face Database[5]: imagens obtidas em laboratório de 105 pessoas (um terço eram atores profissionais), onde cada indivíduo apresentava até 35 expressões faciais, mapeadas através de um sensor 3D baseado em luz e posicionado a um metro e meio do voluntário; *landmarks* foram manualmente adicionados nas imagens 2D geradas e posteriormente recalculados para as posições nas imagens 3D.

- The Extended Cohn-Kanade Dataset (LUCEY, 2010): similar ao dataset (vi), exceto pelo número de sequência das imagens obtidas aumentado em 22% e aumento do número de participantes em 27%, sendo a sequência de imagens totalmente codificada com o método FACS5, bem como os rótulos das expressões faciais realizadas.

- Face and Gesture Recognition (PANIS, 2016): imagens de 82 indivíduos, sendo sua característica principal a variação de idade (recém-nascidos até pessoas com 69 anos).

Os conjuntos de dados Labeled Face Parts in the Wild e HELEN foram desenvolvidos com o objetivo de pesquisar a detecção de partes na face humana (olhos, boca, nariz, etc.). Já no conjunto 300 Faces In-the-Wild Challenge o foco foi um desafio proposto para desenvolver métodos automatizados de inserir *landmarks* em imagens contendo rosto humano. Em relação a CMU Multi-PIE Face Database não há informação sobre a motivação da criação. Em The Annotated Faces in the Wild e AR Face Database, o

[3] Disponível em http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html.
[4] Disponível em http://www2.ece.ohio-state.edu/ aleix/ARdatabase.html.
[5] Disponível em http://bosphorus.ee.boun.edu.tr/Home.aspx.

objetivo era a detecção do rosto humano. Nos conjuntos Cohn-Kanade Dataset e The Extended Cohn-Kanade Dataset, o propósito é o reconhecimento de expressões faciais. Por fim, em Face and Gesture Recognition, os autores almejavam produzir um conjunto de dados que pudesse ser utilizado para detectar o envelhecimento[6].

3.3 Processamento e Extração de Atributos em Imagens

Processar uma imagem significa analisá-la digitalmente por meio de um computador. Uma imagem digital consiste de uma função bi-dimensional, $f(x, y)$, onde x e y são coordenadas em um plano espacial e a amplitude de f sobre os pares de coordenadas (x, y) determina a intensidade da imagem naquele ponto. Quando os valores de x, y e a amplitude de f são todos finitos (discretos), a imagem digital está definida (GONZALEZ, 2009). Processamento, seleção e extração de atributos são usados em imagem para selecionar os atributos que melhor contribuirão para a detecção de um padrão desejado.

Este trabalho, considerando os estudos realizados, abordou os processamentos *mid-level* e *high-level*, segmentando e contextualizando as áreas de interesse por meio de um modelo capaz de distinguir entre os movimentos (classes) de praxias não verbais de interesse.

3.3.1 Processamento

Apesar da falta de consenso nos limites das áreas de Processamento de Imagem, Análise de Imagem e Visão Computacional, Gonzalez (2009) define que existem três tipos de processos envolvendo o processamento digital de uma imagem. O primeiro deles, denominado *low-level*, consiste em operações basilares de processamento, como melhoria no contraste, redução de ruído ou recortes da imagem. Este processo é caracterizado por ter como entrada uma imagem e a saída, depois do processamento, também uma imagem (GONZALEZ, 2009).

Já no processamento *mid-level*, o foco é a segmentação, que é um processo no qual a imagem original é dividida ou segmentada em regiões ou objetos de interesse, para que seja possível o processamento pelo computador e o reconhecimento dos objetos segmentados. Define-se que este processamento recebe como entrada uma imagem, mas

[6] Essa detecção de envelhecimento consiste em três linhas: (i) estimar a idade de uma pessoa baseada em informações da face. (ii) reconhecer uma pessoa, independente das ações causadas pelo envelhecimento e (iii) estimar o envelhecimento futuro da aparência de uma pessoa.

a saída são atributos extraídos dessa imagem, que podem ser bordas, contornos, dentre outros (GONZALEZ, 2009).

Por último, existe o processamento denominado *high-level*, que consiste em tentar dar sentido aos objetos extraídos no passo anterior, com base na análise das imagens em associação com a Visão Computacional (GONZALEZ, 2009). Em outras palavras, é contextualizar aquele objeto na imagem, além de reconhecer, por exemplo, se um objeto é um cachorro ou um gato, inseri-los em um contexto como sentado, correndo no quintal, com medo, dentre outras situações, obviamente de modo automatizado pelo computador.

3.3.2 Extração de Atributos

Conforme explicado em Gonzalez (2009), no processamento *mid-level*, atributos são extraídos das imagens ou de Regiões de Interesse (ROIs) destas, para que possa ser possível, em um novo processamento, atribuir uma classificação do que é aquele objeto/ROI.

Segundo Cui, Huang e Liu (2018), tratando-se de extração de atributos envolvendo a face humana, existem duas linhas de pesquisa majoritárias: (i) extração de atributos das imagens cruas (*raw*), com análise dos pixels em si; ou (ii) extração de atributos advindos de *landmarks*, que serão melhor abordados na Seção 3.1. Em relação à primeira linha de pesquisa, pode-se destacar a análise da intensidade dos pixels que compõe a ROI, além de técnicas destacadas na literatura como Local Binary Pattern (LPB) (HE; WANG, 1989; WANG; HE, 1990), Scale-Invariant Feature Transform (SIFT) (LOWE, 1999), Haar-like features (VIOLA; JONES, 2001) e Histogram of Oriented Gradients (HOG) (DALAL; TRIGGS, 2005).

3.3.3 Seleção de Atributos

Cada atributo do conjunto de dados significa uma dimensão. Em conjuntos com baixas dimensões, analisar os vizinhos mais próximos de um determinado ponto é uma tarefa não tão custosa computacionalmente e em geral pode oferecer um bom resultado. Mas, à medida que a dimensionalidade cresce, a quantidade de pontos vizinhos que devem ser analisados cresce, já que em altas dimensões os vizinhos mais próximos estão bem distantes do ponto de interesse.

Esses pontos estão tão distantes entre si que em um conjunto com 200 dimensões, seria necessário analisar 94% dos pontos como vizinhos mais próximos para delimitar o ponto de interesse. Esse problema é descrito na literatura como maldição da dimensionalidade (NORVIG; RUSSELL, 2014). Por isto, selecionar atributos é uma técnica para tentar reduzir a dimensionalidade do conjunto de dados.

Selecionar atributos significa escolher dentre os disponíveis, aqueles que melhor representam a solução do domínio em que atuam. Contudo, o conceito de seleção de atributos é amplo, podendo significar uma melhora na acurácia final do preditor, facilitar a visualização e entendimento dos dados ou mesmo reduzir a dimensionalidade, impactando na performance de predição de treinamento (GUYON; ELISSEEFF, 2003).

Nem todos os tipos de algoritmos indutores conseguem lidar bem com atributos considerados irrelevantes para prover uma classificação, regressão ou agrupamento, portanto, um dos objetivos desta seleção é eliminar os atributos que não contribuem para a solução desejada (BLUM; LANGLEY, 1997).

Independente da finalidade escolhida para a seleção de atributos, a exemplo, seja para aumentar a acurácia ou a performance da predição, existem basicamente três tipos de metodologias para realizar essa seleção (BLUM; LANGLEY, 1997):

1. **Embutida** - consiste em algoritmos que incorporam um método de seleção de atributos como parte do processo de treinamento. Essa técnica, em geral, é considerada mais rápida, pois não há necessidade de dividir o conjunto de dados em treinamento e teste apenas para validar a seleção de atributos. Uma das formas dessa técnica embutida é realizar a estimativa das alterações da função objetivo, aquela que define a distribuição dos dados, alterando os subconjuntos, removendo ou adicionando atributo. Esta forma é denominada de cálculo de diferença finita. Na Aproximação Quadrática da Função (ou Aproximação de Taylor), ocorre a poda de pesos irrelevantes dados às variáveis de entrada em Redes Neurais. Por fim, uma terceira forma embutida de seleção de atributos é calcular a sensibilidade da função objetivo, baseada no quadrado da derivada da função objetivo com relação ao atributo selecionado, analisando se houve melhora ou não da sensibilidade.

2. **Filtragem** - significa remover atributos irrelevantes antes de realizar a indução do algoritmo, sendo esta técnica independente do algoritmo escolhido (BLUM; LANGLEY, 1997). Umas dos formas de realizar essa técnica é validar cada atributo individualmente correlacionado-o com a função objetivo e medir essa correlação, por meio de informação mútua ou regressão linear, a exemplo. Posteriormente, os atributos com a maior correlação são selecionados, validando essa seleção em um conjunto de teste (BLUM; LANGLEY, 1997). Pode-se citar os algoritmos RELIEF (KIRA; RENDELL, 1992) ou *Principal Component Analysis* (PCA) (JOLLIFFE; CADIMA, 2016) como opções de filtragem.

3. ***Wrapper*** - baseia-se na escolha de um subconjunto de dados que consiga detectar o padrão desejado com a melhor acurácia. Para tanto, é escolhido um subconjunto inicial e realizada uma indução em um algoritmo definido previamente. Após, um atributo é removido ou adicionado ao subconjunto inicial, sendo realizada uma nova

indução. A acurácia de ambas induções são comparadas, e, o subconjunto que produziu a maior acurácia é mantido e o processo se repete até que seja obtida a maior acurácia ou não haja mais subconjuntos para análise (GUYON; ELISSEEFF, 2003). Portanto, essa técnica realiza a análise combinatória dos atributos para encontrar o melhor subconjunto de atributos representativos. Entretanto, esse é um problema denominado NP-difícil (GUYON; ELISSEEFF, 2003), pois, à medida que a dimensionalidade aumenta, a combinação entre os atributos cresce exponencialmente 2^n, cujo n é o número de atributos total do conjunto (BLUM; LANGLEY, 1997).

3.4 Balanceamento de Classes

Existem diversos métodos para corrigir o desbalanceamento de classes para construir um classificador generalista capaz de classificar os dados adequadamente à heurística determinada. Dentre tais métodos, a análise da Precisão (Subseção 5.2.3) em conjunto com a Sensitividade (Subseção 5.2.4), assim como a medida F1 (Subseção 3.4) e técnicas como redimensionamento da amostragem, podem ser citadas.

O desbalanceamento decorre do processo de escolha dos exemplos que compuseram o conjunto de dados final. Por exemplo, nos vídeos que deram base para a extração das imagens de interesse, é comum que o indivíduo permanecesse um tempo maior realizando o movimento de sopro que o de beijo, já que expirar o ar naquele movimento é mais demorado que sua realização.

Para contornar esse problema existem técnicas como a subamostragem ou sobre-amostragem do conjunto de dados. No caso da subamostragem, para balancear as classes de uma forma igualitária, entende-se que as classes de maior valor devem ter seus exemplos removidos até se igualarem ao número de exemplos da classe de menor valor (CHAWLA, 2002). No cenário desta pesquisa, considerando a distribuição de classes como sendo de 7065 sopros, 3041 estalos de língua e 1850 beijos, segundo a subamostragem, deveriam ser descartados, aleatoriamente, 5215 imagens de sopros e 1191 imagens de estalos de língua, para que as três classes se igualassem em 1850 imagens cada.

Já a técnica da sobre-amostragem, considerando a classe majoritária como sendo o sopro com 7065 imagens, estalo de língua e beijo deveriam ter sua amostragem aumentada em 5215 e 4024 imagens respectivamente, para que todas as classes possuíssem um total de 7095 imagens cada.

A aplicabilidade de alguma das técnicas de redimensionamento de amostragem depende do classificador empregado. Por exemplo, algoritmos de AM como Árvore de Decisão, Support Vector Machine e Regressão Logística não se beneficiam com o aumento de dados (*Oversampling*), logo, a técnica de subamostragem (*Undersampling*) pode ser a

Figura 5 – Exemplo do algoritmo da técnica *SMOTE*

Consider a sample (6,4) and let (4,3) be its nearest neighbor.
(6,4) is the sample for which k-nearest neighbors are being identified.
(4,3) is one of its k-nearest neighbors.
Let:
f1_1 = 6 f2_1 = 4 f2_1 - f1_1 = -2
f1_2 = 4 f2_2 = 3 f2_2 - f1_2 = -1
The new samples will be generated as
(f1',f2') = (6,4) + rand(0-1) * (-2,-1)
rand(0-1) generates a random number between 0 and 1.

Fonte: (CHAWLA, 2002)

mais indicada. No caso de Redes Neurais, estas se beneficiam de um maior volume de dados, portanto, a aplicação da técnica de sobre-amostragem pode apresentar melhores resultados (ALOM, 2019; ZHU, 2016).

Considerando que neste trabalho foi aplicada uma Rede Neural, a técnica da sobre-amostragem foi utilizada para balancear as classes.

3.5 Método de Sobre-Amostragem *SMOTE*

Como visto, o método de sobre-amostragem tende a ser utilizado para aumentar a quantidade de amostras das classes minoritárias para igualá-las às quantidades das classes majoritárias. Uma das técnicas mais utilizadas para cumprir esse objetivo é a Técnica de Super-Amostragem Sintética da Minoria (*SMOTE - Synthetic Minority Over-sampling Technique*) (CHAWLA, 2002) - Figura 5.

A técnica consiste em selecionar os k-vizinhos mais próximos de um exemplo da amostra minoritária, tirar a diferença entre o exemplo e seu vizinho mais próximo, multiplicar essa diferença por um número gerado randomicamente entre 0 e 1 e adicionar o resultado dessa multiplicação à diferença inicial (CHAWLA, 2002).

Na Figura 5, o exemplo (vetor de atributos) (6,4) é retirado da amostra minoritária. O vizinho mais próximo, considerando os k-vizinhos mais próximos, é o vetor de atributos (4,3). A diferença entre esses dois vetores de atributos é extraída (4-6) e (3-4), resultado em um novo vetor (-2,-1). Ao exemplo original (6,4) é somada a multiplicação de um número randômico entre 0 e 1 e o novo vetor gerado (-2,-1).

Considerando o objetivo desta pesquisa, de produzir o modelo induzido genérico

o bastante capaz de predizer novos movimentos de interesse, a heurística de avaliação escolhida é o modelo que produza o menor número de falsos positivos possíveis, para garantir que o paciente realize apenas os movimentos idênticos aos classificados como corretos pelo indutor. Nesse intuito, métricas como Precisão, Sensibilidade, Acurácia e principalmente *Log Loss* (Capítulo 6) serão observadas.

3.6 Normalização dos Dados

Os dados coletados por meio de qualquer processo de extração, vão estar distribuídos em algum tipo de intervalo, ainda que infinito na População, mas dentro da amostragem, em intervalos definidos (PYLE, 1999).

Um dos problemas que ocorrem na distribuição de dados numéricos é a variação entre os valores mínimo e máximo que são amostrados. Imagine, a exemplo, a medição de distâncias entre os pontos, resultado nos atributos A e B como sendo 1000 e 12, respectivamente. O fato de 1000 ser maior que 12 não deve ter qualquer peso sobre a inferência que será realizada. Mas de fato, se utilizados os valores brutos, o atributo A receberá um peso muito maior que o atributo B e consequentemente terá uma influência maior ou soberana no treinamento do algoritmo indutor (PYLE, 1999).

Assim, temos a introdução da técnica de normalização que consiste em remapear os valores dos atributos extraídos num intervalo que varie entre 0 e 1 ou entre -1 e +1, com o intuito de suavizar diferenças relevantes entre os valores do intervalo de um atributo. Com o mesmo intervalo, todos os atributos adquirem, inicialmente, a mesma relevância no treinamento e o peso que receberão vai mudar conforme a própria variância de seus dados progridem para a validação do modelo induzido (PYLE, 1999).

Uma técnica de normalização para converter um intervalo em outro entre 0 e 1 é a técnica de *Min-Max Scaling* ou Redimensionamento que consiste na seguinte fórmula:

$$V_n = \frac{v_1 - \min(v_1..v_n)}{\max(v_1..v_n) - \min(v_1..v_n)} \tag{3.1}$$

Na Equação 3.1 V_n representa o valor final normalizado, que é constituído do exemplo a normalizar v_1 menos o valor mínimo do vetor do atributo sendo normalizado, esse resultado é dividido pelo resultado do valor máximo menos o valor mínimo do vetor (PYLE, 1999).

3.7 Medições de Distância

Existem diversos métodos para realizar a medição de distâncias entre pontos como, por exemplo, *landmarks*, ou até entre vetores, constituídos entre vários pares de pontos, dentre as quais destacam-se:

$$\text{Euclidiana} = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \tag{3.2}$$

$$\text{Manhattan} = (p, q) = \sum_{i=1}^{n} |p_i - q_i| \tag{3.3}$$

$$\text{Minkowski} = (X, Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{3.4}$$

$$\text{Canberra} = (p, q) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|} \tag{3.5}$$

$$\text{Mahalanobis} = (x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \tag{3.6}$$

A Equação 3.2 é a distância que constitui a menor reta entre dois pontos (p, q) em um espaço geométrico, onde p e q são coordenadas distintas num plano bi-dimensional que constituem um par de coordenadas. O cálculo é composto pela raiz quadrada da soma da diferença ao quadrado dos pontos (p, q).

Na Equação 3.3, verifica-se a somatória entre a diferença modular de dois pontos (p, q), onde p e q são coordenadas distintas num plano bi-dimensional que constituem um par de coordenadas e n é o número de pares de coordenadas, a qual gera retas verticais e horizontais em um plano bi-dimensional para constituir a distância entre os pontos (p, q). A Distância de *Manhattan* (representada na Equação 3.3), também denominada de Distância *city-block* pois equivale-se ao percurso entre dois pontos, respeitando-se os quarteirões entre o caminho, como em uma cidade, a exemplo.

A Equação 3.4 é a distância de ordem p entre dois pontos (x, y) composta pela somatória da diferença do módulo de (x, y) elevados a p, a um sobre p, onde x e y são pontos de coordenadas no plano bi-dimensional, n é o número de pares de coordenadas e p é um inteiro que significa a ordem de grandeza entre os pontos x e y. Quando $p = 1$, a equação se torna idêntica à medição da Distância *Manhattan*, quando $p = 2$ possui o mesmo resultado da Distância Euclidiana. Quando $p > 2$, tendendo ao infinito, tem-se que a Distância *Minkowski* será idêntica à Distância denominada *Chebyshev*[7]

[7] É considerada como:

$$D_{\text{Chebyshev}}(x, y) := max(|\ x_i - y_i\ |). \tag{3.7}$$

A Distância de *Canberra* (Equação 3.5) consiste da somatória do módulo da diferença entre dois pontos (x, y), dividido pela soma do módulo de x e do módulo de y. Os pontos x e y são coordenadas de um par de coordenadas no plano bi-dimensional e n é a quantidade total de pares de coordenadas. Cada termo da fração terá valores entre 0 e 1, mas não o resultado da somatória em si. A distância de *Canberra* é similar à Distância Manhattan, com exceção da divisão pela somatória do valor absoluto dos pontos (x, y), o que torna essa distância mais sensível à pequenas variações quando os valores dos termos da fração são próximos de zero.

Por fim, a Equação 3.6 representa a Distância *Mahalanobis*, a qual é a medida de variação entre dois vetores de distâncias, compostos pelos pares de coordenadas (x, y). Essa distância é constituída da raiz quadrada da matriz transposta (T) da diferença entre a média (μ) do vetor $v = (x)$ e do vetor $v = (y)$, multiplicada pela matriz inversa de covariância $(S^{-}1)$, sendo esta multiplicada pela matriz da diferença entre a média do vetor $v = (x)$ e do vetor $v = (y)$. Mede-se a separação de dois grupos de variáveis, nesse caso, os pares de distâncias (x, y).

Nesta pesquisa, foi utilizada a distância Euclidiana, considerando que essa distância aplicada em conjunto *landmarks*, possui a característica de não ser afetada por rotação, redimensionamento ou translação [8] da imagem, conforme apresentado em Cui, Huang e Liu (2018), por isso, tendem a manter uma melhor acurácia, dependendo do domínio. A exemplo de Wu, Liu, Xu, Gao, Li e Zhang (2017), os autores utilizam a distância Euclidiana, especificamente, para medir as alterações faciais, sendo realizado o treinamento de uma rede neural de três camadas, construídas especificadamente para o problema em questão, pois aceitam ser um método eficiente para essa finalidade. O modelo generalizado foi aplicado em quatro conjuntos de dados distintos, UvA-NEMO[9] (DIBEKLIOĞLU, 2012), BBC[10], SPOS[11] (PFISTER, 2011) e MMI[12] (VALSTAR; PANTIC, 2010), obtendo a maior acurácia diante de outros trabalhos correlatos (COHN; SCHMIDT, 2004; DIBEKLIOĞLU, 2010; PFISTER, 2011; DIBEKLIOĞLU, 2012; DIBEKLIOĞLU, 2015)

3.8 Aprendizado de Máquina

O Aprendizado de Máquina (AM) consiste na melhoria das tarefas realizadas por um programa, com base na experiência que esse programa obtém ao longo do tempo, ou seja,

podendo ser considerada como o módulo da distância entre dois pontos específicos ou a diferença entre o valor máximo entre dois vetores. Onde x e y são coordenadas de um par de coordenadas num plano de duas dimensões.

[8] Translação é o ato de mover todos os pontos de uma imagem em uma mesma distância e direção

[9] Disponível em https://www.uva-nemo.org

[10] Disponível em http://www.bbc.co.uk/science/humanbody/mind/surveys/smiles/

[11] Disponível em https://www.oulu.fi/cmvs/node/41317

[12] Disponível em https://mmifacedb.eu

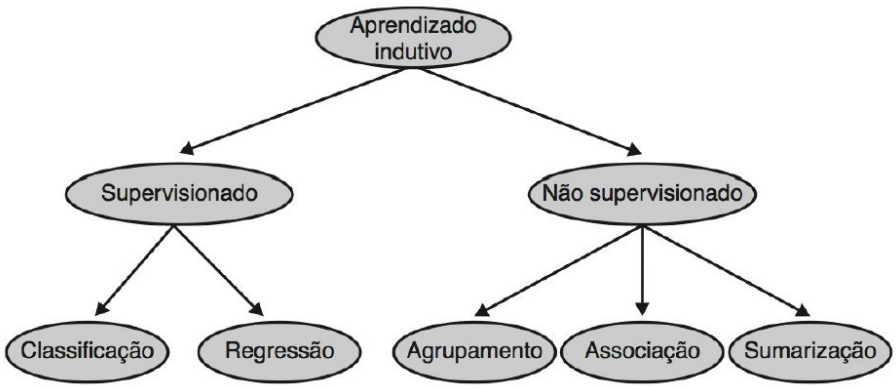
a capacidade da máquina em aprender (MITCHELL, 1997).

Existem determinados métodos para que este aprendizado ocorra, os quais são descritos como derivados do aprendizado indutivo, que tem por finalidade produzir um modelo genérico capaz de ser aplicado a outros modelos. Advém também do método de aprendizado indutivo as formas supervisionada e não supervisionada. A forma supervisionada, desmembra-se em classificação e regressão, que consistem em receber um conjunto de dados já rotulados com classes discretas, no caso da classificação, ou dados contínuos, no caso da regressão (GAMA, 2011).

Na forma não supervisionada não existe a rotulação dos dados, sendo que esta consiste em agrupar os dados baseados em suas semelhanças, sumarizá-los, que constitui em uma forma de descrever os dados ou, por fim, associá-los, que significa determinar se existem padrões nos dados não rotulados apresentados (GAMA, 2011).

A Figura 6 ilustra as formas de Aprendizado de Máquina (GAMA, 2011).

Figura 6 – Tipos de métodos de aprendizados utilizados para desenvolvimento de modelos preditivos em Aprendizado de Máquina e Redes Neurais



Fonte: (GAMA, 2011)

Além dos tipos de aprendizados de Gama, Faceli, Lorena e Carvalho (2011), existem também o aprendizado semi-supervisionado, que é uma forma mista entre as citadas acima, cuja uma pequena parte do conjunto de dados está rotulado, mas a grande maioria não (CHAPELLE, 2006).

Por fim, existe também o aprendizado por reforço, o qual se designa a aprender o que fazer, contudo o programa não é ensinado a fazer, mas sim a analisar o que foi feito e depois, mediante tentativa e erro, é recompensado pelas práticas que levam ao objetivo principal. Um grande exemplo seria jogar um jogo, como o jogo da velha, cujos movimentos e contra-movimentos que levam à vitória, são recompensados (SUTTON, 1998).

Este trabalho explorou o aprendizado indutivo supervisionado para classificar movimentos de praxias não verbais, como beijo, sopro e estalo de língua, analisando dados extraídos de imagens estáticas de pessoas realizando esses movimentos.

3.9 Considerações Finais

Neste capítulo, foram apresentadas técnicas para desenvolver um modelo capaz de generalizar movimentos na face humana, baseando-se no processamento e análise de atributos extraídos de *landmarks*. Como explorado em (CUI, 2018), foi realizada a extração de um vetor de pares de distâncias Euclidianas para compor um conjunto de dados que foi utilizado para realizar o aprendizado supervisionado de diversos indutores, incluindo modelos de Redes Neurais, com o intuito de produzir um modelo generalizado o bastante para classificar entre os movimentos de interesse: beijo, sopro e estalo de língua.

