Universidade Federal de Minas Gerais

Programa de Pós-Graduação em Engenharia Elétrica

Aplicação de Técnicas de Inteligência Computacional para Análise da Expressão Facial em Reconhecimento de Sinais de Libras.

Tamires Martins Rezende

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Aplicação de Técnicas de Inteligência Computacional para Análise da Expressão Facial em Reconhecimento de Sinais de Libras.

Tamires Martins Rezende

Dissertação de Mestrado submetida à Banca Examidadora designada pelo colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Cristiano Leite de Castro
Co-orientadora: Sílvia Grasiella Moreira Almeida

Belo Horizonte - MG
Dezembro de 2016

Rezende, Tamires Martins.
R467a Aplicação de técnicas de inteligência computacional para análise da expressão facial em reconhecimento de sinais de libras [manuscrito] / Tamires Martins Rezende. – 2016.
ix, 90 f., enc.: il.

Orientador: Cristiano Leite de Castro.
Coorientadora: Sílvia Grasiella Moreira Almeida.

Dissertação (mestrado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 64-89.

Bibliografia: f. 59-63.

1. Engenharia elétrica - Teses. 2. Detecção de sinais - Teses. 3. Língua brasileira de sinais - Teses. 4. Expressão facial - Teses. I. Castro, Cristiano Leite de. II. Almeida, Sílvia Grasiella Moreira. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

DISSERTAÇÃO DE MESTRADO *Nº* 955

Aplicação de Técnicas de Inteligência Computacional para Análise da Expressão Facial em Reconhecimento de Sinais de Libras.

Tamires Martins Rezende

Data da defesa: 16/12/2016

"Aplicação de Técnicas de Inteligência Computacional para Análise da Expressão Facial em Reconhecimento de Sinais de Libras "

Tamires Martins Rezende

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 16 de dezembro de 2016.

Por:

Prof. Dr. Cristiano Leite de Castro
(UFMG) - Orientador

Profa. Dra. Silvia Grasiella Moreira Almeida
IFMG-OP ()

Prof. Dr. Frederico Gadelha Guimarães
DEE (UFMG)

Dr. Luiz Carlos Bambirra Torres
Residente Pos-Doutoral (PPGEE-UFMG)

À minha família, especialmente aos meus sobrinhos
Ronaldo Júnio e Heitor.

Agradecimentos

Agradeço a Deus por ter me iluminado nessa etapa. Sua proteção foi indispensável para que eu me sentisse mais segura. Além disso, Deus fez questão de colocar pessoas maravilhosas no meu caminho, que sem as quais esse trabalho não seria possível.

Agradeço ao meu orientador Prof. Cristiano Castro por compreender as minhas limitações e me auxiliar sempre. Obrigada Prof^a. Sílvia Almeida por tantos encontros e conversas. Pra mim é um prazer ter sido sua aluna no IF, poder trabalhar com você novamente e continuar o seu brilhante trabalho. Me inspiro muito em vocês! Vocês são excelentes orientadores! Meu muito obrigada também ao Prof. Frederico Guimarães, por ter sido meu professor no curso técnico e ter me apresentado o mestrado da UFMG, sempre com bons conselhos.

Muitíssimo obrigado a todos do Laboratório MINDS e do grupo MINDS-Libras pelos estudos, cafés, lanches e papos jogados fora. Aos meus amigos de disciplina: Felipe, Ramon e Ciniro que sempre me auxiliaram e me fizeram rir muito!

Às pessoas que encurtavam a distância Ouro Preto - Belo Horizonte, especialmente Andréia e Alessandro, muito obrigada! Chegar em Ouro Preto sempre foi a minha maior alegria!

Meu muito obrigada ao Marcuuuus, pelas infinitas ajudas durante todo o tempo de mestrado e no IF-Itabirito. Sou muito grata a você! Obrigada também aos meus colegas de trabalho do IF-Itabirito e aos meus lindos alunos. Muitas saudades!

Por fim gostaria de agradecer às pessoas que dão sentido à minha vida: meu pai Emidio, minha mãe Nelita, meus irmãos Ronaldo, Luciano e Viviane, meus cunhados Emerson e Cristina. Vocês são os meus exemplos e se um dia eu conseguir ter o caráter do meu pai e a força e vivacidade da minha mãe, serei uma pessoa realizada. Às minhas razões de viver: Ronaldo Júnio e Heitor. O meu amor por estes dois não tem limite. Ao David pelo companheirismo e apoio. A minha família é a coisa mais importante que eu tenho e eu faço tudo por vocês. Obrigada!

E como dizia a Patrícia (IF): "Quem defende em janeiro, defende em dezembro"....

Aqui estou!!!

Fico muito feliz em compartilhar com vocês um trabalho tão motivante.

Obrigada, obrigada e obrigada.

Resumo da dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Escola de Engenharia, como um dos requisitos necessários para obtenção do título de Mestre em Engenharia Elétrica na Universidade Federal de Minas Gerais.

Aplicação de Técnicas de Inteligência Computacional para Análise da Expressão Facial em Reconhecimento de Sinais de Libras.

Tamires Martins Rezende

Dezembro / 2016

Orientadores:	Cristiano Leite de Castro e Sílvia Grasiella Moreira Almeida
Área de Concentração:	Sistemas de Computação e Telecomunicações
Palavras-chave:	Reconhecimento de Sinais, sensor RGB-D, kinect, Libras, k-NN, SVM, LBP

O reconhecimento automático de expressões faciais é um problema complexo que requer a aplicação de técnicas de Inteligência Computacional, em especial aquelas relacionadas a área de Reconhecimento de Padrões. A utilização destas técnicas nesse trabalho tem como objetivo estabelecer uma abordagem que permita a diferenciação de sinais da Língua Brasileira de Sinais, conhecida por Libras, por meio de um dos seus parâmetros fonológicos: as expressões não-manuais. Estas expressões são formadas pelo movimento da face, dos olhos, da cabeça e/ou do tronco. O objetivo principal da presente pesquisa foi mensurar a importância da expressão facial durante a execução do sinal de Libras e verificar se apenas a mudança na fisionomia é suficiente para identificar um sinal. A partir desta premissa, uma metodologia para o reconhecimento automático de sinais da Libras foi estruturada e validada por uma base de dados composta por 10 sinais de Libras capturados por um sensor RGB-D (Kinect). Esta base de sinais foi construída para esta aplicação e nela cada sinal selecionado para sua composição foi executado por apenas um sinalizador. A base de sinais de Libras disponibiliza as coordenadas (x,y) da posição de 121 pontos do rosto e os vídeos de cada gravação de cada sinal. A partir destas informações disponíveis, as etapas a seguir foram implementadas: (i) detecção e recorte da face, que é a região de interesse desse trabalho; (ii) sumarização dos vídeos com as imagens do rosto utilizando o conceito da maximização da diversidade em termos de distância temporal e da diferença de cores no padrão RGB entre os quadros. Esta etapa foi necessária para eliminar informações redundantes e por meio dela foram obtidos os cinco quadros mais significativos das gravações de cada sinal; (iii) criação de dois vetores de características: um a partir da concatenação dos 121 pontos cartesianos disponíveis na base de sinais e outro a partir da informação obtida pela aplicação do descritor de textura LBP (Padrões Locais Binários) em cada um dos quadros significativos; e (iv) classificação dos sinais aplicando o k-NN (k-vizinhos mais próximos) e a SVM (Máquina de Vetores de Suporte). Os melhores parâmetros para estes classificadores (respectivamente o parâmetro k do primeiro, e C e γ do segundo) foram obtidos a partir de validação cruzada. A classificação dos sinais da base criada por meio da característica gerada pela aplicação do descritor LBP nos quadros mais significativos dos vídeos das gravações de cada sinal teve melhor desempenho que a característica derivada da concatenação dos pontos cartesianos. Já em relação aos

classificadores, o SVM retornou melhores taxas de acerto. Com isso, a acurácia média de reconhecimento dos sinais obtida da análise da metodologia proposta aqui foi de 95,3% evidenciando a potencialidade do modelo proposto. Esse trabalho contribui para o crescimento dos estudos que envolvem os aspectos visuais próprios da estrutura da Libras e tem como foco principal a importância da expressão facial na identificação dos sinais de forma automatizada.

Abstract of the Thesis submitted to the Electrical Engineering Graduate Program, Engineering School, in partial fulfillment of the requirements for the degree of Master in Electrical Engineering at the Federal University of Minas Gerais.

Aplicação de Técnicas de Inteligência Computacional para Análise da Expressão Facial em Reconhecimento de Sinais de Libras.

Tamires Martins Rezende

Dezembro / 2016

Advisors:	Cristiano Leite de Castro and Sílvia Grasiella Moreira Almeida
Area of Concentration:	Computer Engineering , Communications and Computation
Keywords:	Signal Recognition, RGB-D sensor, kinect, Libras, k-NN, SVM, LBP

The automatic recognition of facial expressions is a complex problem that requires the application of Computational Intelligence techniques, especially those related to the area of Pattern Recognition. The use of these techniques in this work aims to establish an approach that allows the differentiation of signs of the Brazilian Sign Language, known as Libras, through one of its phonological parameters: non-manual expressions. These expressions are formed by the movement of the face, eyes, head and/or trunk. The main objective of the present research was to measure the importance of facial expression during the execution of the sign of Pounds and to verify if only the change in physiognomy is sufficient to identify a signal. From this premise, a methodology for the automatic recognition of Libras signals was structured and validated by a database composed of 10 Libra signals captured by an RGB-D (Kinect) sensor. This signal base was built for this application and in it each signal selected for its composition was executed by only one flag. The Libras signal base provides the coordinates (x,y) of the 121-point face position and the videos of each recording of each signal. From this available information, the following steps were implemented: (i) face detection and clipping, which is the region of interest in this work; (ii) summarizing videos with face images using the concept of maximizing diversity in terms of temporal distance and color difference in RGB pattern between frames. This step was necessary to eliminate redundant information and through it the five most significant frames of the recordings of each signal were obtained; (iii) creation of two characteristic vectors: one from the concatenation of the 121 cartesian points available in the signal base and another from the information obtained by applying the LBP (Binary Local Patterns) texture descriptor in each of the significant frames; and (iv) classification of the signals by applying k-NN (k-Nearest Neighbors) and SVM (Support Vector Machine). The best parameters for these classifiers (respectively the parameter k of the first, and C and γ of the second) were obtained from cross validation. The classification of the signals of the base created by means of the characteristic generated by the application of the descriptor LBP in the most significant pictures of the videos of

the recordings of each signal had better performance than the characteristic derived from the concatenation of cartesian points. In relation to the classifiers, the SVM returned better hit rates. Thus, the mean accuracy of signal recognition obtained from the analysis of the methodology proposed here was of 95.3% evidencing the potentiality of the proposed model. This work contributes to the growth of studies that involve the visual aspects of the structure of Libras and focuses on the importance of facial expression in the identification of signals in an automated way.