4

Amostragem

Uma das primeiras atividades executadas nesta pesquisa foi a construção de um conjunto de dados com imagens que representassem o padrão almejado a ser reconhecido, como movimento de beijo, sopro e de estalo de língua. Utilizou-se técnicas de Aprendizado de Máquina para reconhecer padrões na fala, uma vez que a identificação dos padrões e a aplicação de forma automatizada pode tornar mais eficaz o tratamento do paciente com problemas de fala.

Foram incrementados os conjuntos de dados de treinamento e teste, respectivamente, para agregar ao campo de Visão Computacional matéria prima para resoluções de problemas envolvendo os movimentos de interesse na face humana e outros correlatos que possam se valer dos mesmos dados.

Nos três conjuntos de dados construídos, nenhum dos voluntários possuíam deficiência físicas ou intelectuais que influenciassem diretamente a coleta dos dados.

4.1 Definições Iniciais

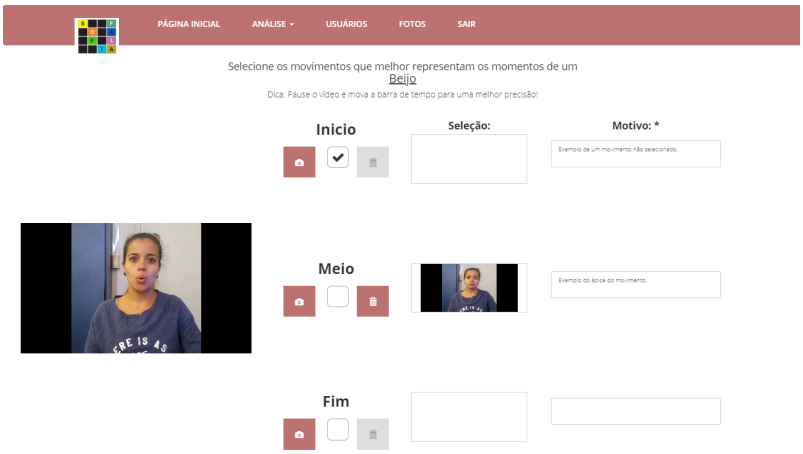
A coleta de dados iniciou em Março/2018 e findou em Junho/2018. Foram selecionados 7 voluntários, sendo 3 mulheres e 4 homens[1]. A idade dos participantes varia entre 25 a 45 anos. Duas das voluntárias eram profissionais da área fonoaudiológica. Não foram estabelecidos critérios de inclusão para os participantes.

O único critério de exclusão estabelecido era a avaliação dos vídeos pelas profissionais fonoaudiológicas participantes de que todos os demais voluntários haviam realizado corretamente os movimentos de interesse, do ponto de vista fonoaudiológico.

Cinco dos voluntários são da raça branca, um deles é asiático e o outro é pardo. Dos voluntários homens, somente um deles possuía barba e bigode, os demais não possuíam

[1] No termo de consentimento assinado pelos participantes, não foi aplicada questão acerca de qual gênero os indivíduos se identificam, e, sim a percepção sobre o gênero aparente.

Figura 7 – Website desenvolvido para proporcionar a captura de avaliações realizadas por profissionais fonoaudiólogos, onde imagens estáticas eram selecionadas para indicar o início, meio e fim dos movimentos de beijo, estalo de língua e sopro, em conjunto com a justificativa para a seleção



Fonte: Autoria própria

nem barba nem bigode. Das três voluntárias mulheres, todas estavam com o cabelo preso, e apenas uma delas usava adereços (brincos). Um dos voluntários utilizava capuz e outro utilizava óculos de grau translúcido.

Não havia nenhuma oclusão intencional parcial ou total da face dos voluntários, contudo, três dos voluntários homens tiveram a parte superior da testa fora do enquadramento das filmagens.

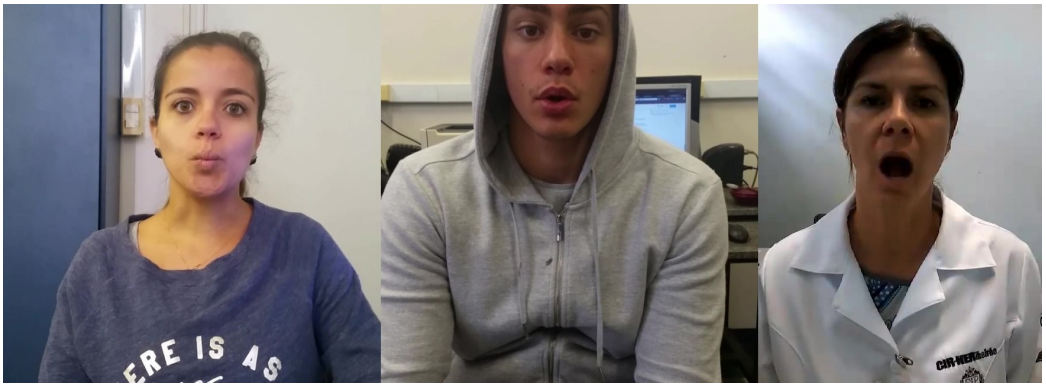
Todos os voluntários foram posicionados frontalmente à câmera onde parte do peitoral, ombros e a cabeça estavam em destaque. O fundo das filmagens é constituído em grande parte por portas, interruptores, monitores e computadores e paredes lisas. Um dos fundos era constituído de material de isolamento acústico, pois uma das voluntárias estava um estúdio profissional de gravação de áudios.

Para realizar a amostragem, foi elaborado um website (veja na Figura 7) para que os profissionais de fonoaudiologia, vinculados ao projeto CNPq "SofiaFala", pudessem selecionar as imagens que ilustrassem o movimento ideal que deveria ser realizado pelo paciente com TF. Para cada tipo de movimento, deveria ser selecionada uma etapa entre o início, meio e fim, para que fosse possível definir toda a execução do movimento.

A partir da coleta dos dados iniciais, foram disponibilizados 21 vídeos únicos, que foram filmados individualmente realizando os três movimentos de interesse (beijo, estalo de língua e sopro). A Figura 8 ilustra três dos voluntários posicionados para as filmagens.

Não havia um padrão pré-estabelecido para as câmeras que capturaram as imagens. Todas os vídeos foram obtidos por câmeras de dispositivos móveis, filmadas sem um suporte fixo.

Figura 8 – Representação de três indivíduos, realizando a execução dos movimentos de beijo, estalo de língua e sopro que foram disponibilizados no website desenvolvido para a avaliação dos profissionais fonoaudiólogos



Fonte: Autoria própria

4.2 Construção do Conjunto de Dados Iniciais

A seleção de profissionais para avaliação ocorreu entre o período de 15/05/2018 a 22/06/2018, quando foram convidados quinze profissionais da área da fonoaudiologia, sendo nove profissionais formados/habilitados e seis alunos da graduação. Ao final, um total de seis pessoas aceitaram o convite para realizarem as seleções, sendo dois alunos de graduação e quatro profissionais habilitados, denominados de avaliadores.

Considerando 21 vídeos no total gerados pela coleta de dados iniciais, e, três tempos diferentes de análises por vídeo (início, meio e fim do movimento), cada um dos seis avaliadores poderia realizar um total de 63 análises. O total máximo de análises teóricas possíveis seria de 378 (63 análises possíveis * 6 avaliadores); porém os avaliadores conseguiram identificar 280 movimentos não ambíguos que correspondem a aproximadamente 75% do total disponível.

Foram produzidas 244 imagens dos movimentos de interesse e 36 justificativas da não possibilidade de seleção do movimento de interesse[2] conforme disposto na Tabela 3.

[2] Os motivos das justificativas relativos à impossibilidade das análises estão disponíveis no Anexo C.

Tabela 3 – Total de análises realizadas no website por profissionais fonoaudiológicos participantes para a composição do conjunto de dados inicial

Movimento	Imagens geradas	Não analisáveis	Total
beijo	114	10	124
estalo de língua	59	25	84
sopro	71	1	72
Subtotal	244	36	280

Fonte: Autoria própria

Pode-se visualizar na Tabela 4 as quantidades selecionadas de cada movimento individualmente.

Tabela 4 – Total de análises realizadas no website por avaliadores fonoaudiológicos participantes, para a construção do conjunto de dados inicial

Movimento	Início	Meio	Fim	Total
beijo	37	39	38	114
estalo de língua	19	20	20	59
sopro	24	23	24	71
Subtotal	80	82	82	244

Fonte: Autoria própria

Entretanto, ao analisar as imagens selecionadas, foi percebido que os movimentos de início e fim selecionados pelos profissionais fonoaudiológicos eram idênticos, em geral, a uma boca em repouso.

Nas Figuras 9, 10 e 11 pode-se ver exemplos das seleções da fase inicial dos movimentos de beijo, estalo de língua e sopro respectivamente.

Figura 9 – Três indivíduos do conjunto de dados iniciais demonstrando a fase inicial do movimento de beijo selecionado por especialistas fonoaudiológicos

Fonte: Autoria própria

Figura 10 – Três indivíduos do conjunto de dados iniciais demonstrando a fase inicial do movimento de estalo de língua selecionado por especialistas fonoaudiológicos

Fonte: Autoria própria

Figura 11 – Três indivíduos do conjunto de dados iniciais demonstrando a fase inicial do movimento de sopro selecionado por especialistas fonoaudiológicos

Fonte: Autoria própria

Referente à fase final, pode ser observado nas Figuras 12, 13 e 14 referente aos movimentos de beijo, estalo de língua e sopro.

Figura 12 – Três indivíduos aleatórios do conjunto de dados iniciais demonstrando a fase final do movimento de beijo selecionado por especialistas fonoaudiológicos

Fonte: Autoria própria

Figura 13 – Três indivíduos aleatórios do conjunto de dados iniciais demonstrando a fase final do movimento de estalo de língua selecionado por especialistas fonoaudiológicos

Fonte: Autoria própria

Figura 14 – Três indivíduos aleatórios do conjunto de dados iniciais demonstrando a fase final do movimento de sopro selecionado por especialistas fonoaudiológicos

Fonte: Autoria própria

Considerando o experimento, as fases de início e fim foram descartadas para indução do modelo, já que se assemelham à boca em repouso (boca fechada ou semi fechada) e podem gerar falsos positivos. Portanto, apenas as fases do meio do movimentos, com 82 imagens no total foram selecionadas de cada um dos movimentos de interesse para compor o conjunto de dados iniciais.

Esse conjunto de dados iniciais se demonstrou ineficaz para induzir um modelo genérico capaz de predizer novos exemplos de treinamento. Os motivos dessa incapacidade são apresentados e discutidos na Seção 6.1.

4.3 Coleta de Dados de Treinamento

A amostragem para treinamento foi inicialmente elaborada em Souza, Souza, Watanabe, Mandrá e Macedo (2019) e incrementada na presente pesquisa. Foram produzidos 356 vídeos com indivíduos realizando o movimento de beijo, 332 vídeos com o movimento de estalo de língua e 348 com o movimento de sopro, totalizando 1036 vídeos. A coleta de vídeos começou em Abril/2017 e finalizou em Junho/2018.

Foram 14 voluntários ao todo, sendo 6 mulheres e 8 homens, variando entre 24 e 50 anos de idade. Duas das voluntárias eram profissionais da área fonoaudiológica. Não foram estabelecidos critérios de inclusão para os participantes. Destes voluntários, oito são brancos, quatro amarelos e dois pardos.

Esses vídeos foram gravados por treze voluntários, sendo seis do sexo feminino e sete do sexo masculino, enquadrados frontalmente à câmera de dispositivos móveis variados, realizando os movimentos de interesse. A Figura 15 ilustra quatro dos voluntários posicionados para as filmagens.

Figura 15 – Quatro indivíduos posicionados frontalmente com foco na face para filmagem dos movimentos de interesse para composição da amostra dos dados para treinamento

Fonte: Autoria própria

Os vídeos foram capturados nas seguintes resoluções 480x640, 640x416, 720x480, 1080x1920, 1280x720, 1920x1080 e em geral a 30 quadros por segundo. Os dados individuais sobre a coleta dos vídeos realizada podem ser observados no Anexo A.

4.4 Conjunto de Dados de Treinamento

Para realizar a amostragem, primeiramente, utilizou-se técnicas de Aprendizado de Máquina para reconhecer padrões na fala, uma vez que a identificação dos padrões e a aplicação de forma automatizada pode tornar mais eficaz o tratamento do paciente com problemas de fala.

Para realizar as induções pretendidas foi necessário extrair as imagens estáticas dos 1.036 vídeos selecionados. Para a extração, foi utilizado o programa *ffmpeg* [3] que extraiu 79284 imagens, sendo, 26949 são dos vídeos de movimentos de beijo, 25080 de movimentos de estalo de língua e 27255 de movimentos de sopro. As imagens extraídas possuem as mesmas resoluções dos vídeos que as originaram.

Após a extração, foi iniciada a seleção das imagens para filtrar apenas as imagens que representassem o ápice de cada movimento. O processo foi realizado manualmente utilizando ferramenta própria desenvolvida para uma seleção mais performática [4].

Portanto, a amostragem final para treinamento é composta por 1850 imagens de beijos, 3401 imagens de estalo de língua e 7065 imagens de sopros, totalizando 12316 imagens.

4.5 Coleta de Dados de Teste

O processo de construção da amostra de testes foi idêntico ao da amostra de treinamento, contudo, todos os voluntários que compuseram essa amostragem não fizeram parte da amostragem de treinamento. Os vídeos com os movimentos de interesse foram obtidos por meio de redes de contatos em efeito bola de neve (*snowballing*), pois um contato solicitava para outro gravar seguindo instruções de um vídeo de autoria própria, e assim por diante. O processo de coleta iniciou em Setembro/2019 e finalizou em Fevereiro/2020. Nenhum de todos os voluntários possuía qualquer tipo de deficiência intelectual ou física aparente.

Foram recebidos 199 vídeos no total, sendo 66 vídeos do movimento de beijo, 67 vídeos do movimento de estalo de língua e 66 vídeos do movimento de sopro. Os vídeos são de 64 voluntários, sendo 39% (25) desses voluntários do sexo masculino e 61% (39) voluntários do sexo feminino, não propositalmente. Tratando-se de raças, 59% (38) dos voluntários são brancos, 33% (21) são pardos, 5% (3) são amarelos e 3% (2) são negros.

[3] *ffmpeg* é um *framework* multimídia capaz de codificar, decodificar, transcodificar, multiplexar, desmutiplexar, transmitir e tocar qualquer arquivo multimídia criado. Disponível em https://ffmpeg.org/download.html. A versão utilizada neste projeto foi a 3.4.8-0ubuntu0.2.

[4] Disponível em https://github.com/pedrohenriquerissato/ImageViewer.

A Figura 16 demonstra as posições dos voluntários paras as filmagens.

Figura 16 – Quatro indivíduos aleatórios posicionados frontalmente com foco na face para filmagem dos movimentos de interesse para composição da amostragem para validação

Fonte: Autoria própria

Não foram estabelecidos critérios de inclusão ou exclusão dos participantes. Do total de voluntários, 61% não utilizam nenhum adereço, 16% utilizavam óculos de grau/-translúcido, 8% utilizavam algum tipo de colar, 6% utilizavam brincos e algum tipo de colar, 5% utilizavam somente brincos, 3% utilizavam *piercing* no nariz e algum tipo de colar e uma pessoa utilizava boné e outra brincos e óculos.

Das voluntários mulheres, 72% estavam com cabelo solto e 28% com o cabelo preso. Em relação aos voluntários masculinos, 48% (12) desses não possuíam barba ou bigode e 40% (10) possuíam barba e bigode, 8% (2) eram calvos e 4% (1) possuíam barba sem bigode.

4.6 Conjunto de Dados de Teste

Foram extraídas 14749 imagens dos vídeos capturados. Estas imagens foram selecionadas manualmente utilizando metodologia e ferramentas idênticas às empregadas na amostragem de treinamento.

Ao final, foram selecionadas 642 imagens do movimento beijo, 726 imagens do movimento de estalo de língua e 1948 imagens do movimento de sopro, totalizando 3316 imagens. As imagens foram selecionadas após a extração dos quadros de todos os vídeos e o movimento de "meio", considerado o ápice dos movimentos de interesse, foram selecionados de cada quadro extraído.

A Tabela 5 apresenta todas as resoluções dos vídeos obtidos na amostragem para compor os conjuntos de dados inicial, treinamento e testes.

Tabela 5 – Resoluções das imagens obtidas para cada tipo de conjunto de dados: (IN) inicial, (TR) treinamento e (TE) teste

	RESOLUÇÕES																						
MOVIMENTO	1920 x 1080		1080 x 1920	856 x 480	720 x 1280		720 x 480		640 x 640	640 x 352	540 x 960	480 x 864	480 x 848		480 x 640			480 x 272	464 x 848	450 x 800	360 x 640	416 x 640	352 x 640	320 x 568
	IN	TR	TR	IN	TR	TE	TR	IN	TE	TE	TE	TE	TE	IN	TR	TE	TE	TE	TE	TE	TR	TE	TE
beijo	27	211	1264	-	-	-	330	6	15	3	7	10	113	6	35	17	53	25	10	28	10	320	41
estalo	15	343	2447	3	98	13	457	-	5	-	9	7	155	2	56	9	21	8	17	17	-	412	53
sopro	16	1030	4835	3	603	86	477	-	28	5	40	21	438	4	120	10	41	40	-	47	-	1154	38

Fonte: Autoria própria

Após a definição das amostras para construção dos conjuntos de dados de treinamento e teste, foi definido o procedimento a ser utilizado para construção de coleção de dados através da técnica conhecida como *landmarks*, considerando o bom desempenho apresentado em Dibeklioğlu, Salah e Gevers (2015), Jin, Qu, Zhang e Gao (2020), Lekdioui, Ruichek, Messoussi, Chaabi e Touahni (2017) no reconhecimento de padrões envolvendo a face humana.

O procedimento criado estabelece marcadores no rosto para extrair dados e medidas. Em seguida, o procedimento estabelece um protocolo para a extração de distâncias Euclidianas entre os marcadores definidos. Com estes marcadores foi realizada a análise da classificação de movimentos não articulatórios.

O Algoritmo 1 expressa os passos essenciais do procedimento para obtenção dos dados necessários para a indução de um modelo capaz de reconhecer os movimentos de beijo, sopro e estalo de língua na face humana.

De acordo com o Algoritmo 1, uma imagem do movimento de interesse é lida, juntamente com um rótulo indicando qual é o movimento. Essa imagem é enviada para um algoritmo da biblioteca *dlib*[5] capaz de detectar faces humanas (linha 1). Se uma face humana for encontrada, então a imagem é enviada para outro algoritmo (SAGONAS, 2013; KAZEMI; SULLIVAN, 2014) para detectar os *landmarks* (linha 3). Caso os *landmarks* sejam localizados, são selecionados 20 *landmarks* da boca (linhas 3, 4 e 5). Posteriormente, é feita uma análise combinatória simples[6] desses vinte *landmarks* da boca, é realizado o cálculo das distâncias Euclidianas conforme a Equação 3.2 e essas distâncias são armazenadas em um vetor (linhas 6, 7 e 8), juntamente com os rótulos contendo o movimento realizado na imagem analisada (linha 8).

[5] Disponível em: http://dlib.net/.

[6] $C_n^r = \binom{n}{r} = \frac{n!}{r!\cdot(n-r)!}$.

Algoritmo 1 Construção do Conjunto de Dados

Entrada: Imagem contendo a face humana e o movimento de interesse.
Saída: Conjunto de Dados Normalizado.

1: Recebe a imagem ($\boldsymbol{I}$) juntamente com o *rótulo* ($\boldsymbol{L}$) do movimento de interesse, analisa para encontrar um rosto:
$Face \leftarrow EncontrarFace(\boldsymbol{I})$
2: **if** $Face$ não está Ø **then**
3: Analisa $Face$ para verificar se existem *landmarks*
$Landmarks \leftarrow EncontrarLandmarks(Face)$
4: **if** $Landmarks$ não está Ø **then**
5: Localiza os 20 *landmarks* da boca:
$Boca \leftarrow PegarLandmarksBoca(Landmarks)$
6: **for** $x \leftarrow 0$ to 20 **do**
7: **for** $y \leftarrow$x+1 to 20 **do**
8: Calcula a distância entre o ponto x e y e adiciona em um vetor:
$Dados \leftarrow VetorDistancias \leftarrow Distancia(Boca[\mathbf{x}], Boca[\mathbf{y}]), \mathbf{L}$
9: **end for**
10: **end for**
11: **end if**
12: **end if**
13: **if** $Dados$ não está Ø **then**
14: O conjunto de $Dados$ é normalizado baseado em $l_1 - norm$:
15: $Dnorm \leftarrow$ Normalização($\boldsymbol{Dados}$)
16: **return** $Dnorm$
17: **end if**
18: **return** Ø

Ao terminar de processar todas as imagens, havendo qualquer vetor de distância registrado, o conjunto de dados é normalizado utilizando-se a regra $l_1 - norm$[7] (linha 15) e ao final o conjunto de dados normalizado é retornado (linha 16). No caso do conjunto de dados não conter nenhuma distância, significa que a face humana ou os *landmarks* não foram encontrados em nenhuma das imagens apresentadas e o ocorrido deve ser investigado. Nesse caso, o algoritmo retorna vazio (linha 18).

Por meio do Algoritmo 1, foi construído o conjunto de dados de treinamento composto por 190 pares de distâncias Euclidianas da análise combinatória entre os pontos 49 a 68 (Figura 4), resultando em um total de 12316 exemplos com seus respectivos rótulos de classe (beijo, estalo de língua e sopro).

Dos 12316 exemplos do conjunto de dados de treinamento, 1850 são do movimento de beijo, 3401 do movimento de estalo de língua e 7065 do movimento de sopro.

A mesma metodologia para criação do conjunto de dados de treinamento foi apli-

[7] Também conhecido como desvio padrão absoluto, do inglês *least absolute deviation*, uma técnica que consiste em encontrar a função que melhor define o conjunto de dados, traçando uma linha reta em um plano bi-dimensional, a qual minimiza a soma dos erros absolutos.

cada para a construção do conjunto de dados de teste, onde um total de 3316 exemplos do conjunto de dados de teste são distribuídos entre 642 do movimento de beijo, 726 do movimento de estalo de língua e 1.948 do movimento de sopro. Este conjunto de teste representa 23% do total de exemplos do conjunto de treinamento.

4.7 Distribuição de Classes

Considerando um conjunto de dados com n exemplos é necessário determinar qual a distribuição de classes para distinguir quais as classes majoritária e minoritária do conjunto de dados.

Pode-se calcular a distribuição de classes utilizando a equação:

$$distr(C_j) = \frac{1}{n} \sum_{i=1}^{n} || \, y_i = C_j \, || \tag{4.1}$$

cuja a Distribuição da Classe j, denominada por $distr(C_j)$ é composta pela multiplicação do total de exemplos da classe j, representado por y_i, pela somatória do total de exemplos do conjunto de dados, representado por n.

Considerando o conjunto de dados de treinamento apresentado em 4.4, aplicando-se a fórmula da distribuição, a distribuição entre as classes beijo, estalo de língua e sopro obtida foi:

$$distr(beijo) = beijo \div \text{total de exemplos} = 1850 \div 12316 = 0,1502 \tag{4.2}$$

$$distr(estalo) = estalo \div \text{total de exemplos} = 3401 \div 12316 = 0,2761 \tag{4.3}$$

$$distr(sopro) = sopro \div \text{total de exemplos} = 7065 \div 12316 = 0,5736 \tag{4.4}$$

Conclui-se portanto que no conjunto de dados utilizado, a classe majoritária é a classe Sopro com um distribuição de 57,36% do total do conjunto e a classe minoritária é a classe Beijo com 15,02% da distribuição total do mesmo conjunto.

Consequentemente, é possível observar que existe um desbalanceamento entre as classes, ocorrido naturalmente por meio do processo de seleção dos exemplos do conjunto de dados. Esse desbalanceamento natural entre as classes impacta diretamente no denominado erro majoritário [8], que não possui relação com o classificador utilizado. Pode-se observar na Equação 4.5 o cálculo desse erro majoritário:

[8] O erro majoritário de um conjunto é definido como sendo 1 menos a distribuição da classe majoritária e estabelece um limite que o erro do classificador deve ficar abaixo.

$$erro - majoritario(Conjunto) = 1 - \text{máx}\{distr(C_i)\} = 1 - 0,5736 = 0,4264 \quad (4.5)$$

onde o erro majoritário do conjunto é dado pelo erro total possível, 1, subtraído da distribuição da classe majoritária, representada por máx$\{distr(C_i)\}$.

4.8 Considerações Finais

Como observado no decorrer do presente capítulo, apresentou-se, aqui, o processo de criação de um conjunto de dados, denominado de conjunto de dados iniciais, baseado em experimentação com profissionais fonoaudiológicos, website específico para coletar as primeiras imagens e metodologia utilizada pelos profissionais para seleção dos melhores movimentos de interesse. Posteriormente, foi criado o conjunto de dados de treinamento, com uma quantidade de imagens de pessoas voluntárias no laboratório do projeto, as quais, realizaram os movimentos de interesse (beijo, estalo de língua e sopro). Por fim, um outro conjunto de dados, descrito como conjunto de dados de teste, foi elaborado com pessoas totalmente diversas das que participaram do conjunto de dados de treinamento. O conjunto de dados de teste foi experimentado com classificadores induzidos para avaliar a predição de novos exemplos e treinar uma rede neural para classificar entre os movimentos de interesse. Os resultados do desempenho desses classificadores são apresentados no Capítulo 6.

5

Método de Reconhecimento Visual de Praxia Não Verbal

Este capítulo apresenta o método desenvolvido para identificar padrões de movimentos não articulatórios como beijo, sopro e estalo de língua na boca humana, o qual foi denominado Reconhecimento Visual de Praxia Não Verbal (RVPNV). O método RVPNV utiliza técnicas e outros métodos das áreas de Visão Computacional e de Aprendizado de Maquina. A motivação foi a produção de um método genérico o suficiente para predizer novos exemplos dos movimentos de interesse com o melhor balanço entre precisão e sensibilidade.

5.1 Processo de Indução

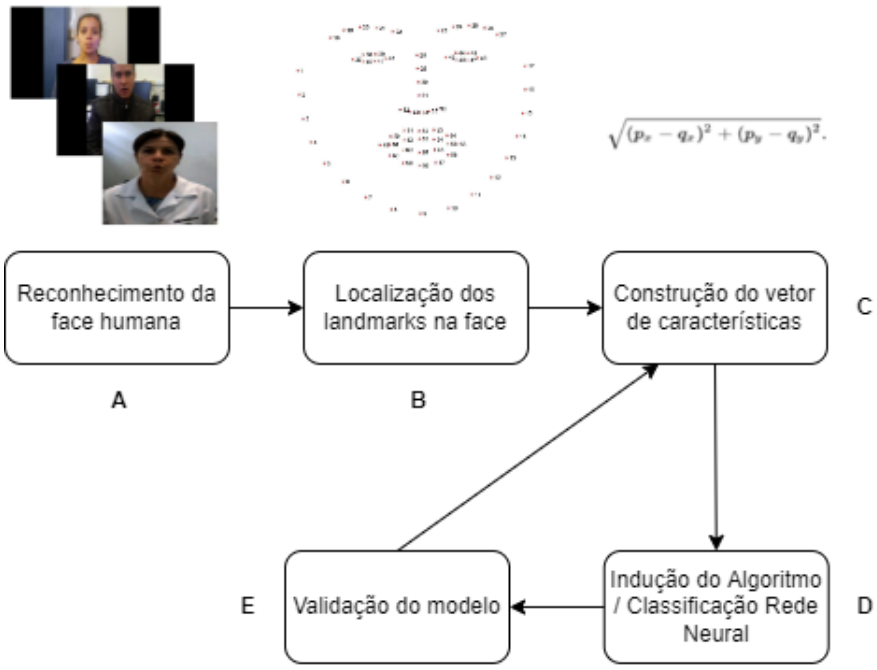
O processo de indução do algorítimo escolhido no presente trabalho contou com diversas etapas (A-E), que são descritas a seguir e foram sumarizadas na Figura 17.

Na etapa A é aplicada uma técnica de reconhecimento facial por meio da biblioteca dlib. Foi utilizada a implementação da linguagem Python[1] na versão 19.18 por meio da função get_frontal_face_detector(). O algoritmo proveniente da citada função, foi desenvolvido baseado no trabalho de Dalal e Triggs (2005), onde são implementados Histogramas de Gradientes Orientados (HOG) para detecção da face humana com um desempenho superior aos modelos correlatos.

De acordo com o trabalho em Johnston e Chazal (2018), apresentado no Mapeamento Sistemático em Rissato, Bulcao-Neto e Macedo (2021), a dlib, especificadamente na função get_frontal_face_detector(), supera o desempenho dos detectores da face humana implementados do framework OpenCV[2] utilizando a técnica expressada em Viola e Jones (2001).

[1] A versão 3.7.3 da linguagem Python foi utilizada.
[2] Disponível em https://opencv.org/.

Figura 17 – Processo de indução de algoritmo / classificação por uma rede neural



Fonte: Autoria própria

Foi utilizada a função load_rgb_image() que recebe como parâmetro o caminho da imagem a ser analisada e retorna uma matriz tridimensional com os valores RGB (Red, Green, Blue) de cada pixel da imagem, considerando que imagens coloridas foram utilizadas.[3]. Não houve qualquer tratamento da imagem advinda, a imagem crua (ou *raw*) é utilizada durante todo o processo.

Ao ser localizada a face humana na função get_frontal_face_detector(), essa é encaminhada para a próxima etapa (B) para localização dos *landmarks*. Caso a face não seja localizada, a imagem é descartada e a execução continua para a próxima imagem, não havendo mais imagens o fluxo é encerrado sem a extração de características.

O objeto da face localizado na etapa A é encaminhado à etapa B para detecção dos *landmarks* que também utilizam a mesma implementação e versão da dlib, agora instanciado por meio de shape_predictor() que recebe como parâmetro de instanciamento o modelo de predição dos *landmarks*.

O modelo para detecção de *landmarks* foi desenvolvido utilizando o trabalho em Kazemi e Sullivan (2014) para estimar a posição da cabeça, associado ao conjunto de

[3] Em geral, o mesmo tipo de retorno é aplicado entre bibliotecas ou frameworks diferentes, inclusive na obtenção da imagem por meio de sequência, como em vídeos. O denominador em comum é o retorno da matriz de dados dos pixels da imagem. Caso a imagem analisada, fosse alterada, apenas para retornar em escala de cinza, é esperado um vetor contendo um valor da intensidade de cor entre o preto absoluto (0) e o branco absoluto (255) para cada pixel.

dados elaborado em Sagonas, Antonakos, Tzimiropoulos, Zafeiriou e Pantic (2016), com o intuito de extrair 68 *landmarks* da face humana.

O preditor dos *landmarks* é carregado com a imagem original e com o objeto da face retornado à etapa A. Caso os *landmarks* não sejam reconhecidos, a execução continua para a próxima imagem, caso não existam mais imagens, a execução é interrompida. Havendo *landmarks*, o objeto retornado é convertido num vetor de tuplas, com cada tupla contendo as coordenadas x e y de cada um dos 68 *landmarks* reconhecidos. Ao final, o vetor com as coordenadas é encaminhado à etapa C.

Ao iniciar a etapa C, é realizado o passo essencial da contribuição deste trabalho, onde o vetor de coordenadas advindo da etapa B é filtrado para extrair apenas as 20 coordenadas dos *landmarks* da boca. Uma análise combinatória entre as coordenadas de cada ponto é elaborada e posteriormente encaminhadas a um laço de repetição para calcular a distância Euclidiana entre cada par de pontos, resultando num vetor de 190 distâncias. Em seguida, aplica-se a normalização $l_1 - norm$ onde a soma dos valores absolutos é utilizada para dividir todos os valores únicos.

No processo de aplicação do modelo, ao final da etapa C, o vetor de distâncias é enviado ao modelo induzido/treinado para a predição da classe, dentro do mesmo laço de repetição descrito na etapa C. Contudo, no processo de indução, os dados normalizados são armazenados externamente para serem utilizados posteriormente para inferência dos modelos.

A etapa D consiste em realizar o processo de indução dos algoritmos e treino da rede neural para geração dos respectivos modelos. Os algoritmos de Árvore de Decisão, k-NN, Random Forest e Support Vector Machine foram utilizados. O framework escolhido para o processo de indução foi o *Waikato Environment for Knowledge Analysis* (WEKA) por ser de código aberto na linguagem Java e os algoritmos para o processo de indução e predição poderem ser portados para dispositivos móveis na plataforma Android.

Na mesma etapa D, foi realizado o treinamento de rede neural do tipo Multi-Layer descrita na Seção 5.1.1. Para a construção da rede neural foi utilizada a biblioteca *scikit-learn* da linguagem Python na versão 0.23.2, por meio do instanciamento da classe *MLPClassifier*. O conjunto de treinamento baseado na Seção 4.4 foi utilizado como entrada de dados para a construção do classificador da rede neural e dos modelos induzidos com o WEKA.

Os modelos gerados na etapa D foram validados na etapa E utilizando o conjunto de testes descrito na Seção 4.6. Os critérios utilizados para análise da validação podem ser observados na Seção 5.2.

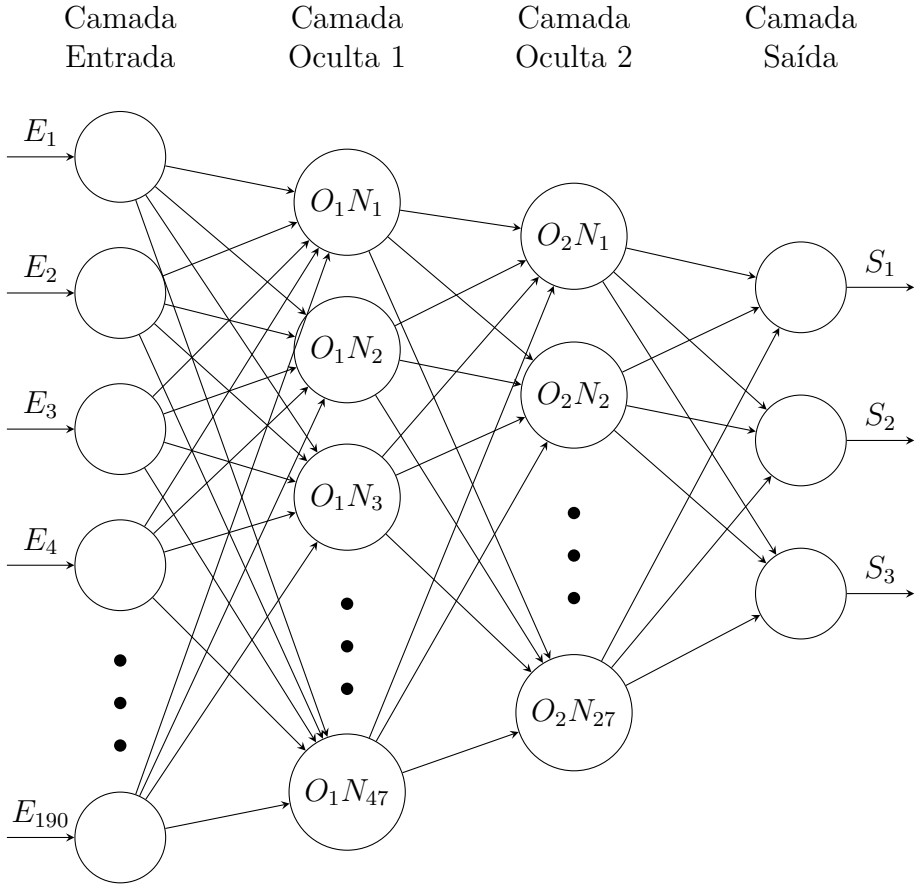
O método RVPNV foi construído de modo a ser replicado em dispositivos móveis que em geral possuem um poder de processamento muito mais reduzido, se comparados a

computadores pessoais ou servidores. Respeitando-se a essência de cada etapa, como na etapa A, a localização da face, na etapa B a extração de 68 *landmakrs* e na etapa C, a extração do vetor de características, com o cálculo da distância e normalização baseada em $l_1 - norm$, basicamente qualquer framework e técnica podem ser empregados, tornando o método altamente extensível ao maior número possível de tipos de dispositivos móveis.

5.1.1 Rede Neural - Multilayer Perceptron

O algoritmo de rede neural Multilayer Perceptron (GARDNER; DORLING, 1998) foi utilizado para criar um classificador para identificar os movimentos de interesse baseados no conjunto criado por meio de 1. Essa rede neural foi moldada conforme descrito na Figura 18.

Figura 18 – Arquitetura da rede neural do tipo Multilayer Perceptron utilizada para classificação dos movimentos de interesse



Fonte: Autoria própria

A rede neural disposta na Figura 18 é composta por quatro camadas no total, sendo a primeira camada, denominada de Camada de Entrada constituída por 190 neurônios, a mesma quantidade de atributos dos conjuntos de treinamento e teste, sem redução de

dimensionalidade. Esta camada inicial recebe cada um dos atributos separadamente e os encaminha para a primeira camada oculta.

As próximas duas camadas são chamadas de Camadas Ocultas e são compostas por 47 e 27 neurônios respectivamente. As Camadas Ocultas possuem os neurônios responsáveis por realizar o cálculo do peso sináptico, da função não linear de ativação e do valor estimado do vetor de gradiente para retropropagação na rede (HAYKIN, 2007).

Por fim, a última camada é composta por três neurônios de saída, um para cada classe de interesse, no caso, beijo, estalo de língua e sopro, responsáveis por expor os resultados obtidos da classificação, juntamente com as probabilidades de cada classe. Esta camada também é responsável por iniciar o processo de retropropagação dos erros para os neurônios da camada oculta dois até a camada oculta um (HAYKIN, 2007).

A função de ativação utilizada nas Camadas Ocultas um e dois foi a *Rectified Linear Unit* (ReLU) ou Unidade Linear Retificada (NAIR; HINTON, 2010) e a função otimizadora dos pesos foi a *Adaptive Momentum Estimation* (ADAM) (ZHANG, 2018).

5.2 Métricas de Avaliação do Classificador

Para avaliar o classificador construído baseando-se em algoritmos de Aprendizado de Máquina ou Rede Neural, as métricas de Acurácia, Precisão, Sensibilidade, Medida F1 e *Log Loss* foram utilizadas, bem como a Matriz de Confusão para visualização dos dados que geram tais métricas.

Acurácia consiste em analisar quantos exemplos do conjunto de testes foram classificados corretamente. A Precisão avalia dos exemplos classificados como uma determinada classe, quantos realmente eram da respectiva classe em destaque, sem considerar nenhum exemplo de falsos negativos. No caso da Sensibilidade, todos os exemplos de uma determinada classe, ainda que falsos negativos, são considerados para avaliação. A métrica *Log Loss* é aplicada para punir mais severamente cada predição erroneamente realizada e possui a propriedade de ser uma métrica capaz de avaliar diferentes algorítimos sob um mesmo domínio. Essas métricas de avaliação do classificador são detalhadas na próximas seções, subdivididas em Matriz de Confusão (5.2.1), Acurácia (5.2.2), Precisão (5.2.3), Sensibilidade (5.2.4), Medida F1 (5.2.5) e *Log Loss* (5.2.6).

5.2.1 Matriz de Confusão

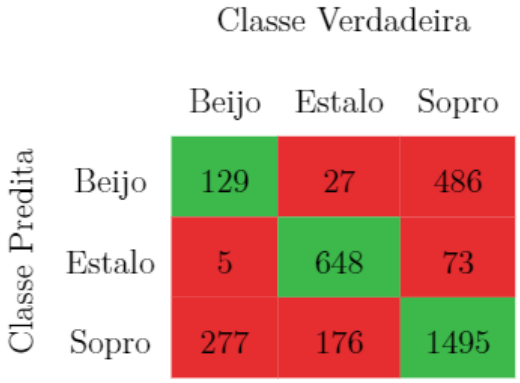
Utilizou-se uma matriz de confusão para visualizar a performance do classificador criado, a partir do método proposto nesta dissertação. As predições corretas e incorretas realizadas

são tabuladas em forma de uma matriz $n \times n$, cujo n é o número de classes capaz de serem preditas. A diagonal principal demonstra os acertos do classificador, enquanto as demais intersecções apresentam os erros da predição (GAMA, 2011).

A aplicação dessa matriz de confusão é substancial para ilustrar o cálculo acerca da acurácia e facilitar o entendimento sobre o desempenho de determinadas classes, especificadamente. Para realizar essa avaliação, é necessário entender quais exemplos do conjunto de dados foram preditos corretamente e erroneamente.

Essa pesquisa possui três classes, beijo, estalo de língua e sopro, tratando-se de um problema de classificação de várias classes. Neste caso, a matriz de confusão e a forma de cálculo da classes positivas e negativas é diferenciada, como pode ser observado na Figura 19.

Figura 19 – Matriz de confusão com dados simulados para três ou mais classes



Fonte: Autoria própria

Para interpretar uma matriz de confusão, os seguintes conceitos são relevantes:

- Verdadeiro Positivo (VP), que se refere ao número de predições em que o classificador previu corretamente a classe positiva como sendo positiva.
- Verdadeiro Negativo (VN), que se refere ao número de predições em que o classificador previu corretamente a classe negativa como sendo negativa.
- Falso Positivo (FP), que se refere ao número de predições em que o classificador previu incorretamente a classe negativa como sendo positiva.
- Falso Negativo (FN), que se refere ao número de predições em que o classificador previu incorretamente a classe positiva como sendo negativa.

Baseando-se na Figura 19, e considerando os conceitos acima apresentados, pode-se calcular[4]:

- Para a classe beijo tem-se:
  VP = 129
  FP = (27 + 486) = 513
  VN = (648 + 1495 + 176 + 73) = 2392
  FN = (5 + 277) = 282

- Para a classe estalo de língua:
  VP = 648
  FP = (5 + 73) = 78
  VN = (1495 + 277 + 129 + 486) = 2387
  FN = (27 + 176) = 203

- Para a classe sopro:
  VP = 1495
  FP = (277 + 176) = 453
  VN = (129 + 648 + 27 + 5) = 809
  FN = (486 + 73) = 559

5.2.2 Acurácia

A Equação 5.1 considerando a acurácia [5] demonstra o cálculo de quantos exemplos foram preditos corretamente dentre todos os cenários possíveis.

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{FP} + \text{VN} + \text{FN}} \tag{5.1}$$

Considerando o modelo melhor avaliado (Tabela 11) referente às classes de beijo e estalo, tem-se a Equação 5.2 aplicada:

$$\text{Acurácia beijo-estalo} = \frac{559 + 717}{559 + 717 + 9 + 83} = \frac{1276}{1368} = 0,9327 \tag{5.2}$$

[4] Nessa exemplificação de uma matriz de confusão para três classes distintas, foi utilizado o modelo baseado em Random Forest para três classes induzido com dados de treinamento (Seção 6.2), por ter obtido o melhor desempenho dentre outros modelos para predizer a classificação entre as três classes de interesse. Não foi necessariamente considerado o melhor modelo preditivo ao final, pois um cenário com duas classes foi elaborado para produzir outros modelos. A elaboração dos cenários e esses modelos para duas classes distintas são melhor elaborados no Capítulo 6)

[5] A acurácia é a medida padrão para avaliar a performance de um classificador.

5.2.3 Precisão

Em termos de precisão[6], pode-se responder a seguinte pergunta: Dos exemplos preditos como sendo da classe X, quantos desses realmente eram da classe X? A Equação 5.3 demonstra o cálculo da precisão.

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falso Positivos}} \tag{5.3}$$

Aplicando a equação 5.3 aos resultados do modelo generalizado capaz de classificar entre beijo e estalo disposto na Tabela 11, a precisão para a classe beijo seria:

$$\begin{aligned} \text{Precisão}_{beijo} &= \frac{\text{beijos classificados certos}}{\text{beijos classificados certos} + \text{estalos como beijos}} \\ \text{Precisão}_{beijo} &= \frac{559}{559+9} = \frac{559}{568} = 0,9841 \end{aligned} \tag{5.4}$$

A Equação 5.4 pode ser lida da seguinte maneira: De todos os exemplos classificados como beijo, quantos realmente eram do movimento de beijo e quantos eram de outro movimento?

Tem-se, ao aplicar a fórmula, que 559 eram beijos e realmente foram classificados como beijo e nove eram do movimento de estalo, mas foram classificados como beijo. Logo, a precisão desse modelo para o movimento de beijo, foi de um em 559 dividido por 568 (559 + 9) ou seja 98,41%.

Já para o movimento de estalo, tem-se a seguinte Equação 5.5 aplicada:

$$\begin{aligned} \text{Precisão}_{estalo} &= \frac{\text{estalos classificados certos}}{\text{estalos classificados certos} + \text{beijos como estalos}} \\ \text{Precisão}_{estalo} &= \frac{717}{717+83} = \frac{717}{800} = 0,8962 \end{aligned} \tag{5.5}$$

A Precisão para classificar movimentos de estalo corretamente foi de 89,62%. No fim, a precisão geral do modelo é a média aritmética de todas as precisões, neste caso, a precisão total é de 94,01%.

[6] Precisão é a fração dos exemplos classificados como positivos de todos possíveis positivos

5.2.4 Sensibilidade

A Equação 5.6 ilustra o cálculo da sensibilidade[7]:

$$\text{Sensibilidade} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falso Negativos}} \tag{5.6}$$

Para descobrir qual a sensibilidade para a classe beijo, por exemplo, basta aplicar a Equação 5.6 aos dados da matriz simulada da Tabela 19. Pode-se visualizar a fórmula aplicada na Equação 5.7.

$$\begin{aligned} \text{Sensibilidade}_{beijo} &= \frac{\text{beijos classificados certos}}{\text{beijos classificados certos} + \text{beijos como estalos}} \\ \text{Sensibilidade}_{beijo} &= \frac{559}{559+83} = \frac{559}{642} = 0,8707 \end{aligned} \tag{5.7}$$

Ao mesmo tempo, considerando a Equação 5.3 da Precisão, tem-se justamente o oposto, haveria uma diminuição da precisão, pois estalos de língua seriam classificados como beijo. No presente cenário, temos que a Sensibilidade aplicada do movimento de beijo foi de 87,07%. A sensibilidade do movimento de estalo é calculada na Equação 5.8.

$$\begin{aligned} \text{Sensibilidade}_{estalo} &= \frac{\text{estalos classificados certos}}{\text{estalos classificados certos} + \text{estalos como beijos}} \\ \text{Sensibilidade}_{estalo} &= \frac{717}{717+9} = \frac{717}{726} = 0,9876 \end{aligned} \tag{5.8}$$

Logo, a sensibilidade do movimento de estalo é de 98,76%. A sensibilidade final do modelo apresentado é a média ponderada de todas as sensibilidades calculadas, do modelo em questão é de 92,91%.

Contudo, se todo conjunto de dados fosse rotulado como uma determinada classe, a sensibilidade calculada seria de 1,0, ou seja, perfeita. Isso porque nenhum dos Falsos Negativos seria classificado como sendo da classe oposta. Para tentar resolver essa problemática, a métrica denominada de medida F1 também foi utilizada.

[7] Sensibilidade ou Revocação é a proporção dos exemplos que foram classificados como positivos dentre todos que realmente eram positivos no conjunto de dados. Por esse motivo também é conhecida como Taxa de Verdadeiro Positivo.

5.2.5 Medida F1

Observa-se a fórmula para cálculo da Medida F1[8]:

$$F_1 = 2 \times \frac{\text{precisão} \times \text{sensibilidade}}{\text{precisão} + \text{sensibilidade}} \tag{5.9}$$

Considerando o modelo disposto na Tabela 11 referente ao modelo que prediz novos exemplos das classes beijo e estalo, temos na Equação 5.10 o seguinte resultado para a Media F1:

$$F_1 = 2 \times \frac{94,01 \times 92,91}{94,01 + 92,91} = 2 \times \frac{8734,46}{186,92} = 2 \times 46,728 = 93,45\% \tag{5.10}$$

A Medida F1 baliza entre as métricas de Precisão e Sensibilidade para que não haja sobreposição de qualquer uma delas.

5.2.6 *Log Loss*

Log Loss ou *Cross-entropy loss* é uma função que calcula a performance de um classificador, independente de qual seja. A cada predição realizada corretamente o valor de *Log Loss* diminui tendendo a 0; e a cada predição errônea o valor aumenta, tendendo ao infinito.

Na Equação 5.11, a função *Log Loss* é calculada como sendo o logaritmo negativo da probabilidade estimada pelo classificador:

$$L_{\log}(y, p) = -(y \times \log(p) + (1 - y) \times \log(1 - p)) \tag{5.11}$$

cujo y é um binário indicando se a classe sendo avaliada é a real e p é a probabilidade calculada para a predição em análise.

Para exemplificar, imagine um classificador induzido para avaliar entre duas classes, cão e gato. Após a indução, é apresentado um novo exemplo para análise, em fase de validação, esse novo exemplo é da classe gato, esse classificador hipotético classificou com os seguintes resultados: classe cão com 13% de probabilidade e classe gato com 87% de probabilidade, logo, o classificador hipotético conclui que gato seria a classe correta. Na Equação 5.12, o valor de *Log Loss* apresentado é:

$$L_{\log}(1, 0, 87) = -(1 \times \log(0,87) + (1-1) \times \log(1-0,87)) = -(-0,139 + 0 * -0,13) = 0,139 \tag{5.12}$$

[8] Também conhecida como *F-Measure* ou *F-Score* é uma média harmônica entre precisão e sensibilidade para balanceá-las, punindo quaisquer valores extremos.

Logo, considerando que o classificador hipotético classificou o novo exemplo como gato, sendo que este exemplo era realmente da classe gato, com uma probabilidade de 87%, o valor de *Log Loss* seria de 0,139, bem próximo a 0, ou seja, é um classificador com uma performance considerável.