Sumário

Lista de Figuras

Lista de Tabelas

Abreviaturas

AAM	*Active Appearance Model*
Aca	Acalmar
Acu	Acusar
AFEW	*Acted Facial Expression in the Wild*
Ani	Aniquilar
ANOVA	Análise de Variância
Apa	Apaixonado
BSL	*Brazilian Sign Language*
CM	Configuração da Mão
CNN	Rede Neural Convolucional (*Convolutional Neural Network*)
dB	Decibéis
Eng	Engordar
ENM	Expressões Não-Manuais
Fel	Felicidade
FN	Falso Negativo
FP	Falso Positivo
GLCM	*Gray Level Co-ocorrence Matrix*
HAPPEI	*Happy People Images*
HOG	Histograma de Gradientes Orientados (*Histogram Oriented Gradients*)
IBGE	Instituto Brasileiro de Geografia e Estatística
ICP	*Iterative Closest Point*
IDE	Ambiente de Desenvolvimento Integrado *Integrated Development Environment*
JAFFE	*Japonese Female Facial Expression Database*
k-NN	k-Vizinhos mais Próximos (*k-Nearest Neighbors*)
LBP	Padrões Locais Binários (*Local Binary Patterns*)
LED	Diodo Emissor de Luz
Libras	Língua Brasileira de Sinais
LIBSVM	*A Library for Support Vector Machines*
M	Movimento
Mag	Magro
MIZ	Momentos Invariantes de Zernike (*Invariant Zernik Moments*)
MLP	Perceptron de Múltiplas Camadas (*Multi-Layer Percetron*)
MSES	*Memetic Sef-Adaptive Evolution Strategies*
OMS	Organização Mundial da Saúde
Or	Orientação da Palma da Mão
PA	Ponto de Articulação
PCA	Análise de Componentes Principais *Principal Component Analysis*
PLS	*Partial Least Squares*
PS	*Perceptron* Simples (*Single Perceptron*)
RBF	Função de Base Radial (*Radial Basis Function*)
RGB-D	Vermelho, Verde, Azul e Profundidade (*Red, Green, Blue and Depth*)
RNA	Rede Neural Artificial

SFEW	*Static Facial Expression in the Wild*
SIBGRAPI	*Conference on Grafics, Pattern and Images*
SIMMEC	Simpósio de Mecânica Computacional
SKD	*Software Development Kit*
SOM	Mapas Auto-Organizáveis (*Self Organized Maps*)
Sor	Sortudo
Sur	Surpresa
SVM	Máquina de Vetores de Suporte (*Support Vector Machines*)
TPr	Taxa de Verdadeiros Positivos
VP	Verdadeiro Positivo
WFPA	*Workshop on Face Processing Applications*
Zan	Zangado

Capítulo 1

Introdução

Sumário

1.1 Motivação

A Língua Brasileira de Sinais, conhecida por Libras, é um meio de comunicação e expressão da comunidade surda. De acordo com a Lei n^o 10.436/2002, a Libras é um "sistema linguístico de natureza visual-motora, com estrutura gramatical própria" que, por sua vez, constitui "um sistema linguístico de transmissão de ideias e fatos, oriundos de comunidades de pessoas surdas do Brasil". Por meio desta lei, a Libras tornou-se a segunda língua oficial do Brasil.

Segundo a Organização Mundial da Saúde (OMS), "mais de 5% da população do mundo - 360 milhões de pessoas - tem surdez incapacitante[1]" (OMS, 2015). Em relação à população brasileira, cerca de 5,09% da população - 9,7 milhões de pessoas - possuem alguma deficiência auditiva, de acordo com o censo 2010 do IBGE (Instituto Brasileiro de Geografia e Estatística) (Almeida, 2014).

Desde que a língua foi oficializada, temas como a inclusão e a acessibilidade dos surdos na sociedade foram mais discutidos. Atualmente existem ações que proporcionam esta acessibilidade tais como leis que obrigam o ensino de Libras em cursos de licenciatura; que exigem a disponibilidade de um intérprete em ambientes públicos, como nos hospitais, para possibilitar o atendimento de um surdo; e que impõe a presença de tradutores ou intérpretes da Libras necessariamente, pelo menos, em eventos em que surdos estejam presentes. Além destas, há ainda uma infinidade de ações que podem ser feitas para auxiliar a aprendizagem de Libras e facilitar a comunicação entre surdos e ouvintes, em especial quando se trata da aplicação de técnicas computacionais.

[1]Perda maior que 40 decibéis (dB) em adultos e uma perda auditiva superior a 30 dB em crianças com audição.

A menor unidade da Libras é chamada sinal. Para determinar seu significado, torna-se importante a localização das mãos em relação ao corpo, a expressão corporal, a orientação da palma da mão, entre outros. Além das características citadas, há um importante parâmetro para diferenciar sinais, denominado expressões não-manuais, ou seja, expressões formadas pelo movimento da face, dos olhos, da cabeça e/ou do tronco que compõem a construção sintática da língua (Almeida, 2014). São parâmetros fonológicos da Libras: o ponto de articulação, a configuração das mãos, o movimento, a orientação da palma da mão e as expressões não-manuais (Almeida, 2014)(Rezende et al., 2016)(Almeida et al., 2014)(Almeida et al., 2013), conforme apresentado na figura 1.1.

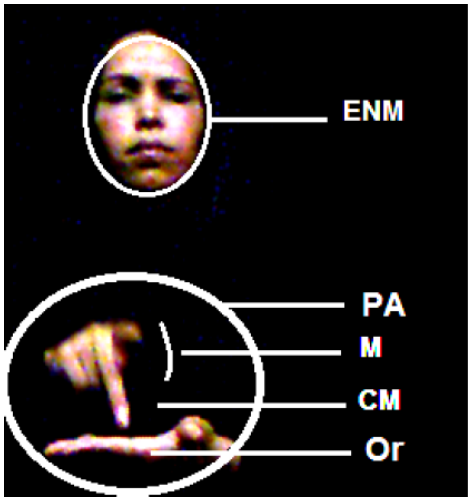


Figura 1.1: Parâmetros fonológicos da Libras: Ponto de articulação (PA), Configuração de mão (CM), Movimento (M), Orientação da palma da mão (Or) e Expressões não-manuais (ENM).

Fonte: Almeida (2014)

Reconhecer sinais de Libras baseado em seus parâmetros fonológicos por meio de ferramentas computacionais é desafiador por vários fatores, sejam eles:

- atualmente não há disponível na literatura uma base completa de sinais da língua e em formato que permita validação robusta de sistemas automáticos;
- o sinal é composto por vários elementos e não é tarefa trivial identificar qual o início e o fim do sinal; e
- um mesmo gesto pode ser executado de formas distintas por pessoas diferentes.

Duduchi e Capovilla (2006) apresentam as dificuldades ao se organizar um dicionário de sinais de Libras de forma acessível aos seus usuários. Uma possibilidade é que o dicionário seja organizado pelo aspecto viso-espacial do sinal, articulação da mão e imagem do movimento envolvido (Duduchi e Capovilla, 2006) e não por ordem alfabética como na língua portuguesa. A figura 1.2 ilustra a representação de uma palavra no Dicionário Enciclopédico Ilustrado Trilíngue[2]. Percebe-se, neste caso, a importância dos desenhos e do alfabeto gestual como partes integrantes para descrever o sinal, utilizando ferramentas visuais acessíveis aos surdos.

[2]O dicionário é dito trilíngue por ser baseado na linguagem oral, escrita e de sinais, além de apresentar os termos em Português, Inglês e Libras.

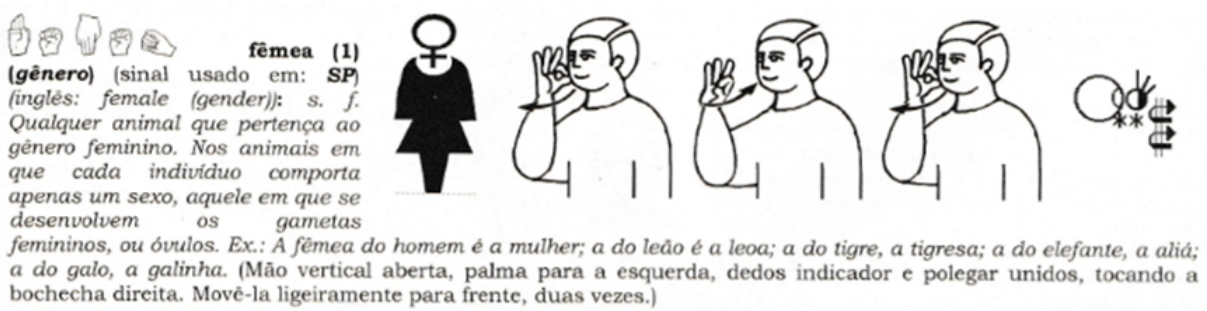


Figura 1.2: Significado da palavra "fêmea" conforme descrito no Dicionário Enciclopédico Ilustrado Trilíngue.