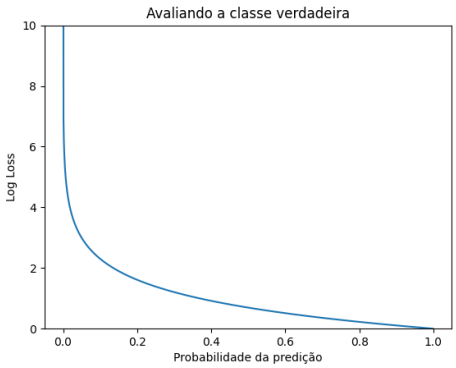
Uma das vantagens da métrica de *Log Loss* é que exemplos que são classificados erroneamente e com uma probabilidade alta são mais penalizados. Considerando o mesmo exemplo anterior, entretanto, o classificador hipotético classificou a classe cão (classe errada) com 87% de probabilidade de ser a real. Logo, o valor de *Log Loss* na Equação 5.13 é:

$$L_{\log}(0,0,87) = -(0 \times \log(0,87) + (1-0) \times \log(1-0,87)) = -(0+1*-2,04) = 2,04 \quad (5.13)$$

Assim sendo, o classificador hipotético que classificou erroneamente a classe real gato como sendo cão, e, com uma probabilidade de 87% obteria um *Log Loss* de 2,04, muito superior a zero e portanto, pode-se considerar o classificador hipotético com uma performance péssima.

Na Figura 20 é possível verificar verificar como ocorre a distribuição da função *Log Loss* em relação às probabilidades atribuídas à classe verdadeira.

Figura 20 – Distribuição da função *Log Loss* quando avaliando uma classe considerada verdadeira



Fonte: Autoria própria

Observando os dados apresentados na Figura 20, nota-se que ao diminuir a probabilidade da predição em relação à classe verdadeira ocorre um considerável aumento na penalização ao classificador. Este efeito é o mesmo para a classe incorreta: à medida que a probabilidade aumenta, a função penaliza o classificador[9] da mesma maneira.

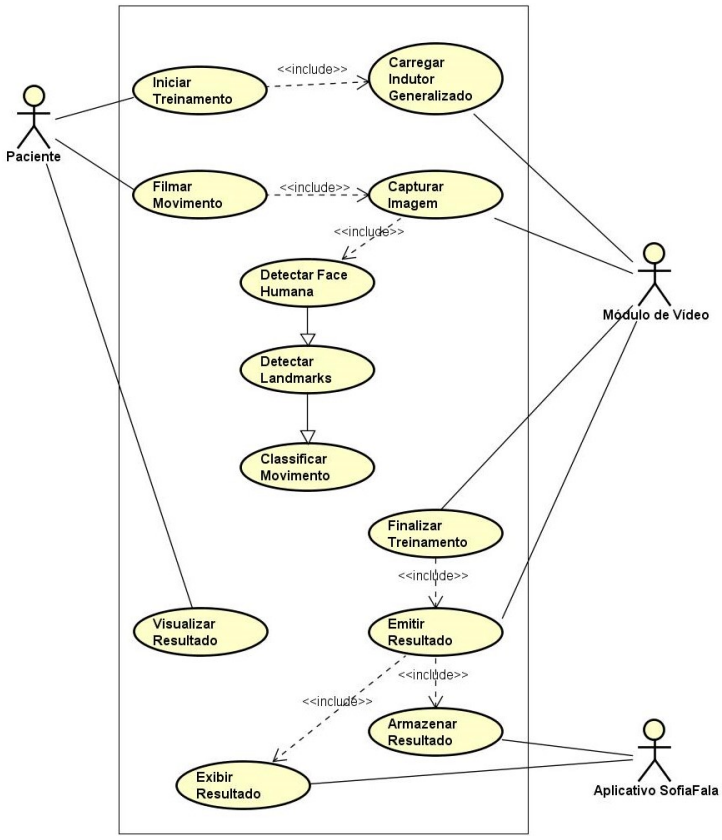
[9] Existe apenas uma problemática quando a probabilidade da predição da classe incorreta é de 100%. Neste caso, o *Log Loss* seria $-log(1-1)$, ou, $-log(0)$, como esse cálculo não é possível, algumas

Portanto, a função *Log Loss* foi utilizada nesta pesquisa para avaliar modelos induzidos, fossem gerados por algoritmos diferentes ou redes neurais. A função *Log Loss* foi capaz de unificar essa comparação, fornecendo uma forma de avaliação de todas as predições realizadas.

5.3 Caso de Uso do Método RVPNV

Para melhor elucidação da aplicação do método apresentado, esta seção apresenta um modelo, representado como Caso de Uso, que ilustra como diferentes usuários podem interagir com um sistema que implemente o método RVNPV, como por exemplo, o "SofiaFala". Nesse sentido, a Figura 21 apresenta um diagrama de proposta para caso de uso, com o intuito de ilustrar a aplicação do modelo induzido gerado, antevendo, assim, sua aplicabilidade em um aplicativo existente para treinamento de fala, tal como o "SofiaFala".

Figura 21 – Classificação do movimento não articulatório



Fonte: Autoria própria

implementações lidam com essa impossibilidade transformando a probabilidade em 0,9999 como no caso da função *log_loss* da biblioteca *scikit learn* do Python. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html.

No diagrama, o paciente utiliza o aplicativo para realizar o treinamento fonoaudiológico. Ao iniciar o treinamento, o programa localiza e carrega o modelo indutor criado por esta pesquisa, o qual realizará a predição dos movimentos de interesse. Posteriormente, os movimentos realizados pelo paciente são filmados. Essas imagens são analisadas para detectar a face humana, aplicar os *landmarks* e classificar o movimento realizado, por meio do indutor previamente carregado. Detectado o movimento de interesse pelo indutor, com a acurácia desejada previamente definida ou finalizado o tempo para a realização do movimento, o treino é encerrado e um resultado é emitido. Esse resultado é apresentado ao paciente e armazenado internamente.

A partir da experiência obtida com o presente trabalho, nosso grupo de estudos iniciou a aplicação dos modelos gerados na extração de métricas das predições, determinando maiores dados para posterior análise de comportamento do modelo, bem como o acompanhamento evolutivo do paciente.

5.4 Considerações Finais

Este capítulo destinou-se à descrição do método RVPNV para reconhecimento de padrões em imagens advindas de movimentos da boca humana de praxias não verbais, como beijo, estalo de língua e sopro. Discorreu-se acerca do algoritmo que descreve o processo de indução de um classificador, desde a utilização do conjunto de dados até sua efetiva implementação para a geração de um modelo concreto.

Também foi apresentada, à Figura 18, a arquitetura da RN MLP utilizada nesta pesquisa. Concomitantemente, foram descritas as métricas de Acurácia, Precisão, Sensibilidade, Medida F1 e *Log Loss*, para avaliação dos classificadores produzidos dentre os quatro cenários estabelecidos (melhor descritos no próximo capítulo. Por fim, uma proposta de caso de uso do método RVPNV foi apresentada para basear a futura integração do melhor classificador obtido, junto ao aplicativo desenvolvido pelo grupo de trabalho "SofiaFala".

6

Resultados e Discussão

Este capítulo reuni os resultados obtidos com a pesquisa apresentada nesta dissertação, usando o método apresentado (Capítulo 5) nos conjuntos de dados inicial, de treinamento e de teste (Capítulo 4) e a discussão dos mesmos com a literatura correlata (Capítulo 2). Por meio da metodologia descrita (Capítulo 5), foram construídos cinco cenários, sendo cada um desses uma evolução do cenário anterior. Os cenários foram desenvolvidos com o objetivo de demonstrar as etapas da aplicação da metodologia para a construção de um modelo classificador, as dificuldades apresentadas, as hipóteses elaboradas e a investigação e análise dessas hipóteses até a obtenção dos resultados finais.

As métricas utilizadas para avaliar a classificação dos modelos induzidos foram apresentadas na Seção 5.2.

No primeiro cenário apresentado na Seção 6.1, realizou-se a indução de algoritmos frequentemente utilizados no campo de Aprendizado de Máquina para construir modelos induzidos com o conjunto de dados iniciais (Seção 4.2) e validados no conjunto de teste (Seção 4.6). Neste cenário, a indução pretendeu distinguir entre as três classes de interesse, beijo, estalo de língua e sopro, e foi apresentada a problemática relacionada à quantidade dos exemplos do conjunto de treinamento.

Já no segundo cenário, apresentado na Seção 6.2, os mesmos algoritmos apresentados na Seção 6.1 foram induzidos para classificar entre as três classes distintas, mas com um novo conjunto de dados, denominado de conjunto de dados de treinamento (Seção 4.4), contendo ao todo 12316 exemplos. Neste cenário, discorreu-se sobre duas problemáticas, a de um possível desbalanceamento entre classes e isso ter ocasionado a não distinção entre as classes de beijo e sopro. A validação dessa hipótese foi apresentada no próprio cenário. Já a segunda hipótese foi proposta em relação à problemática da "maldição da dimensionalidade" (Seção 3.3.3), a qual foi discutida no cenário proposto na Seção 6.3.

Em seguida foi composto o terceiro cenário, expressado na Seção 6.3 que empregou uma Rede Neural do tipo Multilayer Perceptron para abordar a problemática da alta dimensionalidade dos dados, com o intuito de validar se a alta dimensionalidade foi a cau-

sadora do desempenho não aceitável no segundo cenário. As métricas de análise de dados foram aplicadas para comparar os modelos, principalmente a métrica *Log Loss*, específica para comparar modelos induzidos com diferentes algoritmos. Nesse modelo induzido foi aplicado o balanceamento entre as classes aplicando a técnica de sobre-amostragem (Seção 3.5) para equiparar as classes com menos exemplos à majoritária, considerando o desempenho obtido no cenário da Seção 6.2. Nesse cenário, também foi apresentada uma hipótese para a não separação entre os movimentos de beijo e sopro, e, com uma análise mais aprofundada sobre a distribuição dos dados extraídos das amostras foi possível observar o motivo latente da não separação. Para validar a observação apresentada, um quarto cenário foi elaborado.

No quarto cenário, para dirimir dúvidas relativas às propostas dos cenários dois e três, quanto à alta dimensionalidade, a não separação entre as classes de beijo e sopro, o balanceamento de classes, foram realizadas induções com os mesmos algoritmos utilizados nos cenários dois e três, mas com a indução entre duas classes distintas assim separadas entre os pares, beijo e estalo, estalo e sopro e por fim beijo e sopro. Um comparativo geral com os demais cenários foi elaborado para basear a conclusão.

6.1 Cenário de Classificação AM com Dados Iniciais

Nesta seção, o cenário de experimentação utilizou algoritmos de AM treinados com três classes distintas criadas a partir dos dados de amostragem iniciais apresentados na Seção (4.2). A Tabela 6 demonstra o resultado da predição dos classificadores induzidos com os algoritmos algoritmos Árvore de Decisão (J48), k-NN (IBk), Random Forest e Support Vector Machine (Sequential Minimal Optimization) aplicados no conjunto de teste contendo 3316 exemplos não vistos na fase de treinamento.

Todo o treinamento foi realizado, utilizando o programa Weka, que possui diversos algoritmos de Aprendizado de Máquina transcritos em Java[1]. Os algoritmos foram utilizados em suas implementações padrões, exceto o algoritmo k-vizinhos mais próximos cujo valor de k foi alterado para 7. O conjunto de dados para treinamento e o de teste foram normalizados (Seção 3.6).

[1] Esses algoritmos, por estarem escritos em Java, podem ser mais facilmente portados para o sistema operacional Android.

6.1.1 Resultados

Os algoritmos foram utilizados em suas implementações padrões, exceto o algoritmo *k*-vizinhos mais próximos onde o valor de *k* foi alterado para 7. Tanto o conjunto de dados para treinamento quanto o de teste foram normalizados (Seção 3.6).

Tabela 6 – Resultado das predições dos classificadores Árvore de Decisão na implementação J48, k-NN na implementação iBK, Random Forest e Support Vector Machine na implementação Sequential Minimal Optimization (SMO). Os classificadores foram avaliados no conjunto de testes descrito na Seção 4.6. Os itens marcados em negrito são os melhores resultados dentre as métricas apresentadas

	Precisão			Sensibilidade			Medida F1			Acurácia	Log Loss
Classificadores	beijo	estalo	sopro	beijo	estalo	sopro	beijo	estalo	sopro		
Árvore de Decisão (J48)	0,16	0,99	0,55	0,39	**0,27**	0,46	0,23	**0,43**	0,50	40,32	11,55
k-NN (iBK)	**0,23**	**1**	0,59	0,77	0,09	0,33	**0,35**	0,17	0,42	36,34	0,93
Random Forest	0,20	**1**	**0,61**	0,52	0,18	**0,47**	0,29	0,30	**0,53**	**41,86**	**0,73**
SVM (SMO)	0,20	0,99	0,28	**1**	0,24	0	0,34	0,39	0	24,82	0,92

Fonte: Autoria própria

Considerando os dados acima obtidos no presente cenário, a análise dos resultados e discussão em comparação à literatura pertinente corroborou com a hipótese conclusiva desta dissertação (Capitulo 7) que estabelece a relação de similaridade entre os movimentos de beijo e sopro e direciona na produção de modelos separados entre duas classes distintas. O apontamento discutido dos resultados encontra-se a seguir.

6.1.2 Discussão

Ao analisar os resultados apresentados na Tabela 6, foi possível observar que nenhum dos classificadores induzidos foi capaz de predizer novos exemplos com uma acurácia superior a 50%. Entretanto, na fase de treinamento, os algoritmos Árvore de Decisão, Random Forest e *SVM* apresentaram uma acurácia final superior a 70%, já o algoritmo k-NN apresentou uma acurácia superior a 63%. Sob a ótica das métricas de acurácia e *Log Loss*, concluiu-se que o modelo induzido com o algoritmo Random Forest apresentou o melhor resultado, seguido pelo modelo induzido com o algoritmo k-NN, priorizando-se a Acurácia. Os modelos induzidos com os algoritmos SVM e Árvore de Decisão resultaram em terceiro e quarto lugares, respectivamente.

Observando somente acurácia, o modelo do algoritmo SVM apresentou o pior resultado, e, considerando apenas *Log Loss*, o modelo induzido com Árvore de Decisão foi o que apresentou o pior desempenho.

Os dados do conjunto inicial estavam dispostos em 23 imagens do movimento sopro, 39 imagens do movimento beijo e 20 imagens do movimento estalo de língua, não sendo considerados na literatura como desbalanceados, uma vez que o desbalanceamento é caracterizado pela razão de 1:4 ao menos entre as classes minoritária e majoritária conforme apresentado em Krawczyk (2016). No presente cenário, sendo as classes minoritária e majoritária respectivamente as classes estalo de língua (20 imagens) e beijo (39 imagens), essa distribuição não chega à razão de 1:2. Portanto, não foi necessária a aplicação de métodos para correção de desbalanceamento entre classes, como apresentados na Seção 3.4. Os atributos do conjunto de dados contendo 82 imagens foram normalizados para valores do intervalo entre zero e um com o intuito de remover a influência entre os atributos, conforme discutido na Seção 3.6.

A validação desses modelos induzidos foi realizada inicialmente de forma prática, com a análise pontual dos movimentos de interesse de rostos de indivíduos que não participaram da fase de treinamento daqueles modelos, já que o conjunto de teste (Seção 4.6) não existia na época da indução dos classificadores apresentados. Verificou-se que esses modelos não eram generalizados o suficiente para diferenciar entre os movimentos de interesse, por vezes, apresentado certa aleatoriedade em identificar uma mesma imagem estática.

Este cenário é descrito na literatura como *overfitting* (GAMA, 2011), quando o modelo induzido está superajustado aos dados de treinamento e não é capaz de predizer novos exemplos. Foi descartada a hipótese do cenário ter sido causado por *underfitting*, já que nesse caso, o algoritmo induzido não consegue reconhecer padrões, e, portanto, apresenta uma acurácia baixa mesmo na fase de treinamento[2], e consequentemente na fase de testes.

Não houve necessariamente uma surpresa quanto à conclusão do resultado acerca de *overfitting*, já que o conjunto de dados iniciais com 82 imagens no total, continha poucas imagens, comparado empiricamente, em relação a outros conjuntos de dados de outros domínios (ler sobre o Mapeamento Sistemático na Seção 2). Contudo, o cenário de *overfitting* já era promissor, pois significava que os algoritmos, no geral, tinham conseguido aprender algum padrão, em relação ao cenário de *underfitting* onde nenhum padrão encontrado é assimilado.

Considerando a ocorrência de *overfitting* descrita, a próxima etapa foi construir um novo conjunto de dados, denominado de conjunto de dados de treinamento e outro denominado de conjunto de dados de teste, capazes de expressar com maior fidelidade os movimentos de beijo, estalo de língua e sopro. A metodologia para confecção desses conjuntos de dados pode ser observada nas Seções 4.4 e 4.6. Após a construção desses conjuntos, um novo cenário foi explorado na Seção 6.2 para induzir modelos com os dados

[2] Os dados relativos à fase de treinamento podem ser analisados no Anexo D.

do conjunto de treinamento e validar as predições das classificações para os exemplos do conjunto de dados de teste.

Por fim, um segundo cenário foi elaborado para realizar uma nova indução, dos mesmos algoritmos descritos nesta Seção, mas com um novo conjunto de dados para treinamento, denominado de conjunto de treinamento e ou de dados para teste, denominado de conjunto de teste para avaliar a capacidade desses novos modelos em predizer dados inéditos.

6.2 Cenário de Classificação AM com Dados de Treinamento

Nesta seção, o cenário de experimentação utilizou algoritmos de AM treinado com três classes distintas criadas a partir dos dados de treinamento apresentados na Seção 4.4. A Tabela 7 demonstra os classificadores de Árvore de Decisão, *n*-vizinhos mais próximos [3], Random Forest e Support Vector Machine, induzidos utilizando o conjunto de treinamento com 12316 exemplos e validados no conjunto de teste com 3316 exemplos [4].

6.2.1 Resultados

A indução dos algoritmos de AM e a normalização dos conjuntos de dados foram realizadas conforme descritas na Seção 6.1. Os resultados obtidos são apresentados na tabela a seguir:

Tabela 7 – Resultado das predições dos classificadores de Árvore de Decisão na implementação J48, k-NN na implementação iBK, Random Forest e Support Vector Machine na implementação Sequential Minimal Optimization (SMO). Os classificadores foram avaliados no conjunto de testes apresentados na Seção 4.6. Os itens marcados em negrito são os melhores resultados dentre as métricas apresentadas

	Precisão			Sensibilidade			Medida F1			Acurácia	Log Loss
Classificadores	beijo	estalo	sopro	beijo	estalo	sopro	beijo	estalo	sopro		
Árvore de Decisão (J48)	0,26	0,62	**0,73**	0,36	0,72	0,59	0,30	0,67	0,66	57,69	11,22
k-NN (iBK)	0,24	0,78	0,69	0,27	0,79	0,66	0,26	0,79	0,68	61,61	4,18
Random Forest	**0,31**	0,76	**0,73**	0,20	**0,89**	**0,77**	0,24	0,82	**0,75**	**68,52**	**0,56**
SVM (SMO)	0,27	**0,88**	0,72	**0,37**	0,81	0,66	**0,31**	**0,84**	0,69	63,57	0,66

Fonte: Autoria própria

[3] Induzido considerando os sete vizinhos mais próximos.

[4] Os dados originais relativos à fase de treinamento e teste podem ser analisados no Anexo E.

6.2.2 Discussão

Ao proceder com a análise dos resultados sintetizados na Tabela 7, a primeira discussão relevante, quanto à heurística da acurácia, pauta-se na exceção da Árvore de Decisão que obteve uma acurácia de 57,69%. Os demais algoritmos, por sua vez, alcançaram uma acurácia geral entre 60% e 70%, entretanto, ao avaliar conjuntamente com as demais métricas como Precisão, Sensibilidade e Medida F1, verifica-se que o resultado da predição não pôde ser considerado, ao final da análise, nem ao menos adequado.

Essa conclusão decorre do fato que em todos os modelos, a classe beijo não obteve mais que 31,2% de precisão, 36,6% de sensibilidade e 31,1% de medida F1, no melhor modelo avaliado com as predições realizadas no conjunto de teste (Seção 4.6). À primeira análise, essa performance insatisfatória em relação às predições da classe beijo foi atribuída ao desbalanceamento de classes naturalmente ocorrido no conjunto de dados de treinamento, já que dos 12316 exemplos do conjunto de dados de treinamento, apenas 1850 são da classe beijo, ou 6,65% do total do conjunto, tornando-a classe minoritária. Já a classe sopro é considerada a classe majoritária contendo 7065 exemplos, seguida pela classe estalo com um total de 3401 exemplos.

Segundo Krawczyk (2016), o desbalanceamento não estaria caracterizado já que não atingiu a razão de 1:4, contudo, considerando que a razão entre a classe sopro e beijo chega bem próxima a 1:4 (3,81), optou-se por realizar uma nova predição dos mesmos algoritmos induzidos na Seção 6.2.

O objetivo dessa nova predição é verificar se a diferença na quantidade de exemplos entre as classes, era bastante a ponto de influenciar significativamente a precisão, sensibilidade e medida F1 da classe beijo. Para tanto, utilizamos a técnica de subamostragem dos dados (Seção 3.4), cujos exemplos das classes de estalo e sopro foram aleatoriamente removidos até que ambas se igualassem à quantidade de 1850 da classe beijo.

A Tabela 8 apresenta os resultados dos modelos induzidos com os algoritmos Árvore de Decisão (J48), k-NN (iBK), Random Forest e SVM (SMO) contendo 1850 exemplos de cada classe dentre beijo, estalo de língua e sopro.

Comparando as Tabelas 7 e 8, em relação à **precisão** das predições do movimento de beijo, foco da hipótese apresentada acima, notou-se que houve uma perda de desempenho dos modelos induzidos com Árvore de Decisão e Random Forest, sendo essa perda de 0,03 e 0,04 respectivamente, em relação ao treinamento realizado com dados desbalanceados. No caso dos modelos induzidos com algoritmos k-NN e SVM houve um ganho da precisão de 0,01 em cada um desses modelos. De modo geral, houve uma perda média de 0,05 em relação aos modelos induzidos conforme apresentado na Tabela 7.

A métrica sensibilidade é estimada a partir da quantidade de predições assertivas

Tabela 8 – Resultados dos modelos induzidos com os algoritmos Árvore de Decisão (J48), k-NN (iBK), Random Forest e SVM (SMO) com dados de treinamento balanceados contendo 1850 exemplos de cada classe: beijo, estalo de língua e sopro. Os dados em negrito indicam o melhor desempenho dentro daquela métrica

	Precisão			Sensibilidade			Medida F1			Acurácia	Log Loss
Classificadores	beijo	estalo	sopro	beijo	estalo	sopro	beijo	estalo	sopro		
Árvore de Decisão (J48)	0,23	0,70	0,70	0,46	0,87	0,41	0,31	0,77	0,52	52,08	7,12
k-NN (iBK)	0,25	0,77	0,69	0,53	0,78	0,42	0,34	0,78	0,52	52,47	4,67
Random Forest	0,27	0,76	0,72	0,38	**0,92**	**0,56**	0,31	0,83	**0,63**	**60,28**	**0,62**
SVM (SMO)	**0,28**	**0,90**	**0,83**	**0,89**	0,80	0,26	**0,42**	**0,85**	0,39	50,03	0,75

Fonte: Autoria própria

dentre todas as possíveis. Nos resultados das predições apresentados na Tabela 8, houve um aumento de 27% em média na precisão, se comparados com os resultados apresentados na Tabela 7.

Considerando a medida F1, houve um aumento geral em todos os modelos induzidos com dados balanceados, em relação aos treinados com dados desbalanceados. Já que é uma métrica que faz a junção entre precisão e sensibilidade, a influência dessa última métrica acabou elevando em média em 24,3% da Medida F1.

Por fim, a métrica *Log Loss* que objetiva a comparação de algoritmos sob um mesmo domínio, considerando a confiabilidade de todas as predições realizadas, obteve, com exceção do algoritmo Árvore de Decisão, um decréscimo nos demais algoritmos de 11,75% em média. Já considerando somente o algoritmo Árvore de Decisão, houve uma melhora de 63% no desempenho geral do modelo em comparação com a versão induzida com dados desbalanceados.

Avaliando o cenário para validar a hipótese levantada de que o desbalanceamento de classes poderia ser o influenciador no desempenho negativo do movimento de beijo, concluiu-se que apesar da melhoria no desempenho da sensibilidade, e consequentemente da medida F1, o que é interessante, já que dentre os 1850 exemplos da classe beijo, 27% a mais de exemplos foram classificados corretamente em média, em relação às médias de acurácia houve um decréscimo em todos os algoritmos de 17% em média. Considerando a métrica *Log Loss* houve uma piora em todos os modelos, exceto no modelo induzido com o algoritmo de Árvore de Decisão. Mais latente é reparar que a melhoria na sensibilidade das classificações do movimento de beijo, foi obtida às custas da piora considerável do desempenho geral dos modelos em classificar o movimento de sopro. Com referência ao movimento de sopro houve uma piora de 62% da sensibilidade e de 35% da medida F1, obtendo uma melhora de apenas 2% relativa à precisão.

Constata-se que houve um ganho apreciável em relação às métricas do movimento de beijo, mas com esse ganho houve uma piora do desempenho das classificações relativas

do movimento de sopro, enquanto as classificações relativas ao movimento de estalo se mantiveram praticamente estáveis. De um modo geral, o fato de haver uma subamostragem do conjunto de dados causou uma piora no desempenho de todos os algoritmos, com exceção do modelo induzido com Árvore de Decisão, na perspectiva da métrica *Log Loss*. Houve uma dificuldade aparente em realizar a separação entre as classes de beijo e sopro, quaisquer que fosse o modelo utilizado, diferentemente da classe estalo, que foi separável com um desempenho relevante em todos os modelos.

Neste cenário, a problemática tomou um formato perturbante, já que agora o conjunto de dados de treinamento, mesmo na classe minoritária, possuía uma quantidade de exemplos equiparável à outros conjuntos de dados na literatura. Ainda que aplicando uma técnica de sobre-amostragem, e os três movimentos estarem separados entre si e compondo um conjunto de testes com indivíduos totalmente diferentes em comparação ao conjunto de treinamento, as métricas do melhor modelo ainda demonstravam desempenho aquém do desejado. Considerando esse contexto, surgiram duas hipóteses ainda mais ousadas e desafiadoras.

Em razão dos resultados apresentados, duas hipóteses foram relevantes para posterior aprofundamento. A primeira é se a classe beijo obtivesse uma quantidade maior de exemplos, preferencialmente igualando-se à quantidade de exemplos da classe sopro, os modelos seriam capazes de diferenciá-las entre si? A outra hipótese é o fato do conjunto de dados possuir uma alta dimensionalidade, com 190 dimensões, o que nos leva ao problema da "maldição da dimensionalidade" (DONOHO, 2000) (Seção 3.3.3) onde existem 2^{190} subconjuntos de dados, e, dentre esses, um subconjunto considerado ótimo que apresentaria o melhor resultado possível, motivo dos algoritmos escolhidos para classificar entre as classes de sopro e beijo não realizarem a classificação com um desempenho considerável.

Sendo assim, foi elaborado o próximo cenário, considerando uma rede neural do tipo Multilayer Perceptron para avaliar ambas as hipóteses apresentadas.

6.3 Cenário de Classificação RN com Dados de Treinamento

Nesta seção, o cenário de experimentação utilizou algoritmos de RN treinados com três classes distintas criadas a partir dos dados de treinamento apresentados na Seção 4.4. Os dados obtidos (6.3.1) e sua avaliação diante dos achados na literatura (6.3.2) encontram-se, respectivamente, nas próximas subseções.

6.3.1 Resultados

A Tabela 9 apresenta as métricas de avaliação do classificador baseado em Rede Neural, disposta na Seção 5.1.1, aplicado no conjunto de testes (Seção 4.6), para a classificação de 3316 exemplos, sintetizando, portanto, os resultados obtidos no presente cenário.

Tabela 9 – Resultado da predição de 3316 exemplos do conjunto de teste pelo modelo induzido baseado em uma rede neural Multilayer Perceptron

	Precisão			Sensibilidade			Medida F1			Acurácia	Log Loss
Classificadores	beijo	estalo	sopro	beijo	estalo	sopro	beijo	estalo	sopro		
Multilayer Perceptron	0,23	0,65	0,67	0,49	0,95	0,32	0,31	0,77	0,43	49,00	2,21

Fonte: Autoria própria

6.3.2 Discussão

A análise dos resultados obtidos denota que o modelo induzido com a rede neural do tipo Multilayer Perceptron (Seção 5.1.1) não obteve um resultado satisfatório. Os resultados apresentados são similares aos resultados obtidos no cenário anterior, Seção 6.2, nas Tabelas 7 e 8, principalmente se comparado ao modelo induzido com o algoritmo Árvore de Decisão, relativo as métricas de Precisão, Sensibilidade, Medida F1 e Acurácia. Acerca da hipótese sobre a dimensionalidade dos dados, discorrida na Seção 6.2, foi apresentada a influência da alta dimensionalidade do conjunto ser a causadora do desempenho não aceitável dos modelos previamente induzidos. Em Zekic-Susac, Pfeifer e Sarlija (2014) os autores concluem que, apesar de um modelo de rede neural performar melhor que todos os demais algorítimos em um domínio com alta dimensionalidade (94 dimensões), essa diferença não era estatisticamente significante para ser considerada, com exceção em relação ao algoritmo k-NN. Dessa maneira, a alta dimensionalidade não foi considerada como a causadora do desempenho não desejado.

Sobre a hipótese acerca amostragem dos dados disposta na Seção 6.2, foi realizada a indução dos modelos utilizando-se uma subamostragem para igualar os totais de exemplos das três classes do domínio. A hipótese inicial foi de que eventual desbalanceamento entre classes pudesse ser a causa do desempenho não desejável dos modelos. Após obtenção dos resultados e sua análise, a ideia foi refutada, visto que os algoritmos apresentaram desempenho inferior à indução com classes de quantidades diversas, porém com maior número de exemplos.

Portanto, uma hipótese foi aventada quanto à possibilidade de aumentar a quantidade dos exemplos das classes com menores quantidades à da classe majoritária, para

avaliar se a quantidade maior, em conjunto com um balanceamento, produziria modelos capaz de generalizar melhor.

Para produzir tal cenário foi necessária a utilização da técnica *SMOTE* (Seção 3.5) para sobre amostragem dos dados do conjunto de treinamento. Entretanto, mesmo com o sobre dimensionamento do conjunto de dados de treinamento, o desempenho da rede neural ainda se apresentou insatisfatório.

A princípio, o resultado obtido pela RN no presente domínio não era o esperado. Esperava-se que a RN, dada a sua capacidade de aprendizado em domínios complexos, como PLN e a própria análise de imagens, obtivesse um desempenho ao menos comparável ao melhor modelo induzido, no caso, com o algoritmo Random Forest, o que não se confirmou. Contudo, uma robusta literatura indicou comportamento semelhante, mesmo sem atingir as expectativas iniciais em relação ao presente resultado.

Em consulta na literatura, no trabalho Fernández-Delgado, Cernadas, Barro e Amorim (2014) foram avaliados 179 classificadores em 121 conjuntos de dados, e a conclusão foi que algoritmos da família de Random Forest são os que performam melhor na grande maioria dos domínios, seguidos pelos algoritmos da família *SVM* e posteriormente por redes neurais. Esse comportamento ocorreu neste trabalho, como pode ser observado na Tabela 7, sob a ótica da Acurácia e *Log Loss*, e, em conjunto com o resultado obtido na indução da rede neural visível na Tabela 9, onde os algoritmos citados pelos autores daquele estudo performaram exatamente na mesma ordem no presente domínio. Os autores daquele estudo salientam também que dada a complexidade dos dados, as rede neurais tendem a possuir uma melhor acurácia, se comparadas às outras famílias de algoritmos. Contudo, no domínio da presente pesquisa, tal condição não se apresentou, tendo o modelo induzido com a rede neural Multilayer Perceptron obtido a pior acurácia final e o terceiro pior *Log Loss*, atrás dos modelos induzidos com *SVM* e Random Forest, respectivamente, conforme dados da Tabela 7.

Considerando os cenários anteriores e as hipóteses relativas aos tipos de algoritmos de AM utilizados, rede neural ou acerca da quantidade dos exemplos dos conjuntos de dados, e, até o momento os resultados advindos estarem em conformidade com a literatura, o foco da experimentação passou a ser a amostragem dos dados, no tangente à variância. A nova hipótese era analisar se a variância da amostra dos dados de treinamento era suficientemente capaz de explicar os padrões desejados para indução de um modelo generalista o suficiente para predizer novos exemplos.

Primeiramente, foi realizada a indução de um modelo experimental apenas com um indivíduo, afim de avaliar se os atributos selecionados poderiam explicar a separação entre as três classes de interesse: beijo, estalo de língua e sopro. A Tabela 10 demonstra a matriz de confusão desse modelo.

Tabela 10 – Matriz de confusão do modelo induzido com uma Árvore de Decisão na implementação *J48* de um único indivíduo, denominado Indivíduo 1, do conjunto de treinamento original

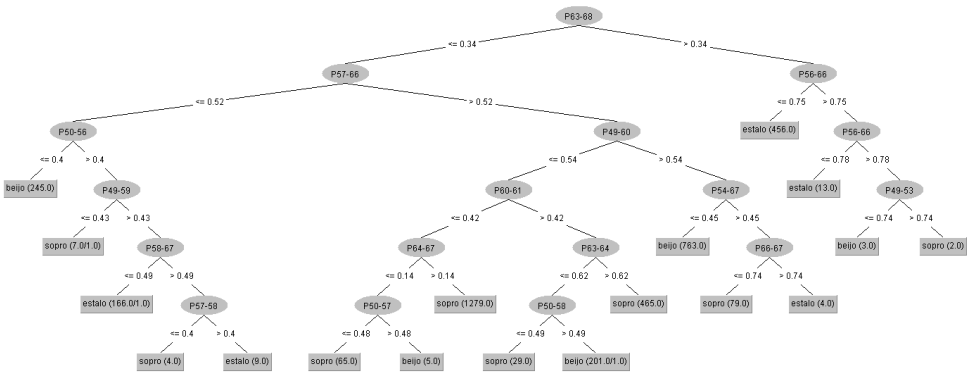
	Matriz de Confusão			
	Beijo	Estalo	Sopro	Total
Beijo	424	1	1	426
Estalo	3	198	1	202
Sopro	1	2	659	662
Total	428	201	661	1290

Fonte: Autoria própria

O modelo descrito na Tabela 10 foi induzido com 1216 exemplos do movimento de beijo, 648 exemplos de estalo e 1929 exemplo de sopro, com um total de 3793 exemplos, dos quais 66% (2503) foram utilizados para treinamento e 34% (1290) para testes de predição. O modelo atingiu uma acurácia final de 99,30%, errando ao predizer apenas nove exemplos. Contudo, o objetivo do experimento era avaliar se os atributos selecionados, um vetor de 190 distâncias Euclidianas entre os 20 pontos da boca humana (Seção 3.1) eram suficientes para justificar a separação dos três movimentos e, ainda que pesem as restrições do testes, a separação entre os movimentos se mostrou possível.

O algoritmo Árvore de Decisão foi utilizado por ser possível visualizar a tomada de decisão realizada pelo algoritmo para avaliar melhor como os atributos foram selecionados, conforme disposto na Figura 22.

Figura 22 – Visualização do modelo induzido com Árvore de Decisão, na implementação J48, demonstrando os atributos utilizados, bem como a tomada de decisão de como os atributos foram escolhidos. O modelo foi treinado apenas com exemplos de treinamento, não normalizados, de um único indivíduo denominado Indivíduo 1, extraído do conjunto de treinamento (Seção 4.4)



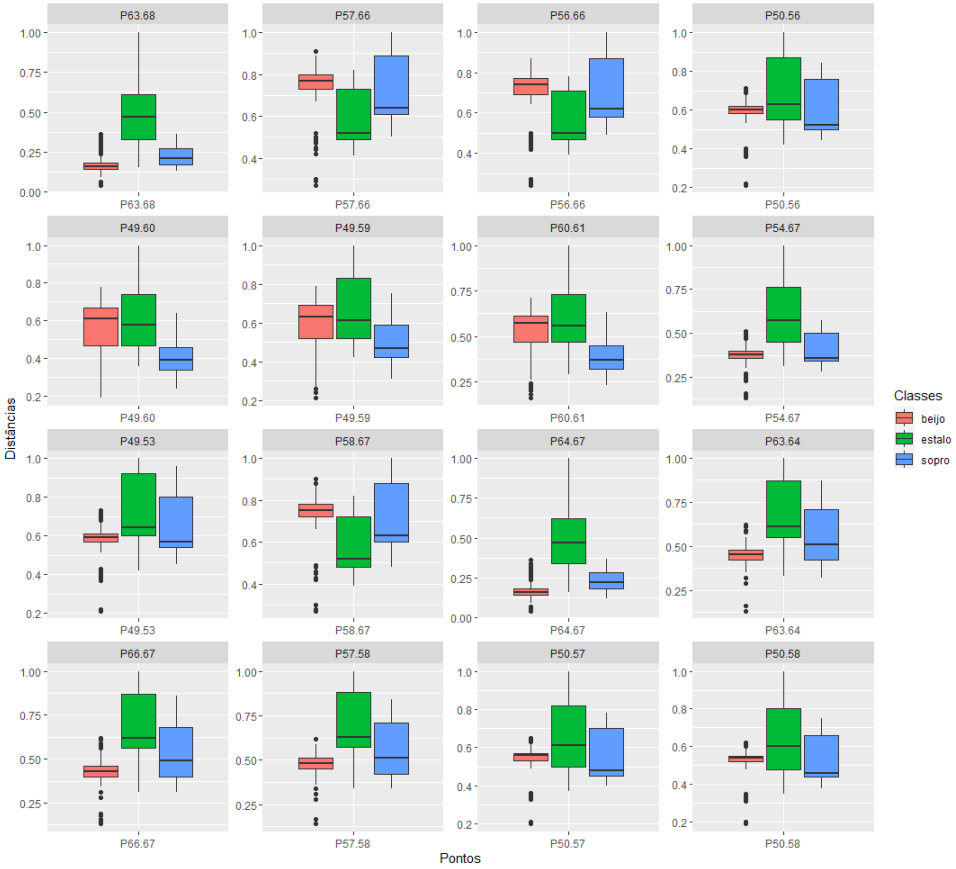
Fonte: Autoria Própria

A árvore disposta na Figura 22 possui cinco nós de profundidade à partir do nó

raiz, nesse caso considerado como sendo o atributo com a distância entre os pontos 63 e 68 (Seção 3.1). Apesar da baixa profundidade, o algoritmo utilizou 16 atributos, quase 10% do total (190) para conseguir predizer entre as três classes[5].

Para avaliar a distribuição dos dados utilizados no treinamento do modelo acima descrito, a Figura 23 apresenta os *boxplot*[6] (FRIGGE, 1989) de todos os atributos escolhidos pelo algoritmo Árvore de Decisão no processo de indução do modelo.

Figura 23 – Exemplificação dos 16 *boxplots* dos atributos utilizados pela árvore de decisão descrita à Figura 22. Representam a distribuição dos movimentos de beijo, estalo de língua e sopro, para o indivíduo, denominado, indivíduo 1, extraído do conjunto de treinamento



Fonte: Autoria Própria

Ao analisar os *boxplots* na Figura 23 em conjunto com a árvore de decisão à Figura 22, observa-se o ponto denominado "P63.68", que é a distância Euclidiana entre os pontos 63 e 68, o qual foi designado como o nó raiz da árvore no modelo induzido. Para continuar a construir a árvore, dois nós no primeiro nível foram abertos, considerando os exemplos

[5] Vale ressaltar que a árvore de decisão é construída baseada nos exemplos fornecidos, e, mesmo em um conjunto de dados apensado ao que foi utilizado no presente treinamento, é possível que não necessariamente os mesmos nós (atributos) sejam definidos, já que novas distâncias podem ser introduzidas, logo, a tomada de decisões pode considerar outros atributos como mais adequados.

[6] Um *boxplot* é uma visualização gráfica da distribuição dos dados apresentados.

que possuíam distâncias menor ou igual a 0,34 e maiores que 0,34. Considerando a unidimensionalidade, ao analisar o respectivo *boxplot* do ponto "P63.68", verifica-se que a distância de menor ou igual a 0,34 separa as classes beijo e estalo da classe sopro, pois, o valor é aproximadamente o início do segundo quartil da distribuição dos exemplos de sopro.

Idealmente, a distribuição observada nos pontos "P63.68" e "P64.67", a exemplo, são as mais desejadas, pois mesmo num plano n-dimensional, seria possível traçar uma linha reta em cada dimensão que pudesse separar as classes desejadas. Contudo, a presente classificação considerou apenas um indivíduo (denominado Indivíduo 1) para fins de exemplificação. As Figuras 24, 25 e 26 representam a distribuição de todos os indivíduos, contudo, considerando apenas o ponto "P58.63"[7], para cada tipo de movimento de interesse.

Ao analisar as Figuras 24, 25 e 26 verifica-se uma hegemonia na distribuição do movimento estalo (Figura 25) em comparação aos movimentos de beijo e sopro. Com exceção dos Indivíduos 1, 6 e 10, todos os demais movimentos de estalos concentraram-se entre as distâncias 1,5 e aproximadamente 6,0, enquanto os movimentos de beijo e sopro possuem uma distribuição entre toda a faixa de distribuição (0 à 1).

Portanto, a própria especificidade da distrição do domínio em questão, com a sobreposição das distâncias entre os três movimentos de interesse, ainda que considerados os 190 atributos escolhidos, dificulta a separação entre os três movimentos de uma vez, principalmente entre os movimentos de sopro e beijo, já que visualmente possuem a mesma amplitude e graficamente possuem distribuições que ocupam todo o espectro de distâncias.

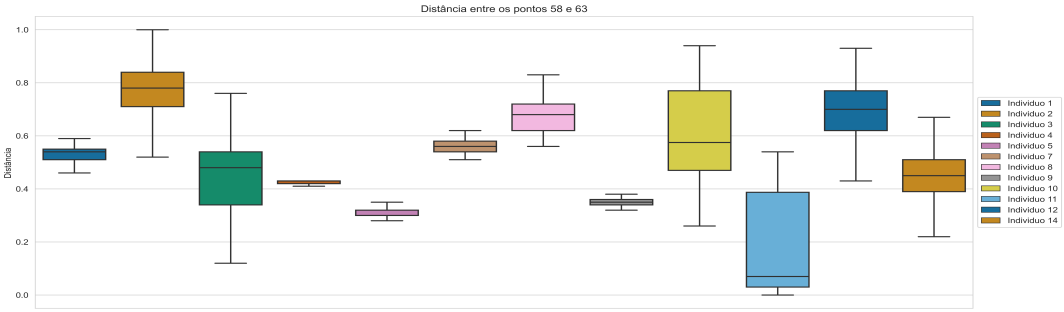
Contudo, a predição do movimento de estalo em relação aos demais sempre foi satisfatória, conforme demonstrado à Tabela 6. Considerando isso, foram elaborados os cenários finais para induzir classificadores para separar entre pares de movimentos.

6.4 Cenário de Classificação AM e RN com Dados de Treinamento

Nesta seção, o cenário de experimentação utilizou algoritmos de AM e RN treinados com duas classes distintas criadas a partir dos dados de treinamento apresentados na Seção 4.4. Os resultados originais extraídos do programa *Weka*, por meio da inferência dos modelos, incluindo todas as matrizes de confusão de cada modelo, podem ser observados no Anexo F.

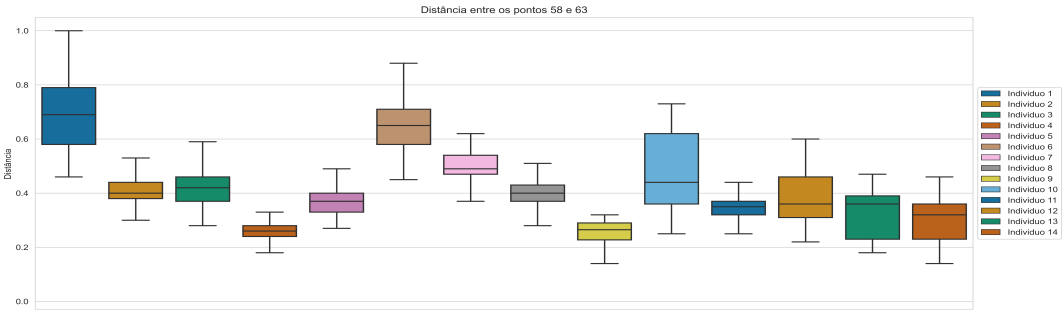
[7] A distância entre os pontos 58 e 63 foi escolhida por representar visualmente a distinção entre o movimento de estalo em relação aos movimentos de beijo e sopro.

Figura 24 – *Boxplots* da distribuição de distâncias Euclidianas entre os pontos 58 e 63 de todos os exemplos do conjunto de treinamento dos indivíduos que realizaram o movimento de beijo



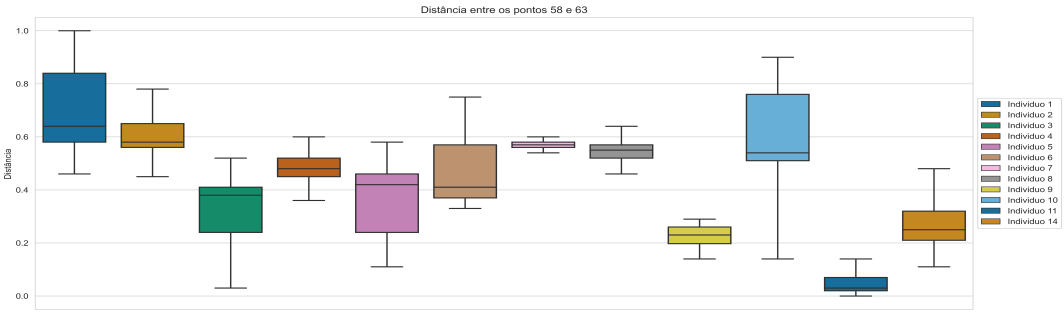
Fonte: Autoria própria

Figura 25 – *Boxplots* da distribuição de distâncias Euclidianas entre os pontos 58 e 63 de todos os exemplos do conjunto de treinamento dos indivíduos que realizaram o movimento de estalo de língua



Fonte: Autoria própria

Figura 26 – *Boxplots* da distribuição de distâncias Euclidianas entre os pontos 58 e 63 de todos os exemplos do conjunto de treinamento dos indivíduos que realizaram o movimento de sopro



Fonte: Autoria própria

6.4.1 Resultados

Os resultados obtidos estão sumarizados na Tabela 11, que apresenta uma compilação das métricas de avaliação (Seção 5.2) dos modelos treinados com os algoritmos Árvore de Decisão (J48), k-NN vizinhos mais próximos, Random Forest, Support Vector Machine (SMO) e Rede Neural (MLP).

6.4.2 Discussão

Quanto à avaliação dos dados obtidos, é possível observar um excelente desempenho em separar a classe estalo das demais. Considerando a Precisão, entre beijo e estalo foi separável com 94% de assertividade no modelo com Random Forest, já entre estalo e sopro, os movimentos foram separados entre si com 93% de exatidão. Já em relação aos movimentos de beijo e estalo, esse índice caiu para 68% de eficácia no melhor modelo (SMO), demonstrando a dificuldade explicitada na Seção 6.3 em separar tais movimentos tão similares entre si. Tal comportamento é refletido em relação às métricas de Sensibilidade e Medida F1.

Avaliando as métricas principais, assim denominadas conforme a heurística estabelecidas no presente trabalho, a Acurácia para os modelos de beijo e estalo obteve um empate em 93% de assertividade entre os modelos induzidos com Random Forest e SVM (SMO). Em relação ao modelo estalo e sopro o algoritmo SVM (SMO) teve um desempenho de 93% de acertos e por fim entre o modelo de beijo e sopro o modelo induzido com o algoritmo Random Forest obteve o melhor desempenho com 65% de exatidão.

Entretanto, a Acurácia não deve ser avaliada isoladamente, e sim, em conjunto com a métrica *Log Loss* que mede a certeza com a qual os modelos ou algoritmos tomam a decisão para predizer a classe correta, em outras palavras, o quão certo o algoritmo estava quando determinou que uma classe era a correta e as demais não fossem. Nesse caso, quanto menor for o valor de *Log Loss*, melhor, e os modelos induzidos com o algoritmo Random Forest apresentaram os menores valores dentre os demais, significativamente menores[8] e portanto, sendo considerado o melhor modelo para separar entre as classes de interesse, quando divididas em pares.

[8] Com exceção do modelo induzido com a Rede Neural *MLP* para separar entre as classes beijo e estalo que obteve um *Log Loss* de 0,27, não significante estatisticamente de 0,22 do modelo induzido com Random Forest para a mesma finalidade.