Fonte: Capovilla et al. (2012a)

O trabalho em questão é um estudo exploratório sobre o potencial de reconhecimento automático de sinais a partir de um importante parâmetro fonológico da Libras: a expressão facial. A expressão facial é uma forma de comunicação não verbal resultante de determinadas configurações ou contrações dos músculos faciais que provocam modificações e deformações na face (Fasel e Luettin, 2003). Dessa forma, uma base de dados experimental com 10 sinais de Libras[3] foi criada para validar tal método de reconhecimento. Cada sinal foi gravado dez vezes pelo sensor RGB-D *Kinect* e executado por apenas um sinalizador[4]. A metodologia implementada para classificar os sinais, a partir do conjunto de dados disponíveis, consistiu em:

1. detecção e recorte da região de interesse (rosto) nos vídeos de cada sinal;
2. sumarização do vídeo utilizando o conceito da maximização da diversidade;
3. criação do vetor de características; e
4. classificação do sinal.

Na literatura ainda são poucos os trabalhos que estudam apenas a contribuição da face para a classificação de um sinal de Libras. Apesar desse trabalho tratar de apenas um dos parâmetros fonológicos da língua, buscou-se complementar o trabalho feito em Almeida (2014) que fez o reconhecimento dos sinais de Libras pelos parâmetros fonológicos relacionados à trajetória das mãos, tendo, em ambos, metodologias muito similares.

Essa dissertação é um estudo exploratório das peculiaridades que envolvem as expressões não-manuais para o reconhecimento da língua brasileira de sinais. Ao final desse estudo é possível identificar qual é a melhor representação para os elementos da base de dados, além de identificar qual o melhor classificador e seus melhores parâmetros para esse tipo de aplicação dentre os classificadores analisados. Outra contribuição desse trabalho é a criação da base experimental contendo sinais de Libras. Este conjunto de dados tenta representar em parte sinais similares e outros diversos para que o sistema seja robusto o suficiente para classificá-los. É uma área de pesquisa em ascensão e que ainda não tem

[3]Sinais presentes na base experimental: (1) Acalmar, (2) Acusar, (3) Aniquilar, (4) Apaixonado, (5) Engordar, (6) Felicidade, (7) Magro, (8) Sortudo, (9) Surpresa e (10) Zangado.

[4]Sinalizador é a pessoa que executa os sinais em Libras.

nenhum sistema robusto para a classificação dos sinais, sendo eles fonemas, frases ou até mesmo uma conversa, a partir do ponto de vista da Visão Computacional. Ainda que existam dificuldades inerentes a este problema de reconhecimento, o projeto é motivante pelo impacto social que um sistema desse porte pode alcançar.

1.2 Objetivos

O objetivo dessa dissertação é realizar um estudo exploratório a partir da aplicação de técnicas de Inteligência Computacional para reconhecer sinais de Libras por meio da expressão facial.

Para que o objetivo seja alcançado, os seguintes objetivos específicos foram estabelecidos:

- Criar uma base experimental de dados, que contenha sinais cuja expressão facial se altere ao longo de sua execução;
- Aplicar um extrator de características que represente bem os dados; e
- Implementar um sistema de classificação.

1.3 Sistemas de Reconhecimento Automático de Padrões

Diante da tamanha diversidade de sistemas de reconhecimento automático de padrões, esse estudo engloba conceitos abordados em trabalhos sobre o reconhecimento de expressões faciais e sobre o reconhecimento de sinais de Libras. Estas abordagens pertencem a uma gama de problemas classificados como reconhecimento de padrões que, por sua vez, é um ramo da Visão Computacional. De acordo com Pedrini e Schwartz (2008), a Visão Computacional procura auxiliar a resolução de problemas altamente complexos, buscando imitar a cognição humana e a habilidade do ser humano em tomar decisões de acordo com as informações contidas, por exemplo, em uma imagem. As tarefas básicas para resolver este tipo de problema são:

- Aquisição da imagem/vídeo;
- Segmentação da região/objetivo de interesse;
- Extração das características sobre a região de interesse;
- Seleção de características; e
- Classificação da imagem/vídeo.

Para alcançar resultados significantes nas áreas citadas, deve-se ter cuidado em relação à representação dos dados, à extração de características, ao algoritmo de aprendizado e

a representatividade dos dados de treinamento. Estes fatores interferem diretamente na qualidade da solução do problema. Dessa forma, em qualquer um dos ramos de pesquisa, deve ser realizado um estudo profundo sobre o problema e os dados a serem trabalhados.

1.3.1 Reconhecimento de Expressões Faciais

Há muitos estudos que buscam entender as emoções humanas a partir da análise das expressões faciais. Ekman e Friesen (1971) estabeleceram em seu trabalho as 6 emoções universais: felicidade, tristeza, surpresa, raiva, desgosto e medo, e muitos estudos tem como foco reconhecê-las. Hamester et al. (2015) identificaram estes sentimentos utilizando uma abordagem de rede neural convolucional (CNN, do inglês, *Convolutional Neural Network*) multi-canais. Cada canal ficou responsável por um tipo de informação e seus resultados se combinaram alcançando uma acurácia média de 95,8%. Os seus experimentos foram aplicados ao *dataset* JAFFE (*Japonese Female Facial Expression Database*) (Lyons et al., 1998).

Pedroso e Salles (2012) também utilizaram o *dataset* JAFFE, reconhecendo os sentimentos em três etapas, sejam elas, detecção da face através do algoritmo de Viola-Jones (Ojala et al., 1996), extração de características pelo método estatístico AAM (*Active Appearance Model*) e, por fim, a classificação dos padrões de emoção. Inicialmente utilizou-se o k-NN (*k-Nearest Neighbors*) como classificador e, em busca de mais robustez, utilizou-se o SVM (*Support Vector Machine*). Suas melhores taxas de acerto foram obtidas quando o SVM com *kernel* RBF (Função de Base Radial, do inglês, *Radial Basis Function*) foi aplicado.

A JAFFE é uma base de dados composta por 219 imagens de 10 indivíduos do sexo feminino em 3 ou 4 poses nas expressões neutra, felicidade, tristeza, surpresa, raiva, nojo e medo. Outras bases de dados de expressão facial também estão disponíveis na literatura, tais como: AFEW (*Acted Facial Expression in the Wild*) (Dhall et al., 2012a), SFEW (*Static Facial Expressions in the Wild*) (Dhall et al., 2011), HAPPEI (*Happy People Images*) (Dhall et al., 2012b) e *Kyoto natural images dataset* (Doi et al., 2003).

Shan et al. (2005), Zhao e Pietikainen (2007) e Shan et al. (2009) fizeram o reconhecimento de expressões faciais extraindo as características das imagens por meio do descritor de textura LBP (*Local Binary Patterns*). Shan et al. (2005) e Shan et al. (2009) validaram seus experimentos usando o *Cohn-Kanade Facial Expression Database* (Kanade et al., 2000), alcançando uma acurácia de até 95% com o classificador SVM.

Em Liu et al. (2016), o reconhecimento das expressões raiva, desgosto, medo, felicidade, tristeza e surpresa foi realizado, alcançando uma taxa de reconhecimento média de 96,3%. Neste trabalho, as características da base estendida *Cohn-Kanade* (CK+) foram extraídas por meio de um algoritmo baseado na combinação de valores dos *pixels* em cinza e nas características obtidas pela aplicação do operador LBP. Além disso, a técnica PCA (Análise de Componentes Principais, do inglês, *Principal Component Analysis*) foi utilizada para reduzir as dimensões das características que são combinadas pelo valor de *pixel* cinza e recursos do LBP.

Por fim, há trabalhos que enfatizam o estado emocional evidenciado pela expressão facial. Essa linha de pesquisa é definida como Computação Afetiva (Sousa et al., 2016).

Sousa et al. (2016) em seu trabalho desenvolveu um processo de reconhecimento de aspectos emocionais, capturando as imagens com o sensor RGB-D *Kinect*. O rosto foi detectado com o algoritmo Viola-Jones e pontos característicos do rosto foram encontrados por meio da técnica CANDIDE-3 (Ahlberg, 2001). Tendo como referência as 6 unidades de animação fornecidas pelo SDK (*Software Development Kit*) do Kinect versão 1.8, o processo de inferência foi modelado. Um limiar entre essas unidades foi estipulado e as emoções foram classificadas.

1.3.2 Reconhecimento de Sinais de Libras

Trabalhos na linha de reconhecimento de sinais de Libras estão cada vez mais presentes na literatura. Devido a complexidade da tarefa, problemas pontuais vem sendo atacados de forma progressiva.

Carneiro et al. (2009), Santos et al. (2015), Gonçalves et al. (2016) e Koroishi e Silva (2015) fizeram estudos baseados na captura de gestos realizados pelas mãos. Carneiro et al. (2009) fez o reconhecimento das 26 letras do alfabeto através da segmentação da mão, aplicando os Movimentos Invariantes de Hu (Hu, 1962) para descrever os objetos das imagens, pré-classificando-os com uma rede SOM (Mapas Auto-Organizáveis, do inglês, *Self Organized Maps*) e classificando-os por meio de redes neurais supervisionados (PS - *Perceptron* Simples, do inglês, *Single Perceptron* e MLP - *Perceptron* de Múltiplas Camadas, do inglês, *Multi-Layer Perceptron*). Ambos os classificadores apresentaram taxas de acerto médias acima de 89%. Gonçalves et al. (2016) também utilizou o alfabeto como padrão, mas excluiu as letras "Z" e "J" que exigem movimento. Assim, seu trabalho atuou em detecção de sinais estáticos. Para o reconhecimento foi utilizado uma RNA (Rede Neural Artificial), obtendo uma acurácia de 88%. Ainda nesta vertente, Estrela et al. (2013) trabalharam com a linguagem americana de sinais e fizeram o reconhecimento do alfabeto manual, também excluindo as letras "Z" e "J". Utilizando o preditor PLS (*Partial Least Squares*), obtiveram 71,51% como melhor resultado de reconhecimento.

Já Santos et al. (2015) alcançou uma taxa média de acerto de 95,70% ao reconhecer 61 configurações de mão. Neste trabalho a mão foi segmentada e um pós processamento foi aplicado para casos em que as mãos não estavam posicionadas na frente do corpo. Para a extração de características, a técnica 2D^{2}LDA (Noushath et al., 2006) que faz a projeção da imagem em matrizes foi aplicada. No fim, os dados foram classificados com k-NN.