Tabela 11 – Resultado das predições dos classificadores de Árvore de Decisão (J48), k-NN, Random Forest, SVM (SMO) e rede neural Multilayer Percetron induzidos com o conjunto de treinamento e avaliados no conjunto de teste, contudo, separados em dois pares de classes por treinamento, quais sejam: beijo e estalo; estalo e sopro; beijo e sopro. Os dados marcados em negrito representam a melhor performance naquela métrica

	Precisão						Sensibilidade						Medida F1						Acurácia				Log Loss		
	beijo e estalo		estalo e sopro		beijo e sopro		beijo e estalo		estalo e sopro		beijo e sopro		beijo e estalo		estalo e sopro		beijo e sopro		beijo e estalo		estalo e sopro		beijo e sopro		beijo e estalo	estalo e sopro	beijo e sopro
Classificadores	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Treinamento	Validação	Validação	Validação	Validação
Árvore de Decisão (J48)	0,98	0,89	0,99	0,86	0,95	0,65	0,98	0,88	0,99	0,84	0,95	0,63	0,98	0,88	0,99	0,85	0,95	**0,64**	0,98	0,88	0,99	0,84	0,95	0,63	2,76	4,54	8,45
k-NN (iBK)	0,99	0,91	0,99	0,89	0,98	0,61	0,99	0,91	0,99	0,89	0,98	0,59	0,99	0,91	0,99	0,89	0,98	0,60	0,99	0,91	0,99	0,89	0,98	0,59	1,08	2,09	5,74
Random Forest	1	**0,94**	1	0,91	1	0,63	1	**0,93**	1	0,90	1	**0,65**	1	**0,93**	1	0,90	1	**0,64**	1	**0,93**	1	0,90	1	**0,65**	**0,22**	**0,23**	**0,68**
SVM (SMO)	0,96	0,92	0,97	**0,93**	0,86	**0,68**	0,96	0,92	0,97	**0,93**	0,85	0,53	0,96	0,92	0,97	**0,93**	0,82	0,56	0,96	0,92	0,98	**0,93**	0,85	0,53	1.43	2,47	16,22
Multilayer Perceptron	-	0,93	-	0,83	-	0,49	-	**0,93**	-	0,89	-	0,48	-	**0,93**	-	0,84	-	0,45	-	**0,93**	-	0,86	-	0,48	0,27	0,86	1,7

Fonte: Autoria própria

6.5 Considerações Finais

O propósito deste capítulo foi apresentar os resultados acerca dos classificadores obtidos com a aplicação do método RVPNV, tal como avaliar os dados obtidos e compará-los à literatura correlata, elucidando as propostas contidas nos objetivos iniciais desta pesquisa. A aplicação do método RNPNV, no conjunto de dados iniciais, para distinguir entre três classes simultaneamente, até os classificadores separados em pares de classes beijo e estalo, estalo e sopro e beijo e sopro, construídos com o conjunto de dados de treinamento e avaliados no conjunto de dados de testes, foi separada em quatro cenários, cada cenário é uma evolução do cenário anterior. Em cada um desses cenários, foram apresentados os resultados obtidos e o desempenho desses foi analisado e discutido com base na literatura. As conclusões finais acerca dos resultados são apresentadas no Capítulo 7.

7

Conclusão

A partir dos resultados obtidos e considerando as limitações do presente estudo, foi possível gerar modelos induzidos com Random Forest capazes de classificar entre os pares de movimentos beijo e estalo, estalo e sopro e beijo e sopro, concluindo-se que:

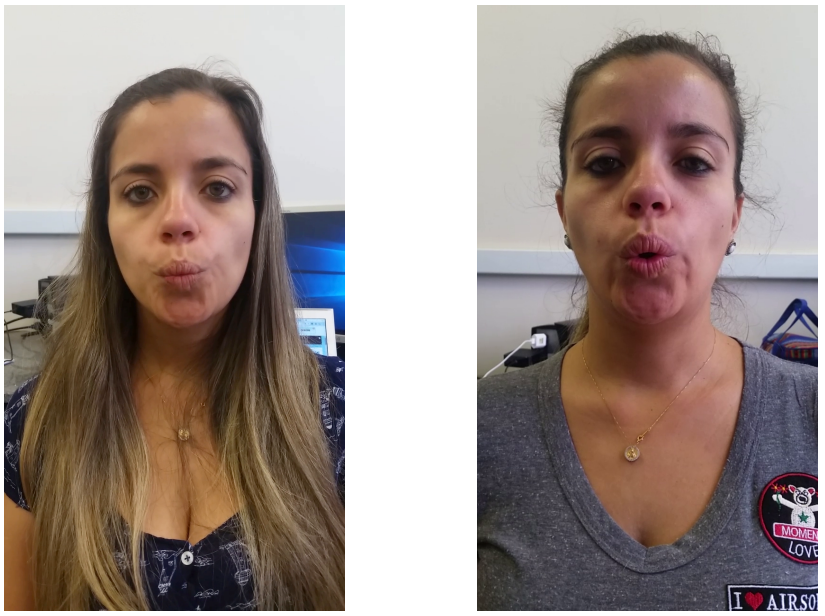
- A separação entre os movimentos sopro e beijo necessita alcançar maior precisão quanto à sua determinação (65%);
- A separação entre os movimentos beijo e estalo foi determinada com precisão adequada (93%), gerando capacidade de prosseguimento na construção do módulo proposto;
- A separação entre os movimentos estalo e sopro foi determinada com precisão adequada (90%), gerando capacidade de prosseguimento na construção do módulo proposto.

Dada a baixa acurácia no geral entre os movimentos de beijo e sopro, é válido notar que ambos os movimentos possuem amplitudes muito similares entre si. A Figura 27 demonstra a execução do ápice dos movimentos de beijo e sopro.

Uma possível diferença aparente é a separação entre os lábios inferiores e superiores internos para a passagem do ar, no caso do sopro. Essa característica é passível de exploração para tentar aumentar a acurácia na classificação entre os movimentos de beijo e sopro.

Por fim, como apontado no Capítulo 2, a literatura apresenta apenas o trabalho Souza, Souza, Watanabe, Mandrá e Macedo (2019) que explorou uma rede neural do tipo LSTM para classificar entre os movimentos de interesse. O movimento de soprou atingiu uma acurácia de 51%, o movimento de beijo uma acurácia de 66% e por fim o de estalo com 81%, com uma acurácia geral média de 66%. Não muito diferente do presente trabalho, quando se compara o melhor modelo induzido para reconhecer entre

Figura 27 – Demonstração de execução dos movimentos de beijo e sopro pelo mesmo indivíduo



(a) Execução do movimento de beijo (b) Execução do movimento de sopro

Fonte: Autoria própria

as três classes concomitantemente (Random Forest - Tabela 6.2), onde o movimento de beijo obteve 20,1% de acurácia, o movimento de sopro obteve 76,7% e o movimento de estalo obteve 89,3% com uma acurácia ponderada de 68,5%. Contudo, considerando os melhores modelos entre duas classes concomitantes, os resultados apresentados nessa dissertação superam, respeitadas as correlações dos movimentos, os resultados apresentados nos demais outros modelos que separaram entre as três classes concomitantemente.

7.1 Trabalhos Futuros

Neste estudo, vislumbrou-se a limitação da impossibilidade de gerar um modelo único capaz de distinguir entre os três movimentos de interesse, determinando, assim, resultados ampla e indiscutivelmente satisfatórios. A separação entre os movimentos de sopro e beijo atingiu, no melhor modelo, 65% de acurácia total, a exemplo. Logo, a exploração dos trabalhos futuros descritos abaixo, pode contribuir para a melhor separação entre tais movimentos.

Para prosseguimento com o estudo da temática, faz-se interessante realizar análise comparativa entre todas as distâncias descritas na Seção 3.7, mensurando assim quais melhor performam sobre o domínio desta pesquisa em conjunto com os algoritmos e redes neurais, determinando o refinamento da qualidade de performance do modelo.

Para tanto, com o intuito de produzir um modelo único capaz de diferenciar com maior assertividade os movimentos de beijo e sopro, demonstra-se atrativa a indução dos modelos através de uma abordagem multimodal como distâncias, tempo e análise do som, utilizando os mesmos dados deste trabalho.

Na etapa A do método RVPNV, foi utilizada a função get_frontal_face_detector() para reconhecimento de face, entretanto, atualmente existe na dlib uma nova função denominada cnn_face_detector(). Essa nova função aplica um modelo mais preciso que o anterior, tal modelo requer mais poder computacional para rodar e foi desenvolvido para ser executado por uma GPU (DLIB, 2022). Mas, com o avanço computacional dos dispositivos móveis, incluindo GPUs mais potentes, pode ser uma implementação que agregará mais precisão no reconhecimento facial. Vale destacar que o método não possuiu penalização estatisticamente considerável pelo não reconhecimento de faces na imagem.

YOLO (BOCHKOVSKIY, 2020) é um framework considerado atualmente o estado da arte para detecção de objetos em tempo real. A verão mais atual denominada YOLOv4 é 10% e 12% mais eficiente que a versão anterior (YOLOv3), nos comparativos de acurácia e eficiência, e, duas vezes mais rápida que o próximo concorrente direto (*EfficientDet*). Recentemente, alguns estudos emergem sobre a utilização do framework para reconhecimento de face humana (QI, 2021; GARG, 2018). Dada a superioridade desse framework na detecção de objetos, é possível explorá-lo para futuras detecções da face com o intuito de aumentar o desempenho em relação ao modelo da *dlib* utilizado nessa pesquisa.

No caso da rede neural convolucional *Mask R-CNN* (HE, 2017) é descrita como sendo o estado da arte em segmentação da imagem. Na segmentação, uma espécie de máscara é adicionada no contorno do objeto sendo detectado e esse objeto pode ser inteiramente seccionado/destacado da imagem, permitindo diversas aplicabilidades além da

detecção. Nesse caso, a *Mask R-CNN* pode ser utilizada também para a detecção de face humana (LIN, 2020). Contudo, existe uma possível lacuna de pesquisa para explorar a *Mask R-CNN* para extração de partes da face humana, como a boca, com o interesse de detectar os movimentos de beijo, estalo de língua e sopro.

Finalmente, outra possibilidade de atuação é a implementação no aplicativo desenvolvido pelo grupo "SofiaFala", de acordo com o caso de uso apresentado na Seção 5.3 que propõe a aplicação efetiva dos melhores classificadores para duas classes produzidos pelo método RVPNV.

Referências Bibliográficas

ALOM, M. Z.; TAHA, T. M.; YAKOPCIC, C.; WESTBERG, S.; SIDIKE, P.; NASRIN, M. S.; HASAN, M.; ESSEN, B. C. V.; AWWAL, A. A. S.; ASARI, V. K. A state-of-the-art survey on deep learning theory and architectures. Electronics, v. 8, n. 3, 2019. ISSN 2079-9292. Disponível em: http://dx.doi.org/10.3390/electronics8030292.

ANAS, L. F.; RAMADIJANTI, N.; BASUKI, A. Implementation of facial expression recognition system for selecting fashion item based on like and dislike expression. In: 2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC). IEEE, 2018. Disponível em: https://doi.org/10.1109/kcic.2018.8628516.

BARBOSA, T. M. M. F.; RABELO, G. R. G.; LIMA, I. L. B.; DELGADO, I. C. Avaliação da linguagem na Síndrome de Down: análise de protocolos desenvolvidos em extensão universitária. In: XXIII CONGRESSO BRASILEIRO DE FONOAUDIOLOGIA. [S.l.: s.n.], 2015. p. 6119.

BEARZOTTI, F.; TAVANO, A.; FABBRO, F. Development of orofacial praxis of children from 4 to 8 years of age. Perceptual and Motor Skills, v. 104, n. 3_suppl, p. 1355–1366, 2007. PMID: 17879670. Disponível em: https://dx.doi.org/10.2466/pms.104.4.1355-1366.

BEH, K. X.; GOH, K. M. Micro-expression spotting using facial landmarks. In: 2019 IEEE 15th International Colloquium on Signal Processing Its Applications (CSPA). [s.n.], 2019. p. 192–197. Disponível em: http://dx.doi.org/10.1109/CSPA.2019.8696059.

BELHUMEUR, P. N.; JACOBS, D. W.; KRIEGMAN, D. J.; KUMAR, N. Localizing parts of faces using a consensus of exemplars. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 35, n. 12, p. 2930–2940, Dec 2013. ISSN 0162-8828. Disponível em: https://dx.doi.org/10.1109/TPAMI.2013.23.

BLUM, A. L.; LANGLEY, P. Selection of relevant features and examples in machine learning. Artificial Intelligence, v. 97, n. 1, p. 245–271, 1997. ISSN 0004-3702. Relevance. Disponível em: https://dx.doi.org/10.1016/S0004-3702(97)00063-5.

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv, 2020. Disponível em: https://dx.doi.org/10.48550/ARXIV.2004.10934.

CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. Semi-supervised Learning. MIT Press, 2006. (Adaptive computation and machine learning). ISBN 9780262033589. Disponível em: https://books.google.com.br/books?id=kfqvQgAACAAJ.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence

Research, AI Access Foundation, v. 16, p. 321–357, jun 2002. Disponível em: https://doi.org/10.1613/jair.953.

COHN, J.; SCHMIDT, K. The timing of facial motion in posed and spontaneous smiles. International Journal of Wavelets, Multiresolution and Information Processing, v. 2, March 2004.

CUI, D.; HUANG, G.-B.; LIU, T. ELM based smile detection using Distance Vector. Pattern Recognition, v. 79, p. 356–369, 2018. Cited By 0. Disponível em: http://dx.doi.org/10.1016/j.patcog.2018.02.019.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01. Washington, DC, USA: IEEE Computer Society, 2005. (CVPR '05), p. 886–893. ISBN 0-7695-2372-2. Disponível em: http://dx.doi.org/10.1109/CVPR.2005.177.

DIBEKLIOĞLU, H.; SALAH, A. A.; GEVERS, T. Are you really smiling at me? Spontaneous versus posed enjoyment smiles. In: FITZGIBBON, A.; LAZEBNIK, S.; PERONA, P.; SATO, Y.; SCHMID, C. (Ed.). Computer Vision – ECCV 2012. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 525–538. ISBN 978-3-642-33712-. Disponível em: http://dx.doi.org/10.1007/978-3-642-33712-3_38.

DIBEKLIOĞLU, H.; SALAH, A. A.; GEVERS, T. Recognition of genuine smiles. IEEE Transactions on Multimedia, v. 17, n. 3, p. 279–294, March 2015. ISSN 1520-9210. Disponível em: http://dx.doi.org/10.1109/TMM.2015.2394777.

DIBEKLIOĞLU, H.; VALENTI, R.; SALAH, A. A.; GEVERS, T. Eyes do not lie: Spontaneous versus posed smiles. In: ACM International Conference on Multimedia. [s.n.], 2010. Disponível em: https://ivi.fnwi.uva.nl/isis/publications/2010/DibekliogluICM2010.

DLIB. CNN Face Detector Example - Dlib C++ Library. 2022. Acessado em 21/05/2022. Disponível em: http://dlib.net/cnn_face_detector.py.html.

DONOHO, D. L. High-dimensional data analysis: The curses and blessings of dimensionality. In: AMS CONFERENCE ON MATH CHALLENGES OF THE 21ST CENTURY. [S.l.: s.n.], 2000.

FARKAS, L. G.; DEUTSCH, C. K. Anthropometric determination of craniofacial morphology. American Journal of Medical Genetics, v. 65, n. 1, p. 1–4, 1996. Disponível em: https://doi.org/10.1002/ajmg.1320650102.

FERNÁNDEZ-DELGADO, M.; CERNADAS, E.; BARRO, S.; AMORIM, D. Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res., JMLR.org, v. 15, n. 1, p. 3133–3181, jan 2014. ISSN 1532-4435.

FRIGGE, M.; HOAGLIN, D. C.; IGLEWICZ, B. Some implementations of the boxplot. The American Statistician, Taylor & Francis, v. 43, n. 1, p. 50–54, 1989. Disponível em: http://dx.doi.org/10.1080/00031305.1989.10475612.

GAMA, J.; FACELI, K.; LORENA, A.; CARVALHO, A. D. Inteligência artificial: uma abordagem de aprendizado de máquina. Grupo Gen - LTC, 2011. ISBN 9788521618805. Disponível em: https://books.google.com.br/books?id=4DwelAEACAAJ.

GARCÍA, H.; ÁLVAREZ, M.; OROZCO, A. Dynamic facial landmarking selection for emotion recognition using Gaussian processes. Journal on Multimodal User Interfaces, v. 11, n. 4, p. 327–340, 2017. Cited By 0. Disponível em: http://dx.doi.org/10.1007/s12193-017-0256-9.

GARDNER, M.; DORLING, S. Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences. Atmospheric Environment, v. 32, n. 14, p. 2627–2636, 1998. ISSN 1352-2310. Disponível em: https://doi.org/10.1016/S1352-2310(97)00447-0.

GARG, D.; GOEL, P.; PANDYA, S.; GANATRA, A.; KOTECHA, K. A deep learning approach for face detection using yolo. In: 2018 IEEE Punecon. [S.l.: s.n.], 2018. p. 1–4.

GARRIDO, P.; ZOLLHöFER, M.; WU, C.; BRADLEY, D.; PÉREZ, P.; BEELER, T.; THEOBALT, C. Corrective 3d reconstruction of lips from monocular video. ACM Transactions on Graphics, Association for Computing Machinery (ACM), v. 35, n. 6, p. 1–11, nov. 2016. Disponível em: https://doi.org/10.1145/2980179.2982419.

GIACCHINI, V.; TONIAL, A.; MOTA, H. Aspectos de linguagem e motricidade oral observados em crianças atendidas em um setor de estimulação precoce. Distúrbios da Comunicação, v. 25, n. 2, 2013. ISSN 2176-2724. Disponível em: https://revistas.pucsp.br/dic/article/view/16478.

GONDHI, N. K.; KOUR, N.; EFFENDI, S.; KAUSHIK, K. An efficient algorithm for facial landmark detection using Haar-like features coupled with corner detection following anthropometric constraints. In: 2017 2nd International Conference on Telecommunication and Networks (TEL-NET). [s.n.], 2017. p. 1–6. Disponível em: https://doi.org/10.1109/TEL-NET.2017.8343517.

GONZALEZ, R. Digital Image Processing. Pearson Education, 2009. ISBN 9788131726952. Disponível em: https://books.google.com.br/books?id=a62xQ2r_f8wC.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. J. Mach. Learn. Res., JMLR.org, v. 3, p. 1157–1182, mar 2003. ISSN 1532-4435. Disponível em: http://dl.acm.org/citation.cfm?id=944919.944968.

HAYKIN, S. Redes Neurais: Princípios e Prática. [S.l.]: Artmed, 2007. ISBN 9788577800865.

HE, D.-C.; WANG, L. Texture unit, texture spectrum and texture analysis. In: 12th Canadian Symposium on Remote Sensing Geoscience and Remote Sensing Symposium,. [S.l.: s.n.], 1989. v. 5, p. 2769–2772.

HE, K.; GKIOXARI, G.; DOLLAR, P.; GIRSHICK, R. Mask r-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017. Disponível em: https://doi.org/10.1109/iccv.2017.322.

IGARASHI, T.; HUGHES, J. F. Voice as sound. In: Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01. ACM Press, 2001. Disponível em: https://doi.org/10.1145/502348.502372.

JIN, B.; QU, Y.; ZHANG, L.; GAO, Z. Diagnosing parkinson disease through facial expression recognition: Video analysis. Journal of medical Internet research, JMIR Publications, v. 22, n. 7, p. e18697–e18697, Jul 2020. ISSN 1438-8871. Disponível em: https://doi.org/10.2196/18697.

JOHNSTON, B.; CHAZAL, P. A review of image-based automatic facial landmark identification techniques. EURASIP Journal on Image and Video Processing, v. 2018, p. 86, 09 2018. Disponível em: http://dx.doi.org/10.1186/s13640-018-0324-4.

JOLLIFFE, I. T.; CADIMA, J. Principal component analysis: a review and recent developments. In: . Royal Society, 2016. v. 374. ISSN 1471-2962. Disponível em: https://doi.org/10.1098/rsta.2015.0202.

KANADE, T.; TIAN, Y.; COHN, J. F. Comprehensive database for facial expression analysis. In: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000. Washington, DC, USA: IEEE Computer Society, 2000. (FG '00), p. 46–. ISBN 0-7695-0580-5. Disponível em: http://dl.acm.org/citation.cfm?id=795661.796155.

KAZEMI, V.; SULLIVAN, J. One millisecond face alignment with an ensemble of regression trees. 2014 IEEE Conference on Computer Vision and Pattern Recognition, p. 1867–1874, 2014.

KIRA, K.; RENDELL, L. A. A practical approach to feature selection. In: Proceedings of the Ninth International Workshop on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992. (ML92), p. 249–256. ISBN 1-5586-247-X. Disponível em: http://dl.acm.org/citation.cfm?id=141975.142034.

KRAWCZYK, B. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, v. 5, n. 4, p. 221–232, Nov 2016. ISSN 2192-6360. Disponível em: https://doi.org/10.1007/s13748-016-0094-0.

KUMAR, A.; KAUR, A.; KUMAR, M. Face detection techniques: a review. Artificial Intelligence Review, Springer Science and Business Media LLC, v. 52, n. 2, p. 927–948, ago. 2018. Disponível em: https://doi.org/10.1007/s10462-018-9650-2.

LE, V.; BRANDT, J.; LIN, Z.; BOURDEV, L.; HUANG, T. S. Interactive facial feature localization. In: Proceedings of the 12th European Conference on Computer Vision - Volume Part III. Berlin, Heidelberg: Springer-Verlag, 2012. (ECCV'12), p. 679–692. ISBN 978-3-642-33711-6. Disponível em: http://dx.doi.org/10.1007/978-3-642-33712-3_49.

LEKDIOUI, K.; RUICHEK, Y.; MESSOUSSI, R.; CHAABI, Y.; TOUAHNI, R. Facial expression recognition using face-regions. In: 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP). [s.n.], 2017. p. 1–6. Disponível em: http://dx.doi.org/10.1109/ATSIP.2017.8075517.

LIN, K.; ZHAO, H.; LV, J.; LI, C.; LIU, X.; CHEN, R.; ZHAO, R. Face detection and segmentation based on improved mask r-CNN. Discrete Dynamics in Nature and Society, Hindawi Limited, v. 2020, p. 1–11, maio 2020. Disponível em: https://doi.org/10.1155/2020/9242917.

LOWE, D. G. Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2. Washington, DC, USA: IEEE Computer Society, 1999. (ICCV '99), p. 1150–. ISBN 0-7695-0164-8. Disponível em: http://dl.acm.org/citation.cfm?id=850924.851523.

LUCEY, P.; COHN, J. F.; KANADE, T.; SARAGIH, J.; AMBADAR, Z.; MATTHEWS, I. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. [s.n.], 2010. p. 94–101. ISSN 2160-7508. Disponível em: https://dx.doi.org/10.1109/CVPRW.2010.5543262.

MELONI, F.; SICCHIERI, B.; MANDRÁ, P.; BULCÃO-NETO, R.; MACEDO, A. A. A nonverbal recognition method to assist speech. In: 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS). [s.n.], 2021. p. 360–365. Disponível em: http://dx.doi.org/10.1109/CBMS52027.2021.00111.

MITCHELL, T. Machine Learning. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: https://books.google.com.br/books?id=EoYBngEACAAJ.

MUSTACCHI, Z. Guia do bebê. Companhia Editora Nacional: Associação Mais, 2009.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. Madison, WI, USA: Omnipress, 2010. (ICML'10), p. 807–814. ISBN 9781605589077. Disponível em: http://dx.doi.org/10.5555/3104322.3104425.

NAKAGAWA, E.; SCANNAVINO, K.; FABBRI, S.; FERRARI, F. Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática. Elsevier Editora Ltda., 2017. ISBN 9788535285970. Disponível em: https://books.google.com.br/books?id=kCspDwAAQBAJ.

NORVIG, P.; RUSSELL, S. Inteligência artificial: Tradução da 3a Edição. [S.l.]: Elsevier Brasil, 2014. ISBN 9788535251418.

OUANAN, H.; OUANAN, M.; AKSASSE, B. Facial landmark localization: Past, present and future. In: 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt). IEEE, 2016. Disponível em: https://doi.org/10.1109/cist.2016.7805097.

PANIS, G.; LANITIS, A.; TSAPATSOULIS, N.; COOTES, T. F. Overview of research on facial ageing using the FG-NET ageing database. IET Biometrics, v. 5, n. 2, p. 37–46, 2016. ISSN 2047-4938. Disponível em: https://doi.org/10.1049/iet-bmt.2014.0053.

PATTERSON, D. Molecular genetic analysis of Down syndrome. Human Genetics, v. 126, n. 1, p. 195–214, Jul 2009. ISSN 1432-1203. Disponível em: https://doi.org/10.1007/s00439-009-0696-.

PFISTER, T.; LI, X.; ZHAO, G.; PIETIKäINEN, M. Differentiating spontaneous from posed facial expressions within a generic facial expression recognition framework. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). [s.n.], 2011. p. 868–875. Disponível em: http://dx.doi.org/10.1109/ICCVW.2011.6130343.

PHILLIPS, P.; WECHSLER, H.; HUANG, J.; RAUSS, P. J. The FERET database and evaluation procedure for face-recognition algorithms. Image and Vision Computing, v. 16, n. 5, p. 295–306, 1998. ISSN 0262-8856. Disponível em: https://doi.org/10.1016/S0262-8856(97)00070-X.

PYLE, D. Data Preparation for Data Mining. 1st. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. ISBN 1558605290.

QI, D.; TAN, W.; YAO, Q.; LIU, J. YOLO5Face: Why Reinventing a Face Detector. arXiv, 2021. Disponível em: https://arxiv.org/abs/2105.12931.

REYES, G.; ZHANG, D.; GHOSH, S.; SHAH, P.; WU, J.; PARNAMI, A.; BERCIK, B.; STARNER, T.; ABOWD, G. D.; EDWARDS, W. K. Whoosh. In: Proceedings of the 2016 ACM International Symposium on Wearable Computers. ACM, 2016. Disponível em: https://doi.org/10.1145/2971763.2971765.

REZENDE, S. Sistemas inteligentes: fundamentos e aplicações. Manole, 2003. ISBN 9788520416839. Disponível em: https://books.google.com.br/books?id=UsJe_PlbnWcC.

RISSATO, P. H. D. G.; BULCAO-NETO, R. de F.; MACEDO, A. A. A systematic mapping on detection of human mouth landmarks. In: SBC. Anais do XVII Workshop de Visão Computacional. 2021. p. 82–87. Disponível em: https://doi.org/10.5753/wvc.2021.18894.

SAGONAS, C.; ANTONAKOS, E.; TZIMIROPOULOS, G.; ZAFEIRIOU, S.; PANTIC, M. 300 Faces In-The-Wild Challenge. Image Vision Comput., Butterworth-Heinemann, Newton, MA, USA, v. 47, n. C, p. 3–18, mar 2016. ISSN 0262-8856. Disponível em: http://dx.doi.org/10.1016/j.imavis.2016.01.002.

SAGONAS, C.; TZIMIROPOULOS, G.; ZAFEIRIOU, S.; PANTIC, M. 300 Faces in-the-Wild Challenge: The first facial landmark localization challenge. In: 2013 IEEE International Conference on Computer Vision Workshops. [s.n.], 2013. p. 397–403. Disponível em: http://dx.doi.org/10.1109/ICCVW.2013.59.

SALMAM, F. Z.; MADANI, A.; KISSI, M. Facial expression recognition using decision trees. In: 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV). [s.n.], 2016. p. 125–130. Disponível em: http://dx.doi.org/10.1109/CGiV.2016.33.

SHIVASHANKAR, S. G.; HIREMATH, S. Emotion sensing using facial recognition. In: 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon). [s.n.], 2017. p. 830–833. Disponível em: http://dx.doi.org/10.1109/SmartTechCon.2017.8358489.

SOUZA, F. C. M.; SOUZA, A. C. C.; WATANABE, C. Y. V.; MANDRÁ, P. P.; MACEDO, A. A. An analysis of visual speech features for recognition of non-articulatory sounds using machine learning. International Journal of Computer Applications, Foundation of Computer Science (FCS), NY, USA, New York, USA, v. 177, n. 16, p. 1–9, Nov 2019. ISSN 0975-8887. Disponível em: http://dx.doi.org/10.5120/ijca2019919393.

STAGNI, F.; GIACOMINI, A.; GUIDI, S.; CIANI, E.; BARTESAGHI, R. Timing of therapies for Down syndrome: the sooner, the better. Frontiers in Behavioral Neuroscience, v. 9, p. 265, 2015. ISSN 1662-5153. Disponível em: http://dx.doi.org/10.3389/fnbeh.2015.00265.

SUTTON, R.; BARTO, A.; BARTO, R.; BARTO, C.; BACH, F. Reinforcement Learning: An Introduction. Bradford Book, 1998. (A Bradford book). ISBN 9780262193986. Disponível em: https://books.google.com.br/books?id=CAFR6IBF4xYC.

VALSTAR, M. F.; PANTIC, M. Induced disgust, happiness and surprise: an addition to the MMI Facial Expression Database. In: . [S.l.: s.n.], 2010.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. [s.n.], 2001. v. 1, p. I–511–I–518 vol.1. ISSN 1063-6919. Disponível em: http://dx.doi.org/10.1109/CVPR.2001.990517.

WANG, L.; HE, D.-C. Texture classification using texture spectrum. Pattern Recogn., Elsevier Science Inc., New York, NY, USA, v. 23, n. 8, p. 905–910, aug 1990. ISSN 0031-3203. Disponível em: http://dx.doi.org/10.1016/0031-3203(90)90135-8.

WU, P.; LIU, H.; XU, C.; GAO, Y.; LI, Z.; ZHANG, X. How do you smile? Towards a comprehensive smile analysis system. Neurocomputing, v. 235, p. 245–254, 2017. Cited By 0. Disponível em: http://dx.doi.org/10.1007/s12193-017-0256-9.

YANG, T.; SHU, C.; ZHOU, N. A joint facial point detection method of deep convolutional network and shape regression. In: 2016 23rd International Conference on Pattern Recognition (ICPR). [s.n.], 2016. p. 543–548. Disponível em: https://doi.org/10.1109/ICPR.2016.7899690.

YU, X.; YANG, F.; HUANG, J.; METAXAS, D. N. Explicit occlusion detection based deformable fitting for facial landmark localization. In: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG). [s.n.], 2013. p. 1–6. Disponível em: https://doi.org/10.1109/FG.2013.6553723.

ZEKIC-SUSAC, M.; PFEIFER, S.; SARLIJA, N. A comparison of machine learning methods in a high-dimensional classification problem. Business Systems Research Journal, v. 5, 09 2014. Disponível em: http://dx.doi.org/10.2478/bsrj-2014-0021.

ZHANG, E. Y. A multimodal networked kissing machine for mobile phones. In: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct. ACM, 2016. Disponível em: https://doi.org/10.1145/2957265.2963115.

ZHANG, Z. Improved Adam optimizer for deep neural networks. In: 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). [s.n.], 2018. p. 1–2. Disponível em: http://dx.doi.org/10.1109/IWQoS.2018.8624183.

ZHU, X.; RAMANAN, D. Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. [s.n.], 2012. p. 2879–2886. ISSN 1063-6919. Disponível em: http://dx.doi.org/10.1109/CVPR.2012.6248014.

ZHU, X.; VONDRICK, C.; FOWLKES, C. C.; RAMANAN, D. Do we need more training data? International Journal of Computer Vision, v. 119, n. 1, p. 76–92, Aug 2016. ISSN 1573-1405. Disponível em: https://doi.org/10.1007/s11263-015-0812-2.

Anexos

A

Descrição Da Coleta De Dados Para o Conjunto De Treinamento

Tabela 12 – Descrição da coleta de vídeos para formação do conjunto de treinamento referente ao movimento de Beijo

Indivíduo	Qtd Vídeos	Raça	Adereços	Observações
Indivíduo 1	18	Branca	Brincos/Colar	Cabelo Preso
Indivíduo 1	88	Branca	Colar	Cabelo Solto
Indivíduo 2	21	Branca	Brincos	Cabelo Preso
Indivíduo 3	23	Branca	Óculos	Cabelo Solto
Indivíduo 4	18	Branca	-	Com barba/bigode
Indivíduo 5	19	Branca	-	Sem barba/bigode
Indivíduo 6	4	Branca	-	Sem barba/bigode
Indivíduo 6	17	Branca	Capuz	Sem barba/bigode
Indivíduo 7	12	Amarela	Óculos	Sem barba/bigode
Indivíduo 8	21	Amarela	Óculos	Cabelo Solto
Indivíduo 9	2	Branca	-	Cabelo Preso
Indivíduo 10	8	Parda	-	Sem barba/bigode
Indivíduo 10	10	Parda	Boné	Sem barba/bigode
Indivíduo 10	1	Parda	Boné e Óculos Escuros	Sem barba/bigode
Indivíduo 11	33	Branca	-	Cabelo Preso
Indivíduo 12	21	Parda	Boné	Com barba/bigode
Indivíduo 13	17	Amarela	Óculos	Sem barba/bigode
Indivíduo 14	23	Amarela	-	Sem barba/bigode

Fonte: Autoria própria

Tabela 13 – Descrição da coleta de vídeos para formação do conjunto de treinamento referente ao movimento de Estalo

Indivíduo	Qtd Vídeos	Raça	Adereços	Observações
Indivíduo 1	95	Branca	Brincos,Colar	Cabelo Preso
Indivíduo 2	13	Branca	Brincos	Cabelo Preso
Indivíduo 3	19	Branca	Óculos	Cabelo Solto
Indivíduo 4	17	Branca	-	Com barba/bigode
Indivíduo 5	15	Branca	-	Sem barba/bigode
Indivíduo 6	15	Branca	Capuz	Sem barba/bigode
Indivíduo 6	10	Branca	-	Sem barba/bigode
Indivíduo 7	11	Amarela	Óculos	Sem barba/bigode
Indivíduo 8	23	Amarela	Óculos	Cabelo solto
Indivíduo 9	2	Branca	-	Cabelo Preso
Indivíduo 10	6	Parda	Boné	Sem barba/bigode
Indivíduo 10	11	Parda	-	Sem barba/bigode
Indivíduo 11	38	Branca	-	Cabelo Preso
Indivíduo 12	19	Parda	Boné	Com barba/bigode
Indivíduo 13	15	Amarela	Óculos	Sem barba/bigode
Indivíduo 14	23	Amarela	-	Sem barba/bigode

Fonte: Autoria própria

Tabela 14 – Descrição da coleta de vídeos para formação do conjunto de treinamento referente ao movimento de Sopro

Indivíduo	Qtd Vídeos	Raça	Adereços	Observações
Indivíduo 1	105	Branca	Brinco e Colar	Cabelo Preso
Indivíduo 2	17	Branca	-	Cabelo Preso
Indivíduo 3	19	Branca	Óculos	Cabelo Solto
Indivíduo 4	17	Branca	-	Com cabelo/bigode
Indivíduo 5	19	Branca	-	Se barba/bigode
Indivíduo 6	17	Branca	Capuz	Sem barba/bigode
Indivíduo 6	3	Branca	-	Sem barba/bigode
Indivíduo 7	14	Amarela	Óculos	Sem barba/bigode
Indivíduo 8	19	Amarela	Óculos	Cabelo Solto
Indivíduo 9	2	Branca	-	Cabelo Preso
Indivíduo 10	10	Parda	Boné	Sem barba/bigode
Indivíduo 10	8	Parda	-	Sem barba/bigode
Indivíduo 11	35	Branca	-	Cabelo Preso
Indivíduo 12	21	Parda	Boné	Com cabelo/bigode
Indivíduo 13	19	Amarela	Óculos	Sem barba/bigode
Indivíduo 14	23	Amarela	-	Sem barba/bigode

Fonte: Autoria própria

B

Descrição Da Coleta De Dados Para o Conjunto De Testes

Tabela 15 – Descrição da coleta de vídeos para formação do conjunto de testes referente aos movimentos de beijo, estalo de língua e sopro. Nos movimentos de beijo e sopro foram produzidos 66 vídeos cada e no movimento de estalo o indivíduo 55 produziu um vídeo a mais, totalizando 67 vídeos

Indivíduo	Raça	Adereços	Observações
Indivíduos 42 e 47	Amarela	-	Cabelo solto
Indivíduo 20	Amarela	Colar	Cabelo solto
Indivíduos 29 e 46	Branca	-	Cabelo preso
Indivíduos 18,19,36,37,40,43,48,70,72 e 76	Branca	-	Cabelo solto
Indivíduos 15,50,53 e 65	Branca	-	Com barba/bigode
Indivíduo 41	Branca	-	Com barba/bigode e calvo
Indivíduos 22,55,59 e 60	Branca	-	Sem barba/bigode
Indivíduo 49	Branca	Brincos	Cabelo preso
Indivíduos 16 e 28	Branca	Brincos	Cabelo solto
Indivíduo 56 e 57	Branca	Brincos e Colar	Cabelo preso
Indivíduo 73	Branca	Brincos e Óculos	Cabelo solto
Indivíduos 24 e 62	Branca	Colar	Cabelo solto
Indivíduo 68	Branca	Colar	Sem barba/bigode
Indivíduos 46 e 64	Branca	Óculos	Cabelo preso
Indivíduos 26 e 54	Branca	Óculos	Cabelo solto
Indivíduo 33	Branca	Óculos	Com barba/bigode
Indivíduo 67	Branca	Óculos	Com barba sem bigode
Indivíduo 21	Branca	Óculos	Sem barba/bigode
Indivíduo 35	Branca	Óculos	Sem barba/bigode e calvo
Indivíduo 32	Branca	Piercing no Nariz e Colar	Cabelo solto
Indivíduo 23	Negra	Óculos	Cabelo solto
Indivíduo 51	Negra	Colar	Cabelo solto
Indivíduos 34 e 69	Parda	-	Cabelo preso
Indivíduos 38,39,45,52 e 66	Parda	-	Cabelo solto
Indivíduos 27,30,74 e 78	Parda	-	Com barba/bigode
Indivíduos 17,25,44,58 e 75	Parda	-	Sem barba/bigode
Indivíduo 77	Parda	Óculos	Sem barba/bigode
Indivíduo 61	Parda	Piercing no Nariz e Colar	Cabelo solto
Indivíduos 63 e 71	Parda	Brincos e Colar	Cabelo preso
Indivíduo 31	Parda	Boné	Com barba/bigode

Fonte: Autoria própria

C

Experimento do site: Motivo dos movimentos não analisáveis

Tabela 16 – Resultado das avaliações realizadas por profissionais e alunos de fonoaudiologia em relação aos vídeos disponibilizados no *website* que não puderem ser avaliados e os motivos para tanto

Avaliador(a)	Tipo movimento	Motivo
PROFISSIONAL 1	BEIJO	Não retornou ao repouso inicial
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	ESTALO	Não foi possível visualizar o movimento
PROFISSIONAL 1	BEIJO	Movimento inadequado
PROFISSIONAL 1	BEIJO	Movimento inadequado
PROFISSIONAL 1	BEIJO	Movimento inadequado
PROFISSIONAL 1	BEIJO	Não retorna ao repouso e já parte para outro movimento- sorriso
PROFISSIONAL 1	SOPRO	Não consegui perceber o inflar da bochecha
PROFISSIONAL 2	BEIJO	PROBLEMA. INSERIDO MANUALMENTE.
ALUNO 1	BEIJO	lábios unidos, contraídos e projetados. Obs: Referência Aula de Anatomia.com
ALUNO 1	BEIJO	não fecha a boca
ALUNO 1	BEIJO	lábios unidos, contraídos e projetados. Obs: Referência Aula de Anatomia.com
ALUNO 1	BEIJO	não abre a boca durante o movimento
ALUNO 1	ESTALO	o participante realizou a abertura da boca, porém não elevou a ponta da língua em direção ao palato
ALUNO 1	ESTALO	O participante não realizou um movimento rápido de sucção, gerando um som de clique.
ALUNO 1	ESTALO	Com a boca semi aberta, o participante eleva a ponta da língua em direção ao palato, porém paciente está mal posicionado com a cabeça inclinada
ALUNO 1	ESTALO	O paciente não fechou a boca

Fonte: Autoria própria

D

Resultados Originais das Induções de Algoritmos com Três Classes Distintas Induzidos com Conjunto de Dados Iniciais

São disposto em seguida os resultados originais das induções dos algoritmos Árvore de Decisão, *Random Forest*, *Support Vector Machine* e k-nn vizinhos mais próximos extraídos do programa *Weka*.

Algumas métricas não foram objetos da heurística estabelecida na presente pesquisa, a exemplo *Kappa Statistic*, *Mean absolute error*, *Root mean squared error*, *Matthews correlation coefficient - MCC*, *Receiver Operating Characteristic* (ROC) Area e *Precision-Recall Curves* (PRC) Area.

D.1 Árvore de Decisão

O Algoritmo de Árvore de Decisão, na implementação J48, foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka*. Os resultados originais da classificação e predição de novos exemplos podem ser observados na Figura 28.

Figura 28 – Resultados originais da indução (a) e predição (b) do algoritmo Árvore de Decisão, na implementação J48, para classificação entre as classes beijo, estalo e sopro

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          63               76.8293 %
Incorrectly Classified Instances        19               23.1707 %
Kappa statistic                          0.6311
Mean absolute error                      0.1758
Root mean squared error                  0.3902
Relative absolute error                 41.3882 %
Root relative squared error             84.7157 %
Total Number of Instances               82

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,652    0,102    0,714      0,652    0,682      0,567    0,710     0,583     sopro
               0,821    0,233    0,762      0,821    0,790      0,587    0,770     0,686     beijo
               0,800    0,048    0,842      0,800    0,821      0,765    0,872     0,682     estalo
Weighted Avg.  0,768    0,151    0,768      0,768    0,767      0,625    0,778     0,656

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  6  2 |  a = sopro
  6 32  1 |  b = beijo
  0  4 16 |  c = estalo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1337               40.3197 %
Incorrectly Classified Instances      1979               59.6803 %
Kappa statistic                          0.0289
Mean absolute error                      0.421
Root mean squared error                  0.6021
Total Number of Instances             3316

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,456    0,521    0,555      0,456    0,501      -0,064   0,439     0,614     sopro
               0,386    0,473    0,164      0,386    0,230      -0,069   0,471     0,192     beijo
               0,275    0,001    0,990      0,275    0,431      0,475    0,710     0,488     estalo
Weighted Avg.  0,403    0,398    0,575      0,403    0,433      0,053    0,505     0,504

=== Confusion Matrix ===

    a    b    c   <-- classified as
  889 1057    2 |    a = sopro
  394  248    0 |    b = beijo
  319  207  200 |    c = estalo
```

(b) Fonte: Autoria própria

D.2 *Random Forest*

O Algoritmo *Random Forest* foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka*. Os resultados originais da classificação e predição de novos exemplos podem ser observados na Figura 29.

Figura 29 – Resultados originais da indução (a) e predição (b) do algoritmo *Random Forest*, para classificação entre as classes beijo, estalo e sopro

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          63               76.8293 %
Incorrectly Classified Instances        19               23.1707 %
Kappa statistic                          0.6299
Mean absolute error                      0.226
Root mean squared error                  0.3325
Relative absolute error                 53.2017 %
Root relative squared error             72.1793 %
Total Number of Instances               82

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,609    0,085    0,737      0,609    0,667      0,558    0,877     0,807     sopro
                 0,846    0,233    0,767      0,846    0,805      0,614    0,886     0,859     beijo
                 0,800    0,065    0,800      0,800    0,800      0,735    0,854     0,843     estalo
Weighted Avg.    0,768    0,150    0,767      0,768    0,765      0,628    0,876     0,841

=== Confusion Matrix ===

  a  b  c   <-- classified as
 14  7  2 |  a = sopro
  4 33  2 |  b = beijo
  1  3 16 |  c = estalo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1388               41.8577 %
Incorrectly Classified Instances      1928               58.1423 %
Kappa statistic                          0.0702
Mean absolute error                      0.3836
Root mean squared error                  0.4578
Total Number of Instances             3316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,475    0,431    0,611      0,475    0,535      0,044    0,605     0,678     sopro
                 0,519    0,500    0,199      0,519    0,288      0,014    0,532     0,194     beijo
                 0,178    0,000    1,000      0,178    0,302      0,380    0,885     0,797     estalo
Weighted Avg.    0,419    0,350    0,616      0,419    0,436      0,112    0,653     0,610

=== Confusion Matrix ===

    a    b    c   <-- classified as
  926 1022    0 |    a = sopro
  309  333    0 |    b = beijo
  281  316  129 |    c = estalo
```

(b) Fonte: Autoria própria

D.3 *Support Vector Machine (SMO)*

O Algoritmo foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka* e na implementação *Sequential Minimal Optimization - SMO*. Os resultados originais da classificação e predição de novos exemplos podem ser observados na Figura 30.

Figura 30 – Resultados originais da indução (a) e predição (b) do algoritmo *SVM*, na implementação SMO, para classificação entre as classes beijo, estalo e sopro

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          59               71.9512 %
Incorrectly Classified Instances        23               28.0488 %
Kappa statistic                          0.528
Mean absolute error                      0.2954
Root mean squared error                  0.3825
Relative absolute error                 69.5322 %
Root relative squared error             83.0477 %
Total Number of Instances               82

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,261    0,034    0,750      0,261    0,387      0,344    0,678     0,456     sopro
                 0,949    0,442    0,661      0,949    0,779      0,544    0,762     0,657     beijo
                 0,800    0,032    0,889      0,800    0,842      0,797    0,873     0,760     estalo
Weighted Avg.    0,720    0,228    0,741      0,720    0,684      0,549    0,766     0,626

=== Confusion Matrix ===

  a  b  c   <-- classified as
  6 15  2 |  a = sopro
  2 37  0 |  b = beijo
  0  4 16 |  c = estalo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances         823               24.8191 %
Incorrectly Classified Instances      2493               75.1809 %
Kappa statistic                          0.0636
Mean absolute error                      0.418
Root mean squared error                  0.5192
Total Number of Instances             3316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,003    0,010    0,278      0,003    0,005      -0,046   0,588     0,634     sopro
                 1,000    0,927    0,206      1,000    0,341      0,122    0,536     0,206     beijo
                 0,242    0,000    0,994      0,242    0,390      0,445    0,710     0,543     estalo
Weighted Avg.    0,248    0,185    0,421      0,248    0,154      0,094    0,605     0,531

=== Confusion Matrix ===

    a    b    c   <-- classified as
    5 1942    1 |    a = sopro
    0  642    0 |    b = beijo
   13  537  176 |    c = estalo
```

(b) Fonte: Autoria própria

D.4 *k*-vizinhos mais próximos (*k*nn/iBK)

O Algoritmo foi utilizado com a modificação do parâmetro k para com sendo sete vizinhos mais próximos e na implementação *iBk*. Os resultados originais da classificação e predição de novos exemplos podem ser observados na Figura 31.

Figura 31 – Resultados originais da indução (a) e predição (b) do algoritmo *k*nn, na implementação iBK, para classificação entre as classes beijo, estalo e sopro

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          52               63.4146 %
Incorrectly Classified Instances        30               36.5854 %
Kappa statistic                          0.4193
Mean absolute error                      0.3035
Root mean squared error                  0.3978
Relative absolute error                 71.444  %
Root relative squared error             86.3566 %
Total Number of Instances               82

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,478    0,203    0,478      0,478    0,478      0,275    0,656     0,396     sopro
                 0,744    0,279    0,707      0,744    0,725      0,464    0,835     0,791     beijo
                 0,600    0,097    0,667      0,600    0,632      0,522    0,799     0,687     estalo
Weighted Avg.    0,634    0,213    0,633      0,634    0,633      0,425    0,776     0,655

=== Confusion Matrix ===

  a  b  c   <-- classified as
 11  6  6 |  a = sopro
 10 29  0 |  b = beijo
  2  6 12 |  c = estalo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1205               36.339  %
Incorrectly Classified Instances      2111               63.661  %
Kappa statistic                          0.0585
Mean absolute error                      0.4114
Root mean squared error                  0.5023
Total Number of Instances             3316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,329    0,330    0,587      0,329    0,422      -0,001   0,562     0,622     sopro
                 0,773    0,621    0,230      0,773    0,355      0,126    0,522     0,209     beijo
                 0,094    0,000    1,000      0,094    0,171      0,273    0,720     0,440     estalo
Weighted Avg.    0,363    0,314    0,608      0,363    0,354      0,084    0,589     0,502

=== Confusion Matrix ===

    a    b    c   <-- classified as
  641 1307    0 |    a = sopro
  146  496    0 |    b = beijo
  305  353   68 |    c = estalo
```

(b) Fonte: Autoria própria

E

Resultados Originais das Induções de Algoritmos com Três Classes Distintas Induzidos com Conjunto de Dados de Treinamento

São disposto em seguida os resultados originais das induções dos algoritmos Árvore de Decisão, *Random Forest*, *Support Vector Machine* e k-nn vizinhos mais próximos extraídos do programa *Weka*.

Algumas métricas não foram objetos da heurística estabelecida na presente pesquisa, a exemplo *Kappa Statistic*, *Mean absolute error*, *Root mean squared error*, *Matthews correlation coefficient - MCC*, *Receiver Operating Characteristic* (ROC) Area e *Precision-Recall Curves* (PRC) Area.

E.1 Árvore de Decisão

O Algoritmo de Árvore de Decisão, na implementação J48, foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka*. Os resultados originais da classificação e predição de novos exemplos podem ser observados na Figura 32.