Koroishi e Silva (2015) tiveram como objeto de interesse a mão direita. Para reconhecer a mão, utilizaram-se modelos estruturados em nuvem de pontos e o algoritmo ICP (*Iterative Closest Point*), obtendo 65% de acerto para 48 testes envolvendo 12 sinais dinâmicos diferentes. Utilizando 40 sinais da libras, dentre letras, números e palavras, Bastos (2015) criou a sua própria base de dados e com os descritores HOG (Histograma de Gradientes Orientados, do inglês, *Histogram Oriented Gradients*) e MIZ (Momentos Invariantes de Zernike, do inglês, *Invariant Zernike Moments*), alcançou 96,77% aplicando uma rede neural para classificação.

Freitas et al. (2014b) classificaram expressões faciais gramaticais na língua de sinais. Eles criaram sua própria base de dados, capturando as expressões com *Kinect*. Em cada um dos quadros do vídeo capturado, as coordenadas (x,y,z) de 17 pontos da face foram

computadas. Com isso, cada vídeo foi representado pela distância e pelos ângulos entre estes pontos e uma técnica de Aprendizagem de Máquina classificou as expressões.

Por fim, destacam-se os estudos realizados em Almeida (2014) e Almeida et al. (2014), os quais foram a base para esse trabalho. Nele a captura dos sinais de Libras foi realizada por meio do sensor RGB-D *Kinect* que forneceu as imagens de intensidade RGB, de profundidade e imagens que marcam posições do corpo humano para cada uma das gravações dos sinais. A extração de características baseou-se nos parâmetros fonológicos da língua: ponto de articulação, movimento, orientação da palma da mão e configuração da mão, e a classificação foi realizada aplicando a máquina de vetores de suporte. Por meio desta metodologia pode-se verificar que a extração de características a partir da estrutura fonológica é um método promissor, alcançando uma acurácia média de 80%.

Nota-se aqui a diversidade de abordagens tanto no estudo das expressões faciais quanto no reconhecimento de sinais em Libras. Estas pesquisas justificam a necessidade de um sistema automático para o reconhecimento da língua de sinais brasileira e a criação de uma base de dados que proporcione a validação deste sistema.

1.4 Organização do Trabalho

A dissertação está organizada da seguinte forma:

- O **Capítulo 2** descreve as ferramentas utilizadas e o protocolo de gravação da base experimental de dados;
- O **Capítulo 3** explica os descritores utilizados nesse trabalho para extrair as características da base de dados;
- O **Capítulo 4** demonstra a metodologia formulada para classificar os sinais de Libras;
- O **Capítulo 5** apresenta os resultados obtidos, além de uma análise estatística destes valores;
- O **Capítulo 6** expõe as conclusões desse estudo e as sugestões para trabalhos futuros.

1.5 Lista de Publicações

Os seguintes trabalhos científicos foram aceitos para publicação durante a elaboração da dissertação e estão expostos no apêndice A:

Simpósio:

1. XII Simpósio de Mecânica Computacional (XII SIMMEC)
   Reconhecimento de Expressões Faciais da Língua Brasileira de Sinais (LIBRAS) utilizando os classificadores k-NN e SVM.

Conferência:

1. SIBGRAPI - XXIX Conference on Graphics, Patterns and Images
   WFPA - Workshop on Face Processing Applications
   An approach for Brazilian Sign Language (BSL) recognition based on facial expression and k-NN classifier.

Capítulo 2

Base de Dados

Sumário

2.1 Introdução

Um dos desafios desse trabalho foi encontrar uma base de dados de sinais de Libras. Atualmente, não existe uma base padronizada contendo os sinais em um formato que permita a validação de sistemas de classificação computacional de forma robusta. O que há na literatura são trabalhos que criaram o seu próprio conjunto de dados para validar suas metodologias (Freitas et al., 2014b)(Almeida, 2014)(Almeida et al., 2014)(Dias et al., 2006)(Koroishi e Silva, 2015)(Kadir et al., 2004)(Gonçalves et al., 2016)(Diniz et al., 2013)(Santos et al., 2015)(Chao et al., 2013). Freitas et al. (2014b) criaram uma base de dados composta por 18 vídeos gravados por um sensor *Kinect*, na qual em cada vídeo, um usuário executa 5 vezes, em frente ao sensor, 5 frases em Libras que exigem o uso de uma expressão facial gramatical. Este conjunto de dados está disponível na plataforma da *UCI Machine Learning Repository* (Freitas et al., 2014c). Já em Almeida (2014), uma base de dados composta por 34 sinais de Libras foi criada, mas com o foco na trajetória das mãos. Portanto, não houve grande mudança na expressão facial mesmo quando a fisionomia fazia parte do sinal.

Dessa forma, para estudar a importância da face na composição de um sinal e o quanto um sinal pode ser entendido e até mesmo diferenciado pela expressão facial, uma nova base foi criada com o foco na fisionomia. Assim, nesse capítulo são descritas as ferramentas computacionais utilizadas para a criação da base de dados, bem como o seu protocolo de gravação.

2.2 Ferramentas Computacionais

Há um vasto conjunto de ferramentas computacionais disponíveis que podem ser utilizadas para a construção de uma base de dados. A decisão sobre qual *software* e *hardware* utilizar partiu da experiência e estudos já realizados por Almeida (2014). Para capturar o vídeo de um sinal de Libras, optou-se por utilizar sensor RGB-D *Kinect* operado pelo *software* *nuiCaptureAnalyse* e para manipular os dados gerados utilizou-se o Matlab R2016b e o RStudio.

2.2.1 Sensor *Kinect*

O sensor RGB-D *Kinect*[5] é uma câmera de baixo custo muito utilizada nos videogames atuais como acessório para jogos interativos. Operado por um *software* específico nesse trabalho, o *Kinect* permitiu gravar quatro informações:

1. vídeo de intensidade RGB;
2. vídeo com as imagens de profundidade;
3. as coordenadas dos 20 pontos do corpo ilustrados na figura 2.1; e
4. as coordenadas de 121 pontos da face.

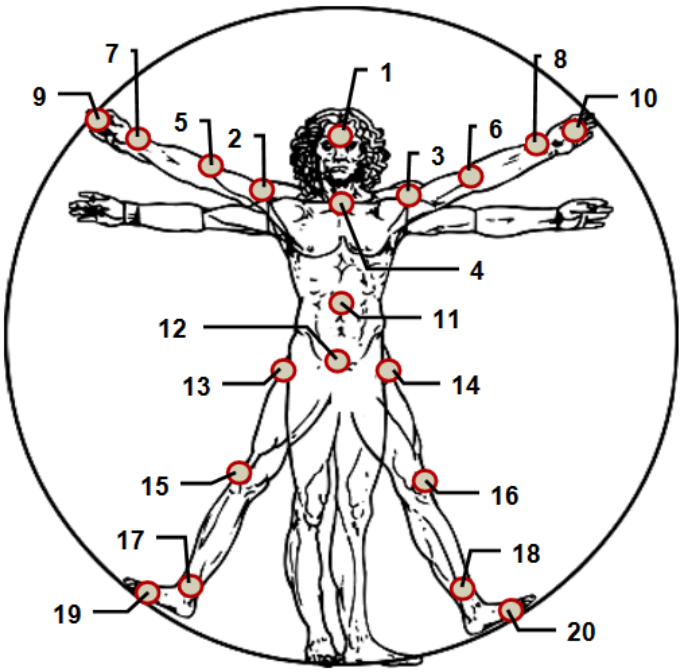


Figura 2.1: Pontos do corpo humano capturados pelo *Kinect*.
Fonte: Almeida (2014)

[5] https://developer.microsoft.com/en-us/windows/kinect

2.2.2 *Software nuiCaptureAnalyse*

O *nuiCaptureAnalyse*[6] é o *software* que opera o *Kinect*. Desenvolvido pela *Cadavid Concepts*, este *software* grava, simultaneamente, as informações disponibilizadas pelo sensor, as coordenadas de 121 pontos do rosto, como ilustra a figura 2.2, e diversas informações em formatos específicos de *softwares* proprietários, tais como .AVI, .BVH, .EXR, .MAT. O *nuiCaptureAnalyse* utiliza como plataforma o sistema operacional *Windows* 7.

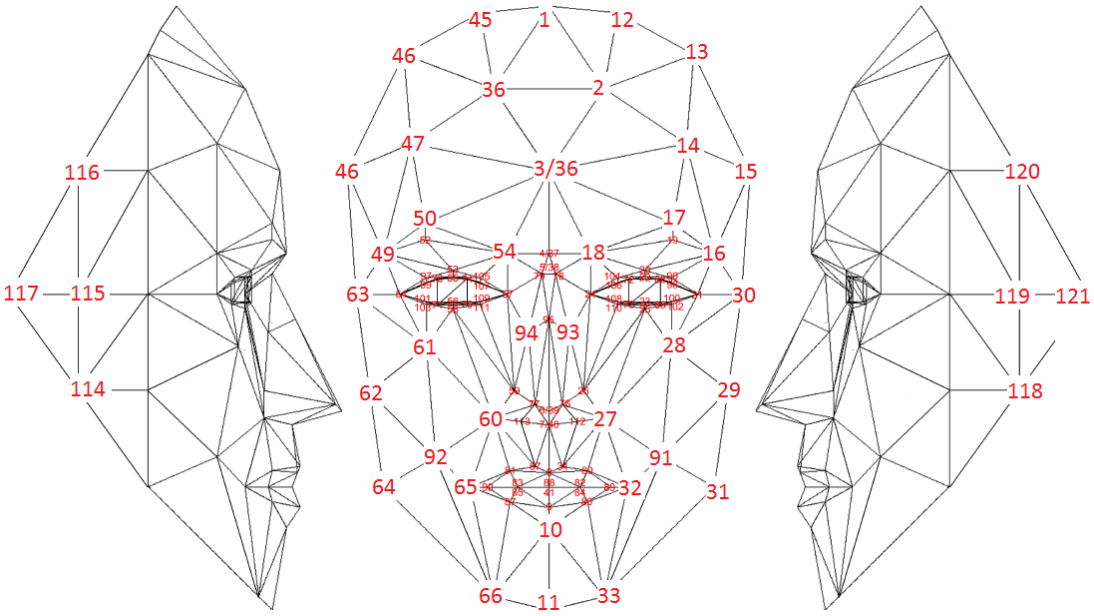


Figura 2.2: 121 pontos da face (com alguns pontos com destaque).
Fonte: CadavidConcepts

2.2.3 Manipulação dos Dados

Para a manipulação dos dados obtidos nas gravações foram utilizadas duas ferramentas para simulação e análise matemática: Matlab R2016b[7] e RStudio[8].