Figura 32 – Resultados originais da indução (a) e predição (b) do algoritmo Árvore de Decisão, na implementação J48, para classificação entre as classes beijo, estalo e sopro

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11692               94.9334 %
Incorrectly Classified Instances       624                5.0666 %
Kappa statistic                          0.9114
Mean absolute error                      0.0376
Root mean squared error                  0.1803
Relative absolute error                  9.8486 %
Root relative squared error             41.2768 %
Total Number of Instances            12316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,972    0,009    0,975      0,972    0,973      0,963    0,980     0,951     estalo
                 0,961    0,056    0,959      0,961    0,960      0,906    0,950     0,937     sopro
                 0,863    0,024    0,866      0,863    0,864      0,840    0,925     0,798     beijo
Weighted Avg.    0,949    0,038    0,949      0,949    0,949      0,912    0,954     0,920

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3305   65   31 |    a = estalo
   58 6790  217 |    b = sopro
   26  227 1597 |    c = beijo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1913               57.69   %
Incorrectly Classified Instances      1403               42.31   %
Kappa statistic                          0.3083
Mean absolute error                      0.2862
Root mean squared error                  0.5229
Total Number of Instances             3316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,722    0,124    0,619      0,722    0,667      0,567    0,780     0,460     estalo
                 0,595    0,311    0,732      0,595    0,657      0,281    0,589     0,667     sopro
                 0,357    0,245    0,259      0,357    0,300      0,099    0,562     0,227     beijo
Weighted Avg.    0,577    0,257    0,616      0,577    0,590      0,308    0,626     0,536

=== Confusion Matrix ===

   a    b    c   <-- classified as
 524  109   93 |    a = estalo
 225 1160  563 |    b = sopro
  97  316  229 |    c = beijo
```

(b) Fonte: Autoria própria

E.2 *Random Forest*

O Algoritmo *Random Forest* foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka*. Os resultados originais da classificação e predição de novos exemplos podem ser observados na Figura 33.

Figura 33 – Resultados originais da indução (a) e predição (b) do algoritmo *Random Forest*, para classificação entre as classes beijo, estalo e sopro

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       12079               98.0757 %
Incorrectly Classified Instances       237                1.9243 %
Kappa statistic                          0.9662
Mean absolute error                      0.0416
Root mean squared error                  0.1126
Relative absolute error                 10.911  %
Root relative squared error             25.7952 %
Total Number of Instances            12316

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,990    0,002    0,994      0,990    0,992      0,989    1,000     1,000     estalo
               0,992    0,030    0,978      0,992    0,985      0,964    0,999     0,999     sopro
               0,922    0,006    0,967      0,922    0,944      0,935    0,997     0,986     beijo
Weighted Avg.  0,981    0,019    0,981      0,981    0,981      0,966    0,999     0,997

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3368   21   12 |    a = estalo
   14 7005   46 |    b = sopro
    7  137 1706 |    c = beijo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        2272               68.5163 %
Incorrectly Classified Instances      1044               31.4837 %
Kappa statistic                          0.4337
Mean absolute error                      0.2787
Root mean squared error                  0.3711
Total Number of Instances             3316

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,893    0,078    0,761      0,893    0,822      0,771    0,977     0,931     estalo
               0,767    0,409    0,728      0,767    0,747      0,364    0,739     0,760     sopro
               0,201    0,105    0,314      0,201    0,245      0,114    0,661     0,299     beijo
Weighted Avg.  0,685    0,278    0,655      0,685    0,666      0,405    0,776     0,708

=== Confusion Matrix ===

    a    b    c   <-- classified as
  648   73    5 |    a = estalo
  176 1495  277 |    b = sopro
   27  486  129 |    c = beijo
```

(b) Fonte: Autoria própria

E.3 *Support Vector Machine (SMO)*

O Algoritmo foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka* e na implementação *Sequential Minimal Optimization - SMO*. Os resultados originais da classificação e predição de novos exemplos podem ser observados na Figura 34.

Figura 34 – Resultados originais da indução (a) e predição (b) do algoritmo *SVM*, na implementação SMO, para classificação entre as classes beijo, estalo e sopro

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10505               85.2956 %
Incorrectly Classified Instances      1811               14.7044 %
Kappa statistic                          0.7182
Mean absolute error                      0.2566
Root mean squared error                  0.3291
Relative absolute error                 67.2631 %
Root relative squared error             75.3701 %
Total Number of Instances            12316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,937    0,020    0,946      0,937    0,942      0,919    0,979     0,927     estalo
                 0,985    0,300    0,816      0,985    0,893      0,735    0,843     0,812     sopro
                 0,192    0,005    0,866      0,192    0,315      0,372    0,745     0,346     beijo
Weighted Avg.    0,853    0,178    0,859      0,853    0,819      0,732    0,866     0,774

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3187  192   22 |    a = estalo
   70 6962   33 |    b = sopro
  112 1382  356 |    c = beijo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        2108               63.5706 %
Incorrectly Classified Instances      1208               36.4294 %
Kappa statistic                          0.3835
Mean absolute error                      0.3098
Root mean squared error                  0.4017
Total Number of Instances             3316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,815    0,032    0,877      0,815    0,845      0,805    0,937     0,824     estalo
                 0,658    0,358    0,723      0,658    0,689      0,295    0,656     0,680     sopro
                 0,366    0,237    0,270      0,366    0,311      0,115    0,631     0,247     beijo
Weighted Avg.    0,636    0,263    0,669      0,636    0,650      0,372    0,713     0,628

=== Confusion Matrix ===

    a    b    c   <-- classified as
  592   98   36 |    a = estalo
   68 1281  599 |    b = sopro
   15  392  235 |    c = beijo
```

(b) Fonte: Autoria própria

E.4 *k*-vizinhos mais próximos (*k*-NN/iBK)

O Algoritmo foi utilizado com a modificação do parâmetro k para com sendo sete vizinhos mais próximos e na implementação *iBk*. Os resultados originais da classificação e predição de novos exemplos podem ser observados na Figura 35.

Figura 35 – Resultados originais da indução (a) e predição (b) do algoritmo k-NN, na implementação iBK, para classificação entre as classes beijo, estalo e sopro

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       12098               98.2299 %
Incorrectly Classified Instances       218                1.7701 %
Kappa statistic                          0.9689
Mean absolute error                      0.0181
Root mean squared error                  0.0959
Relative absolute error                  4.7486 %
Root relative squared error             21.9599 %
Total Number of Instances            12316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,994    0,002    0,995      0,994    0,994      0,992    0,999     0,999     estalo
                 0,991    0,026    0,981      0,991    0,986      0,967    0,997     0,996     sopro
                 0,928    0,006    0,963      0,928    0,945      0,936    0,992     0,979     beijo
Weighted Avg.    0,982    0,016    0,982      0,982    0,982      0,969    0,997     0,994

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3379   10   12 |    a = estalo
    8 7003   54 |    b = sopro
    8  126 1716 |    c = beijo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        2043               61.6104 %
Incorrectly Classified Instances      1273               38.3896 %
Kappa statistic                          0.3361
Mean absolute error                      0.2696
Root mean squared error                  0.4418
Total Number of Instances            3316

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,791    0,062    0,782      0,791    0,786      0,726    0,925     0,775     estalo
                 0,665    0,423    0,692      0,665    0,678      0,241    0,666     0,688     sopro
                 0,269    0,200    0,244      0,269    0,256      0,067    0,584     0,252     beijo
Weighted Avg.    0,616    0,300    0,625      0,616    0,620      0,314    0,707     0,623

=== Confusion Matrix ===

   a    b    c   <-- classified as
 574  124   28 |    a = estalo
 145 1296  507 |    b = sopro
  15  454  173 |    c = beijo
```

(b) Fonte: Autoria própria

F

Resultados Originais das Induções de Algoritmos com Duas Classes Distintas

São disposto em seguida os resultados originais das induções dos algoritmos Árvore de Decisão, *Random Forest*, *Support Vector Machine* e k-nn vizinhos mais próximos extraídos do programa *Weka*.

Algumas métricas não foram objetos da heurística estabelecida na presente pesquisa, a exemplo *Kappa Statistic*, *Mean absolute error*, *Root mean squared error*, *Matthews correlation coefficient - MCC*, *Receiver Operating Characteristic* (ROC) Area e *Precision-Recall Curves* (PRC) Area.

F.1 Árvore de Decisão

O Algoritmo de Árvore de Decisão, na implementação J48, foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka*. Os resultados originais da classificação e predição de novos exemplos podem ser observados nas Figuras 36, 37 e 38.

Figura 36 – Resultados originais da indução (a) e predição (b) do algoritmo Árvore de Decisão, na implementação J48, para classificação entre as classes **beijo** e **estalo**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        5160               98.267  %
Incorrectly Classified Instances        91                1.733  %
Kappa statistic                          0.962
Mean absolute error                      0.0193
Root mean squared error                  0.13
Relative absolute error                  4.2391 %
Root relative squared error             27.2228 %
Total Number of Instances             5251

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,975    0,013    0,976      0,975    0,975      0,962    0,983     0,967     beijo
                 0,987    0,025    0,986      0,987    0,987      0,962    0,983     0,985     estalo
Weighted Avg.    0,983    0,021    0,983      0,983    0,983      0,962    0,983     0,979

=== Confusion Matrix ===

    a    b   <-- classified as
 1803   47 |    a = beijo
   44 3357 |    b = estalo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1211               88.5234 %
Incorrectly Classified Instances       157               11.4766 %
Kappa statistic                          0.7672
Mean absolute error                      0.1162
Root mean squared error                  0.3365
Total Number of Instances             1368
Ignored Class Unknown Instances               1949

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,790    0,030    0,958      0,790    0,866      0,778    0,587     0,232     beijo
                 0,970    0,210    0,839      0,970    0,900      0,778    0,916     0,653     estalo
Weighted Avg.    0,885    0,126    0,895      0,885    0,884      0,778    0,761     0,456

=== Confusion Matrix ===

   a   b   <-- classified as
 507 135 |   a = beijo
  22 704 |   b = estalo
```

(b) Fonte: Autoria própria

Figura 37 – Resultados originais da indução (a) e predição (b) do algoritmo Árvore de Decisão, na implementação J48, para classificação entre as classes **estalo** e **sopro**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10338               98.777  %
Incorrectly Classified Instances       128                1.223  %
Kappa statistic                          0.9721
Mean absolute error                      0.0143
Root mean squared error                  0.1088
Relative absolute error                  3.2569 %
Root relative squared error             23.2373 %
Total Number of Instances            10466

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,979    0,008    0,984      0,979    0,981      0,972    0,987     0,973     estalo
                 0,992    0,021    0,990      0,992    0,991      0,972    0,987     0,989     sopro
Weighted Avg.    0,988    0,017    0,988      0,988    0,988      0,972    0,987     0,984

=== Confusion Matrix ===

    a    b   <-- classified as
 3328   73 |    a = estalo
   55 7010 |    b = sopro
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        2251               84.181  %
Incorrectly Classified Instances       423               15.819  %
Kappa statistic                          0.6326
Mean absolute error                      0.1606
Root mean squared error                  0.3971
Total Number of Instances             2674
Ignored Class Unknown Instances                1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,850    0,161    0,663      0,850    0,745      0,643    0,777     0,526     estalo
                 0,839    0,150    0,937      0,839    0,885      0,643    0,777     0,883     sopro
Weighted Avg.    0,842    0,153    0,863      0,842    0,847      0,643    0,777     0,786

=== Confusion Matrix ===

    a    b   <-- classified as
  617  109 |    a = estalo
  314 1634 |    b = sopro
```

(b) Fonte: Autoria própria

Figura 38 – Resultados originais da indução (a) e predição (b) do algoritmo Árvore de Decisão, na implementação J48, para classificação entre as classes **beijo** e **sopro**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8459               94.885  %
Incorrectly Classified Instances       456                5.115  %
Kappa statistic                          0.8434
Mean absolute error                      0.0558
Root mean squared error                  0.2205
Relative absolute error                 16.9617 %
Root relative squared error             54.3692 %
Total Number of Instances             8915

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,867    0,030    0,884      0,867    0,876      0,843    0,919     0,831     beijo
                 0,970    0,133    0,965      0,970    0,968      0,843    0,919     0,953     sopro
Weighted Avg.    0,949    0,112    0,949      0,949    0,949      0,843    0,919     0,928

=== Confusion Matrix ===

    a    b   <-- classified as
 1604  246 |    a = beijo
  210 6855 |    b = sopro
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1622               62.6255 %
Incorrectly Classified Instances       968               37.3745 %
Kappa statistic                          0.0586
Mean absolute error                      0.3749
Root mean squared error                  0.6022
Total Number of Instances             2590
Ignored Class Unknown Instances                727

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,343    0,280    0,287      0,343    0,313      0,059    0,530     0,226     beijo
                 0,720    0,657    0,769      0,720    0,743      0,059    0,497     0,603     sopro
Weighted Avg.    0,626    0,564    0,649      0,626    0,637      0,059    0,505     0,510

=== Confusion Matrix ===

    a    b   <-- classified as
  220  422 |    a = beijo
  546 1402 |    b = sopro
```

(b) Fonte: Autoria própria

F.2 *Random Forest*

O Algoritmo *Random Forest* foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka*. Os resultados originais da classificação e predição de novos exemplos podem ser observados nas Figuras 39, 40 e 41.

Figura 39 – Resultados originais da indução (a) e predição (b) do algoritmo *Random Forest*, para classificação entre as classes **beijo** e **estalo**

```
=== Summary ===

Correctly Classified Instances        5251               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0077
Root mean squared error                  0.0277
Relative absolute error                  1.6837 %
Root relative squared error              5.8021 %
Total Number of Instances             5251

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     beijo
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     estalo
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

    a    b   <-- classified as
 1850    0 |    a = beijo
    0 3401 |    b = estalo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1276               93.2749 %
Incorrectly Classified Instances        92                6.7251 %
Kappa statistic                          0.8641
Mean absolute error                      0.1636
Root mean squared error                  0.2483
Total Number of Instances             1368
Ignored Class Unknown Instances               1949

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,871    0,012    0,984      0,871    0,924      0,869    0,677     0,305     beijo
                 0,988    0,129    0,896      0,988    0,940      0,869    0,968     0,864     estalo
Weighted Avg.    0,933    0,074    0,938      0,933    0,932      0,869    0,831     0,601

=== Confusion Matrix ===

   a   b   <-- classified as
 559  83 |   a = beijo
   9 717 |   b = estalo
```

(b) Fonte: Autoria própria

Figura 40 – Resultados originais da indução (a) e predição (b) do algoritmo *Random Forest*, para classificação entre as classes **estalo** e **sopro**

```
=== Summary ===

Correctly Classified Instances       10466               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0063
Root mean squared error                  0.0229
Relative absolute error                  1.4326 %
Root relative squared error              4.8965 %
Total Number of Instances            10466

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     estalo
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     sopro
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

    a    b   <-- classified as
 3401    0 |    a = estalo
    0 7065 |    b = sopro
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        2415               90.3141 %
Incorrectly Classified Instances       259                9.6859 %
Kappa statistic                          0.7648
Mean absolute error                      0.1605
Root mean squared error                  0.259
Total Number of Instances             2674
Ignored Class Unknown Instances                 1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,887    0,091    0,784      0,887    0,833      0,768    0,970     0,922     estalo
                 0,909    0,113    0,956      0,909    0,932      0,768    0,970     0,989     sopro
Weighted Avg.    0,903    0,107    0,909      0,903    0,905      0,768    0,970     0,971

=== Confusion Matrix ===

    a    b   <-- classified as
  644   82 |    a = estalo
  177 1771 |    b = sopro
```

(b) Fonte: Autoria própria

Figura 41 – Resultados originais da indução (a) e predição (b) do algoritmo *Random Forest*, para classificação entre as classes **beijo** e **sopro**

```
=== Summary ===

Correctly Classified Instances        8915               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0213
Root mean squared error                  0.0528
Relative absolute error                  6.4743 %
Root relative squared error             13.0122 %
Total Number of Instances             8915

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     beijo
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     sopro
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

    a    b   <-- classified as
 1850    0 |    a = beijo
    0 7065 |    b = sopro
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1691               65.2896 %
Incorrectly Classified Instances       899               34.7104 %
Kappa statistic                          0.0184
Mean absolute error                      0.4007
Root mean squared error                  0.473
Total Number of Instances             2590
Ignored Class Unknown Instances                727

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,223    0,205    0,263      0,223    0,241      0,018    0,581     0,218     beijo
                 0,795    0,777    0,756      0,795    0,775      0,018    0,432     0,546     sopro
Weighted Avg.    0,653    0,635    0,634      0,653    0,643      0,018    0,469     0,465

=== Confusion Matrix ===

    a    b   <-- classified as
  143  499 |    a = beijo
  400 1548 |    b = sopro
```

(b) Fonte: Autoria própria

F.3 *Support Vector Machine (SMO)*

O Algoritmo foi utilizado sem modificações dos parâmetros adicionais ofertados pelo programa *Weka* e na implementação *Sequential Minimal Optimization - SMO*. Os resultados originais da classificação e predição de novos exemplos podem ser observados nas Figuras 42, 43 e 44.

Figura 42 – Resultados originais da indução (a) e predição (b) do algoritmo *SVM*, na implementação SMO, para classificação entre as classes **beijo** e **estalo**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        5034               95.8675 %
Incorrectly Classified Instances       217                4.1325 %
Kappa statistic                          0.9086
Mean absolute error                      0.0413
Root mean squared error                  0.2033
Relative absolute error                  9.0547 %
Root relative squared error             42.5561 %
Total Number of Instances             5251

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,922    0,021    0,959      0,922    0,940      0,909    0,950     0,912     beijo
                 0,979    0,078    0,958      0,979    0,968      0,909    0,950     0,952     estalo
Weighted Avg.    0,959    0,058    0,959      0,959    0,958      0,909    0,950     0,938

=== Confusion Matrix ===

    a    b   <-- classified as
 1705  145 |    a = beijo
   72 3329 |    b = estalo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1257               91.886  %
Incorrectly Classified Instances       111                8.114  %
Kappa statistic                          0.8381
Mean absolute error                      0.0811
Root mean squared error                  0.2849
Total Number of Instances             1368

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,969    0,125    0,872      0,969    0,918      0,843    0,922     0,860     beijo
                 0,875    0,031    0,969      0,875    0,920      0,843    0,922     0,914     estalo
Weighted Avg.    0,919    0,075    0,924      0,919    0,919      0,843    0,922     0,889

=== Confusion Matrix ===

   a   b   <-- classified as
 622  20 |   a = beijo
  91 635 |   b = estalo
```

(b) Fonte: Autoria própria

Figura 43 – Resultados originais da indução (a) e predição (b) do algoritmo *SVM*, na implementação SMO, para classificação entre as classes **estalo** e **sopro**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10196               97.4202 %
Incorrectly Classified Instances       270                2.5798 %
Kappa statistic                          0.9406
Mean absolute error                      0.0258
Root mean squared error                  0.1606
Relative absolute error                  5.8801 %
Root relative squared error             34.2936 %
Total Number of Instances            10466

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,942    0,010    0,978      0,942    0,960      0,941    0,966     0,940     estalo
                 0,990    0,058    0,972      0,990    0,981      0,941    0,966     0,969     sopro
Weighted Avg.    0,974    0,043    0,974      0,974    0,974      0,941    0,966     0,960

=== Confusion Matrix ===

    a    b   <-- classified as
 3203  198 |    a = estalo
   72 6993 |    b = sopro
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        2483               92.8571 %
Incorrectly Classified Instances       191                7.1429 %
Kappa statistic                          0.8155
Mean absolute error                      0.0714
Root mean squared error                  0.2673
Total Number of Instances             2674
Ignored Class Unknown Instances                 1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,835    0,036    0,895      0,835    0,864      0,816    0,899     0,792     estalo
                 0,964    0,165    0,940      0,964    0,952      0,816    0,899     0,932     sopro
Weighted Avg.    0,929    0,130    0,928      0,929    0,928      0,816    0,899     0,894

=== Confusion Matrix ===

    a    b   <-- classified as
  606  120 |    a = estalo
   71 1877 |    b = sopro
```

(b) Fonte: Autoria própria

Figura 44 – Resultados originais da indução (a) e predição (b) do algoritmo *SVM*, na implementação SMO, para classificação entre as classes **beijo** e **sopro**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        7574               84.9579 %
Incorrectly Classified Instances      1341               15.0421 %
Kappa statistic                          0.3915
Mean absolute error                      0.1504
Root mean squared error                  0.3878
Relative absolute error                 45.7278 %
Root relative squared error             95.6386 %
Total Number of Instances             8915

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,301    0,007    0,921      0,301    0,454      0,474    0,647     0,422     beijo
                 0,993    0,699    0,844      0,993    0,913      0,474    0,647     0,844     sopro
Weighted Avg.    0,850    0,555    0,860      0,850    0,818      0,474    0,647     0,757

=== Confusion Matrix ===

    a    b   <-- classified as
  557 1293 |    a = beijo
   48 7017 |    b = sopro
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1374               53.0502 %
Incorrectly Classified Instances      1216               46.9498 %
Kappa statistic                          0.0931
Mean absolute error                      0.4695
Root mean squared error                  0.6852
Total Number of Instances             2590
Ignored Class Unknown Instances                727

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,632    0,503    0,293      0,632    0,400      0,112    0,606     0,239     beijo
                 0,497    0,368    0,804      0,497    0,614      0,112    0,449     0,565     sopro
Weighted Avg.    0,531    0,401    0,677      0,531    0,561      0,112    0,488     0,484

=== Confusion Matrix ===

   a   b   <-- classified as
 406 236 |   a = beijo
 980 968 |   b = sopro
```

(b) Fonte: Autoria própria

F.4 *k*-vizinhos mais próximos (*k*-NN/iBK)

O Algoritmo foi utilizado com a modificação do parâmetro k para com sendo sete vizinhos mais próximos e na implementação *iBk*. Os resultados originais da classificação e predição de novos exemplos podem ser observados nas Figuras 45, 46 e 47.

Figura 45 – Resultados originais da indução (a) e predição (b) do algoritmo k-NN, na implementação iBK, para classificação entre as classes **beijo** e **estalo**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        5231               99.6191 %
Incorrectly Classified Instances        20                0.3809 %
Kappa statistic                          0.9917
Mean absolute error                      0.0056
Root mean squared error                  0.0541
Relative absolute error                  1.2226 %
Root relative squared error             11.3323 %
Total Number of Instances             5251

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,995    0,003    0,994      0,995    0,995      0,992    0,999     0,999     beijo
                 0,997    0,005    0,997      0,997    0,997      0,992    0,999     0,999     estalo
Weighted Avg.    0,996    0,004    0,996      0,996    0,996      0,992    0,999     0,999

=== Confusion Matrix ===

    a    b   <-- classified as
 1841    9 |    a = beijo
   11 3390 |    b = estalo
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1245               91.0088 %
Incorrectly Classified Instances       123                8.9912 %
Kappa statistic                          0.8193
Mean absolute error                      0.1085
Root mean squared error                  0.2627
Total Number of Instances             1368
Ignored Class Unknown Instances              1949

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,894    0,076    0,913      0,894    0,903      0,819    0,660     0,263     beijo
                 0,924    0,106    0,908      0,924    0,916      0,819    0,932     0,733     estalo
Weighted Avg.    0,910    0,092    0,910      0,910    0,910      0,819    0,804     0,512

=== Confusion Matrix ===

   a   b   <-- classified as
 574  68 |   a = beijo
  55 671 |   b = estalo
```

(b) Fonte: Autoria própria

Figura 46 – Resultados originais da indução (a) e predição (b) do algoritmo k-NN, na implementação iBK, para classificação entre as classes **estalo** e **sopro**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10445               99.7994 %
Incorrectly Classified Instances        21                0.2006 %
Kappa statistic                          0.9954
Mean absolute error                      0.0033
Root mean squared error                  0.0383
Relative absolute error                  0.7579 %
Root relative squared error              8.1713 %
Total Number of Instances            10466

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,996    0,001    0,997      0,996    0,997      0,995    1,000     0,999     estalo
                 0,999    0,004    0,998      0,999    0,999      0,995    1,000     1,000     sopro
Weighted Avg.    0,998    0,003    0,998      0,998    0,998      0,995    1,000     1,000

=== Confusion Matrix ===

    a    b   <-- classified as
 3389   12 |    a = estalo
    9 7056 |    b = sopro
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        2382               89.08    %
Incorrectly Classified Instances       292               10.92    %
Kappa statistic                          0.7244
Mean absolute error                      0.1247
Root mean squared error                  0.3047
Total Number of Instances             2674
Ignored Class Unknown Instances                1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,802    0,076    0,797      0,802    0,799      0,724    0,921     0,762     estalo
                 0,924    0,198    0,926      0,924    0,925      0,724    0,921     0,957     sopro
Weighted Avg.    0,891    0,165    0,891      0,891    0,891      0,724    0,921     0,904

=== Confusion Matrix ===

   a    b   <-- classified as
 582  144 |    a = estalo
 148 1800 |    b = sopro
```

(b) Fonte: Autoria própria

Figura 47 – Resultados originais da indução (a) e predição (b) do algoritmo k-NN, na implementação iBK, para classificação entre as classes **beijo** e **sopro**

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8710               97.7005 %
Incorrectly Classified Instances       205                2.2995 %
Kappa statistic                          0.929
Mean absolute error                      0.034
Root mean squared error                  0.1327
Relative absolute error                 10.347  %
Root relative squared error             32.7114 %
Total Number of Instances             8915

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,924    0,009    0,964      0,924    0,943      0,929    0,990     0,979     beijo
                 0,991    0,076    0,980      0,991    0,986      0,929    0,990     0,995     sopro
Weighted Avg.    0,977    0,062    0,977      0,977    0,977      0,929    0,990     0,992

=== Confusion Matrix ===

    a    b   <-- classified as
 1709  141 |    a = beijo
   64 7001 |    b = sopro
```

(a) Fonte: Autoria própria

```
=== Summary ===

Correctly Classified Instances        1519               58.6486 %
Incorrectly Classified Instances      1071               41.3514 %
Kappa statistic                         -0.032
Mean absolute error                      0.4231
Root mean squared error                  0.5616
Total Number of Instances             2590
Ignored Class Unknown Instances                727

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,277    0,312    0,227      0,277    0,249      -0,032   0,537     0,211     beijo
                 0,688    0,723    0,743      0,688    0,715      -0,032   0,393     0,528     sopro
Weighted Avg.    0,586    0,621    0,615      0,586    0,599      -0,032   0,429     0,449

=== Confusion Matrix ===

    a    b   <-- classified as
  178  464 |    a = beijo
  607 1341 |    b = sopro
```

(b) Fonte: Autoria própria