O Matlab é um programa fabricado pela *MathWorks* muito utilizado nas áreas de engenharia e computação, que através de *toolboxes* (conjunto de funções) possui recursos para o aprendizado de máquina, processamento de sinais, processamento de imagem, visão computacional, comunicações, finanças computacionais, design controle, robótica, entre muitos outros.

O RStudio é um ambiente de desenvolvimento integrado (IDE) para a linguagem R. Apesar de ser uma ferramenta *open-source* idealizada para cálculos estatísticos e gráficos, com o passar dos anos a comunidade desenvolveu vários pacotes com recursos variados como nos *toolboxes* do Matlab.

[6] http://nuicapture.com/
[7] http://www.mathworks.com/products/matlab/whatsnew.html?s_tid=tb_16b
[8] https://www.rstudio.com/products/rstudio/

2.3 Protocolo de Gravação

Criar um protocolo de gravação de uma base de dados não é uma tarefa simples. Diante do aparato físico que se tinha acesso, buscou-se padronizar a gravação de todas as amostras (vídeos dos sinais) para que a metodologia proposta não seja enviesada por possíveis diferenças nas gravações. Com isso, todas as gravações dos sinais seguiram os requisitos descritos:

- a posição do sinalizador (aquele que executa os sinais em Libras) é fixa (sentado em uma cadeira);
- a distância entre o sinalizador e sensor é fixa ($\approx$ 1,2 metros) e suficiente para capturar os movimentos do rosto;
- as posições relativas às pernas não foram gravadas;
- todos os sinais foram gravados por uma mesma pessoa (sinalizador) em um mesmo local;
- cada sinal foi gravado 10 vezes; e
- o sinalizador inicia e finaliza o sinal com as mãos sobre as pernas e expressão facial neutra.

Dessa forma, o primeiro passo para a criação da base foi a escolha dos sinais. Com a ajuda de um intérprete da língua, optou-se pelos sinais - Acalmar, Acusar, Aniquilar, Apaixonado, Engordar, Felicidade, Magro, Sortudo, Surpresa e Zangado - cuja expressão facial se alterava ao longo de sua execução. A figura 2.3 ilustra todos estes sinais e no apêndice B encontra-se um descritivo de como executar cada sinal, de acordo com Capovilla et al. (2012a) e Capovilla et al. (2012b).

Em seguida os sinais foram gravados. Um cenário foi montado de forma que o sinalizador ficava em uma posição fixa (sentado em uma cadeira) à uma distância de aproximadamente 1,2 metros do capturador (sensor RGB-D), como ilustrado na figura 2.4. Esta configuração foi adotada, porque a região das pernas não compõe a região de interesse. Além da iluminação já existente no ambiente, lâmpadas de LED foram inseridas ao lado do sensor com o foco voltado para a face do sinalizador e o tecido *Chroma Key* atrás do sinalizador também compunha o cenário.

A base foi gravada por apenas uma pessoa, pois o objetivo do trabalho não é diferenciar as formas de execução dos sinais por diferentes sinalizadores. Os limites dessa pesquisa estão em desenvolver uma metodologia capaz de diferenciá-los entre si.

2.4 A Base de Sinais

Cada um dos 10 sinais foram executados 10 vezes por um único sinalizador, totalizando 100 amostras para a composição de uma base de dados balanceada. O número de quadros de cada amostra varia em cada gravação, como mostra a tabela 2.1. Esta variação se deve

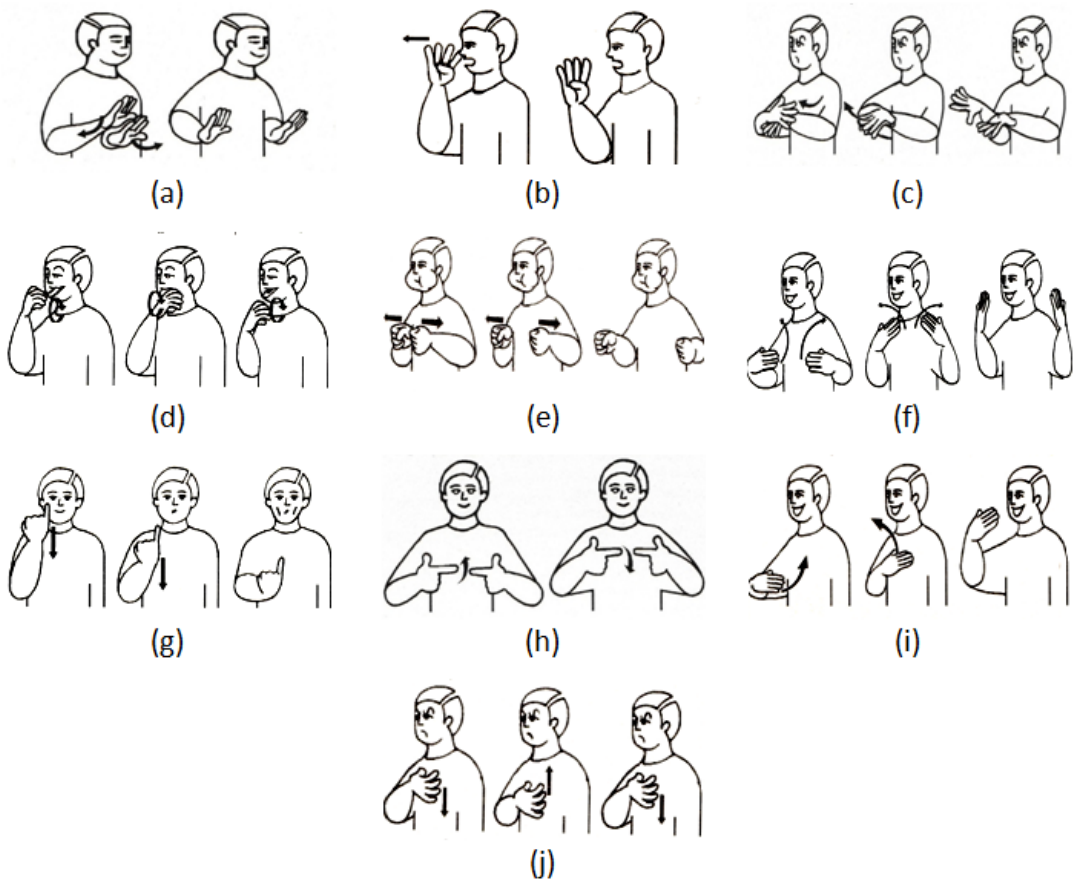


Figura 2.3: Sinais: (a)Acalmar, (b)Acusar, (c)Aniquilar, (d)Apaixonado, (e)Engordar, (f)Felicidade, (g)Magro, (h)Sortudo, (i)Surpresa e (j)Zangado.
Fonte: Capovilla et al. (2012a) e Capovilla et al. (2012b)

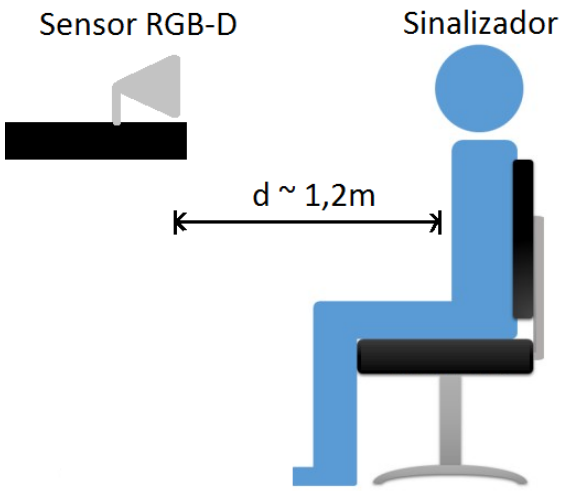


Figura 2.4: Cenário criado para a gravação dos sinais (*indoor*).
Fonte: Rezende et al. (2016) (Adaptado)

a velocidade com que o sinal é gravado e estes valores são aceitáveis quando se compara com o valor médio de quadros do sinal em questão. Dessa forma, esta base disponibiliza os quadros que compõem o vídeo de cada sinal, gravados a uma taxa de 30 quadros por

segundo, e uma matriz com as coordenadas x-y dos 121 pontos do rosto (Figura 2.2), como ilustrado na figura 2.5. Para exemplificar uma amostra da base de dados, a figura 2.6 apresenta os quadros que compõem uma das gravações.

Tabela 2.1: Número de quadros de cada sinal em cada uma das 10 gravações.

Sinal	**Número de quadros em cada gravação G_n (n=1 a 10)**										**Valor médio**
	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_8	G_{10}	
Acalmar	90	87	86	90	112	91	125	95	100	102	98
Acusar	49	48	51	50	45	52	50	60	74	59	54
Aniquilar	64	58	63	62	78	69	74	68	73	82	61
Apaixonado	92	92	68	85	100	100	97	112	127	94	97
Engordar	50	62	37	67	81	77	87	71	72	75	70
Felicidade	81	76	88	104	92	88	88	80	81	82	86
Magro	81	107	86	65	86	86	83	86	89	95	86
Sortudo	67	69	73	71	77	80	85	85	86	88	78
Surpresa	50	50	47	52	67	58	86	70	70	69	61
Zangado	61	50	61	67	64	77	74	64	69	69	66

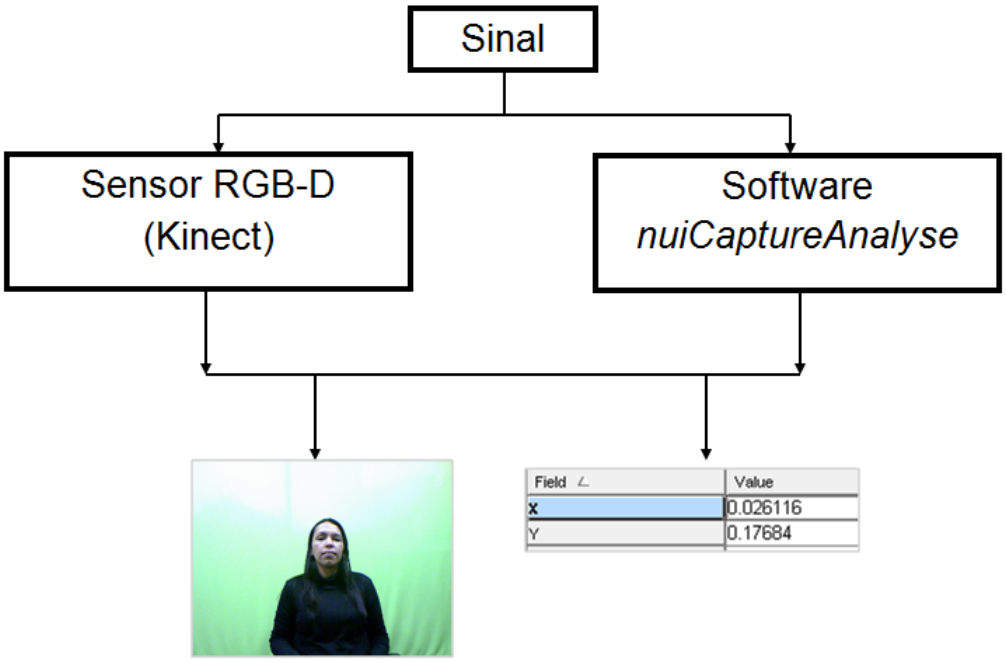


Figura 2.5: Base de dados formada pelos quadros dos vídeos dos sinais e pelas coordenadas dos pontos capturados da face.

Apesar do conjunto de dados ser pequeno, construiu-se uma base com sinais bem diversos e outros bem similares, aproximando-se da situação real da língua. Vale ressaltar que, dentro dos recursos disponíveis, a preocupação maior foi em relação ao número de gravações de cada sinal.

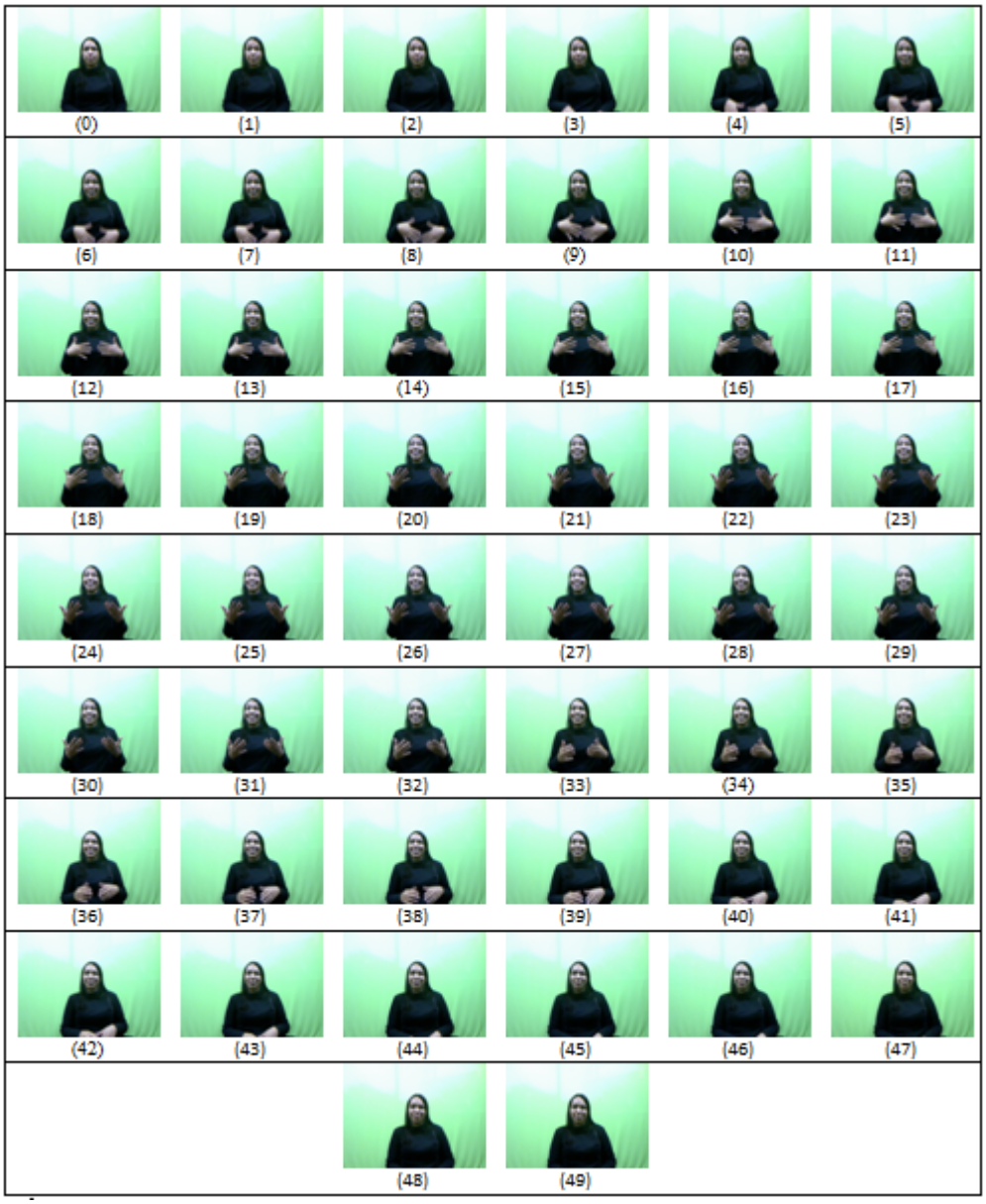


Figura 2.6: 50 quadros da 2^a gravação do sinal Surpresa.

2.5 Conclusões

A criação da base de dados foi uma etapa muito importante do trabalho. A combinação de todas as ferramentas utilizadas levaram a um conjunto de dados que proporcionasse a validação da metodologia proposta. Vale ressaltar que há tecnologias mais recentes, no entanto, o foco do trabalho não é fazer este estudo. Além disso, o baixo número de amostras é um fator a ser melhorado em um estudo futuro.

Outro ponto importante a ser destacado é o passo-a-passo para a execução de cada sinal descrito no apêndice B. Neste descritivo, há sinais que não citam a mudança na expressão facial, no entanto, em todas as palavras escolhidas há um sentimento por trás do seu significado e a língua torna possível a demonstração deste sentimento através da expressão, dando ênfase ao sinal executado.

Capítulo 3

Extração de Características

Sumário

3.1 Introdução

A base experimental de sinais de Libras disponibiliza os quadros dos vídeos das gravações dos sinais e as coordenadas cartesianas de um modelo da face. Neste modelo, 121 pontos são marcados em posições como lábios, olhos, sobrancelhas, testa, dentre outros, como apresentado na figura 3.1. Para que estes dados sejam utilizados em um sistema de classificação é necessário que eles estejam dispostos em uma estrutura passível de interpretação pelo classificador e quanto mais representativa, melhor será o desempenho da classificação. A partir dos dados que foram disponibilizados na base experimental, optou-se por descrever os sinais utilizando ambas as informações, quais sejam, imagens e pontos cartesianos, e fazer uma análise do desempenho delas individualmente.

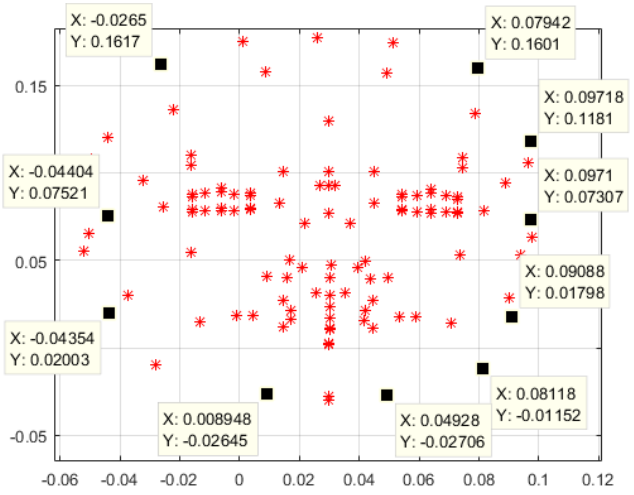


Figura 3.1: Modelo da face com as 121 coordenadas cartesianas.

Nesse trabalho, a extração de características foi realizada após a detecção da região de interesse que, no problema testado, é a face. Um vetor de características foi construído a partir das coordenadas cartesianas e outro a partir da aplicação do descritor de textura LBP (Padrões Locais Binários). Vale ressaltar também que as características foram extraídas dos sinais já sumarizados, ou seja, escolheu-se alguns quadros para representar cada sinal. Todas estas etapas são descritas detalhadamente no Capítulo 4.

3.2 Extração de Características

Embora para o ser humano o reconhecimento de objetos e pessoas seja uma atividade de fácil aprendizado, quando se trata de Visão Computacional, esta tarefa não é trivial. Neste caso é necessário encontrar uma técnica computacional que faça a interpretação ou entendimento de uma cena e retorne características que a represente.

De acordo com Jain et al. (2000), um método de extração de características determina um subespaço apropriado de dimensionalidade m, a partir de um espaço de dimensionalidade d, que representa a cena, sendo $m \leq d$. Em outras palavras, a extração de características cria elementos, quantitativos ou qualitativos, a partir de transformações das características originais do objeto de estudo (Ramos, 2012). Na área de reconhecimento de padrões o desempenho do classificador está diretamente relacionado com quão discriminantes são as características extraídas o que, consequentemente, está relacionado com o poder do extrator.

Diante da importância da extração de características em um problema de reconhecimento de padrões, as seções 3.2.1 e 3.2.2 apresentam, respectivamente, o vetor de características construído nesse trabalho a partir da característica dos pontos do modelo da face e o vetor de características construído a partir do descritor LBP aplicado à imagem.

3.2.1 Extração de Características: Pontos Cartesianos do Modelo da Face

Na literatura há trabalhos que utilizam informações das coordenadas cartesianas para representar gestos. Valverde et al. (2012) fizeram o reconhecimento dos gestos "vem", "tchau", "helicoidal", "direita" e "frente". Ao capturar o gesto por meio do sensor RGB-D *Kinect*, as coordenadas (x,y,z) da mão de cada um dos quadros do vídeo foram obtidas. Com isso, o vetor de pontos extraído de cada gesto foi convertido em um vetor de características (Valverde et al., 2012). Já a extração de características realizada em Pedroso e Salles (2012) utilizou a técnica AAM que fez a modelagem estatística baseada na forma e textura de objetos similares (Pedroso e Salles, 2012). Neste caso, 68 pontos foram marcados manualmente nas imagens do *dataset* JAFFE, obtendo informações de posições representativas da face, tais como os olhos e a boca. Vários procedimentos foram realizados e juntamente com uma técnica de textura formou-se o modelo estatístico de aparência. Já Freitas et al. (2014b) estruturaram dois vetores de características: um com as distâncias entre 17 coordenadas (x,y) do rosto em cada quadro dos vídeos e com seus respectivos ângulos e a segunda representação foi realizada adicionando a informação da

profundidade, o eixo z, ao final do primeiro vetor.

Assim, como apresentado nos trabalhos de Valverde et al. (2012), Pedroso e Salles (2012) e Freitas et al. (2014b), nesse trabalho optou-se por concatenar os 121 pontos disponíveis na base em um único vetor de características. Vale ressaltar que para cada gravação de cada sinal, apenas 5 quadros significativos de cada vídeo foram utilizados para representar um sinal (etapa de Sumarização) e, consequentemente, apenas as informações destes que foram utilizadas para a composição do vetor de características, obtendo a seguinte estrutura:

$$Vetor = \Big[\underbrace{x_1 \;\; y_1 \;\; \ldots \;\; x_{121} \;\; y_{121}}_{quadro\ 1} \;\; \underbrace{x_1 \;\; y_1 \;\; \ldots \;\; x_{121} \;\; y_{121}}_{quadro\ 2} \;\; \ldots \;\; \underbrace{x_1 \;\; y_1 \;\; \ldots \;\; x_{121} \;\; y_{121}}_{quadro\ 5} \Big]_{1\times 1210}$$

3.2.2 Extração de Características: Descritor LBP

Em relação às imagens RGB existem vários extratores que são aplicáveis aos casos de reconhecimento de face e/ou expressão facial, sejam eles: Filtro de Gabor (Júnior et al., 2016)(Ghosal et al., 2009), HOG (Histograma de Gradientes Orientados) (Chao et al., 2013)(Bastos, 2015) e LBP (Padrões Locais Binários) (Musci et al., 2011)(Shan et al., 2005)(Shan et al., 2009).

Júnior et al. (2016) fizeram o reconhecimento facial buscando robustez à oclusão, à variação de iluminação e uma baixa dimensionalidade do vetor de características. Uma abordagem do filtro Gabor, denominada filtro de Gabor curvo, juntamente com a entropia de forma, representaram as imagens. O filtro de Gabor (Li e Allinson, 2008) é invariante à iluminação, rotação, escala e translação (Júnior et al., 2016).

De acordo com Bastos (2015), o HOG (Dalal e Triggs, 2005) é um descritor utilizado em trabalhos onde deseja-se reconhecer objetos em imagens. É uma ferramenta invariante a rotação que tem sido amplamente utilizada em diferentes problemas de Visão Computacional (Dalal e Triggs, 2005). Tanto o filtro Gabor quanto o HOG geram histogramas e a combinação deles representa a imagem na sua totalidade.

Na aplicação desenvolvida nesse trabalho, utilizou-se o descritor LBP (Padrões Locais Binários, do inglês, *Local Binary Patterns*)(Ojala et al., 1996) para a extração de características das imagens devido a vários fatores apresentados na literatura. Shan et al. (2005) mostraram que ele é robusto e estável quando aplicado a imagens de baixa resolução, Ojala et al. (2002) apresentaram evidências empíricas que esta é uma ferramenta invariante à rotação, além de ser invariante à transformações monotônicas da escala de cinza (Musci et al., 2011). De acordo com Ahonen et al. (2004), as características obtidas com o LBP são representações eficientes para imagens da face.

Entretanto, o principal critério para a escolha do LBP foi a sua não dependência da resolução da imagem. Como este parâmetro não foi um item controlado ao criar a base, buscou-se pela ferramenta que não dependesse deste fator. Vale ressaltar, também, a análise feita em Shan et al. (2009) que comparou o tempo e o custo de memória da sua implementação, no Matlab, para extração de características com LBP e Gabor. A tabela 3.1 mostra estes valores, indicando a baixa dimensionalidade do vetor gerado pela aplicação do LBP e o baixo tempo computacional gasto nesta situação.

Tabela 3.1: Tempo e custo de memória para extração de características aplicando o descritor LBP e o filtro Gabor.

	LBP	**Gabor**
Memória (dimensão da característica)	2478	42.650
Tempo para extração de características	0,03s	30s

Fonte: Shan et al. (2009) (Adaptado)

Em Shan et al. (2005) o LBP é aplicado para extrair características de imagens da base de dados *Cohn-Kanade Database*. Neste trabalho, uma análise entre Gabor e LBP também foi apresentada. O LBP teve um desempenho melhor que o Gabor, explicitando que o LBP necessita de um menor custo computacional. Já Musci et al. (2011) utilizou o LBP para descrever imagens de sensoriamento remoto. A comparação realizada neste caso foi entre o LBP e o descritor *Gray Level Co-ocorrence Matrix* - GLCM (Haralick et al., 1973) baseado em matrizes de co-ocorrência. Novamente, o uso do LBP permitiu o alcance dos melhores índices de acurácia na classificação.

O LBP é aplicado a cada *pixel* relacionado a um conjunto de vizinhos igualmente espaçados e equidistantes do *pixel* de referência (Musci et al., 2011). Dessa forma, cada *pixel* é substituído por um valor binário que é obtido pela comparação de uma matriz quadrada contendo os *pixels* vizinhos, respeitando a seguinte regra:

$$b_{ij} = \begin{cases} 0, & \text{se} \quad p_{ij} \leq p_c, \\ 1, & \text{se} \quad p_{ij} > p_c. \end{cases} \tag{3.1}$$

sendo p_{ij} o valor do *pixel* na posição (i,j) e p_c o valor do *pixel* central.

A figura 3.2 exemplifica uma operação realizada pelo LBP. Em uma janela 3x3 de uma imagem, seus tons de cinza são convertidos para a escala de 0 a 255. A partir desta nova janela de valores, os *pixels* das bordas são comparados com o *pixel* central respeitando a equação 3.1. Por fim, o *pixel* central é substituído pelo valor em decimal correspondente ao vetor binário resultante da aplicação da equação.

O exemplo apresentado mostra uma janela 3x3 da imagem. No entanto, o LBP pode ser estendido para vários tamanhos de janela variando-se o número de vizinhos (P) e o raio (R), para tornar possível a captura de características em estruturas de longa escala (Shan et al., 2005). Exemplos do LBP expandido são apresentados na figura 3.3.

O operador LBP produz 2^P saídas correspondentes a 2^P padrões binários (*bins*) que são formados pelos P *pixels* da configuração de vizinhança, ou seja, 2^P valores binários que serão convertidos para decimal e, consequentemente, substituirão o *pixel* central. Entretanto, há alguns *bins* que contêm mais informações do que outros (Ojala et al., 2002). Portanto, é possível usar apenas um subconjunto dos padrões 2^P para descrever a textura das imagens (Shan et al., 2009). Este subconjunto é chamado de padrões uniformes (Ojala et al., 2002). O LBP é chamado de uniforme quando a palavra binária contém até duas transições de 0 para 1 ou vice versa, considerando a palavra circular, como em 00000000, 001110000 e 11100001. Com isso, o número de *bins* para cada configuração é reduzido de 2^P para (equação 3.2):

$$bins = (P * (P - 1)) + 3 \tag{3.2}$$

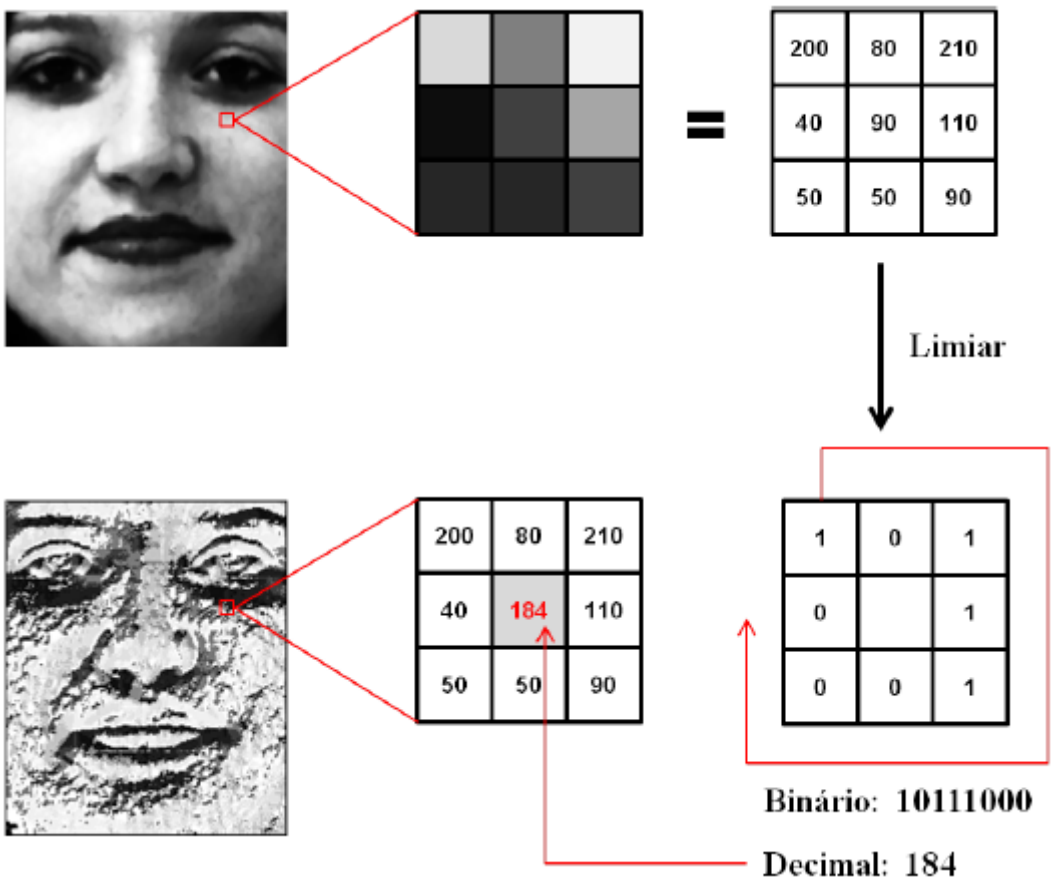


Figura 3.2: Apresentação do descritor LBP aplicado a uma imagem da face. Neste exemplo, a janela possui dimensão 3x3. O valor em decimal, 184, equivalente ao binário 10111000, foi obtido pela comparação dos *pixels* da borda com o *pixels* central.
Fonte: Amaral e Thomaz (2012)

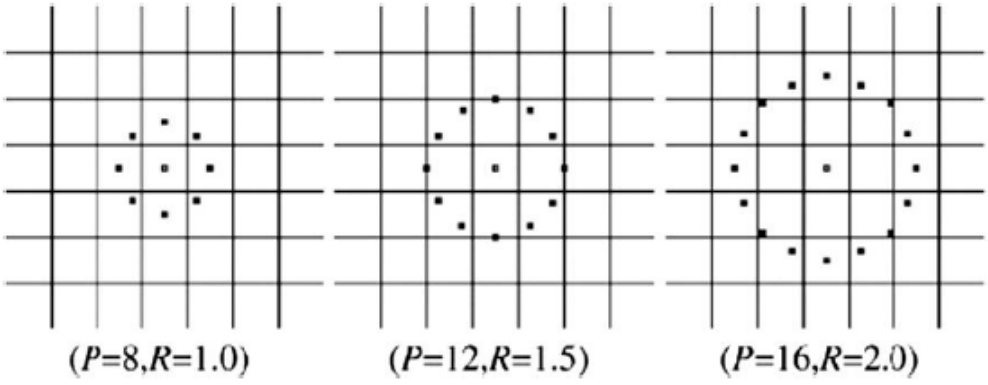


Figura 3.3: LBP expandido.
Fonte: Shan et al. (2009)

Ou seja, o número de *bins*, para 8 *pixels* vizinhos, é $2^P = 2^8 = 256$ no LBP padrão e $(P * (P-1)) + 3 = (8 * (8-1)) + 3 = 59$ no LBP uniforme, como mostra a figura 3.4. Cada uma das barras na figura indicam o número de ocorrência, na imagem, de cada um dos números binários uniformes. Para descrever os padrões uniformes, representado por "$u2$", para uma vizinhança de P *pixels* e raio R, utiliza-se a representação: $LBP^{u2}_{P,R}$.

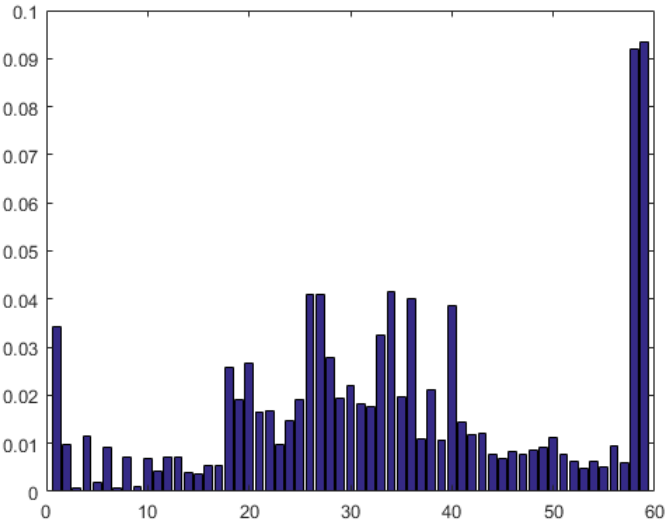


Figura 3.4: 59 bins obtidos pela aplicação do operador LBP uniforme em uma imagem.

Para aplicar o LBP, a imagem pode ou não ser dividida e para cada região será obtido um histograma LBP que contém a informação daquela região. O histograma é composto pela ocorrência de cada um dos padrões uniformes como, por exemplo, quantas vezes o padrão 001110000 foi computado naquela região. Por fim, os histogramas regionais são concatenados para a construção da representação global da imagem da face (Ahonen et al., 2004), como exemplificado na figura 3.5.

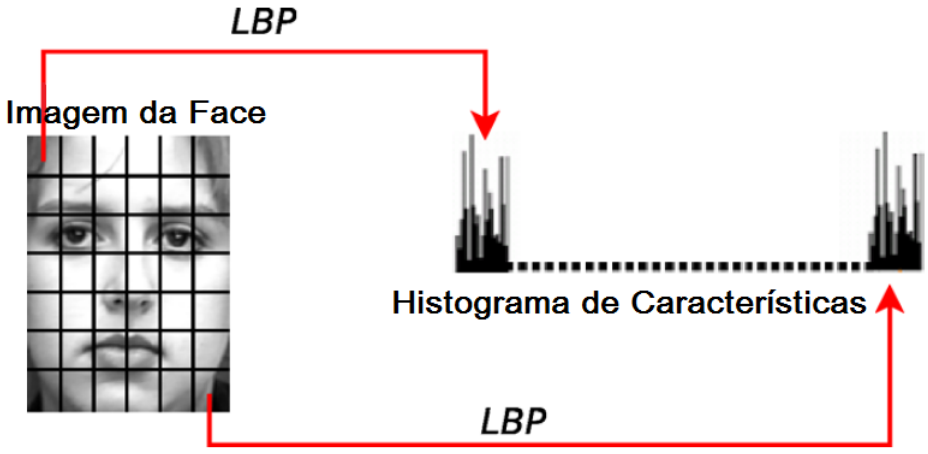


Figura 3.5: Imagem dividida em regiões de onde histogramas LBP são extraídos e concatenados em histograma geral.
Fonte: Shan et al. (2009) (Adaptado)

Alguns parâmetros podem ser otimizados para melhorar a eficácia da extração de características escolhida. Nesse trabalho foi utilizado o LBP com padrões binários, variando o número de vizinhos (8 ou 12) e o raio (1 ou 2), resultando nos seguintes operadores: $LBP^{u2}_{8,1}$, $LBP^{u2}_{8,2}$, $LBP^{u2}_{12,1}$ e $LBP^{u2}_{12,2}$.

3.3 Conclusão

Esse capítulo descreveu como foi realizada a extração de características das informações disponibilizadas pela base de dados experimental. Para os dois casos, pontos e imagens, a construção do vetor de características é a concatenação das características extraídas, formando vetores que contêm informações distintas e que, consequentemente, geraram vetores com tamanhos distintos.

Em relação aos pontos, as informações extraídas fornecem dados relativos a forma e posição. Já em relação às imagens, tem-se informação da análise da textura para cada quadro. O desempenho do classificador depende de quão representativa e distintas são estas informações, e do tamanho do vetor de características que elas geraram. Há classificadores que tem uma performance reduzida quando o vetor de características tem alta dimensão e este é um ponto a ser analisado nesse trabalho.

As quatro configurações do LBP testadas, $LBP^{u2}_{8,1}$, $LBP^{u2}_{8,2}$, $LBP^{u2}_{12,1}$ e $LBP^{u2}_{12,2}$, tiveram como base o trabalho de Shan et al. (2009) que aplicou o operador $LBP^{u2}_{8,2}$. Optou-se, então, por realizar uma variação no raio e no número de vizinhos para que fosse possível testar, empiricamente, o desempenho dos classificadores e, consequentemente, obter a configuração mais representativa para as imagens. Ao final deste estudo, tem-se a configuração relativa a um bom *trade-off* entre a taxa de acerto na classificação e o tamanho do vetor de características.

Capítulo 4

Metodologia

Sumário

4.1 Introdução

Após a criação da base de dados, conforme descrito no Capítulo 2, um modelo para a classificação dos sinais foi definido. Teve-se como ponto de partida as etapas básicas do problema de reconhecimento de padrões, descritas na seção 1.3, e a metodologia aplicada em Almeida (2014).

De posse do conjunto de dados, no caso desse trabalho, detectou-se a face que é a região de interesse. Em seguida, viu-se a necessidade de compactar a quantidade de quadros que compunha cada sinal, pois como mostrado na tabela 2.1, para cada sinal o número de quadros é variável por causa do tempo de execução de cada vídeo. Aplicou-se, então, a ferramenta de sumarização descrita em Almeida et al. (2015) que elimina os quadros redundantes. Isto é importante, uma vez que as características extraídas compõem um vetor que terá o mesmo tamanho para qualquer sinal. O vetor de características é quem descreve o sinal e é a entrada para as técnicas de classificação utilizadas nesse trabalho: o k-NN (k vizinhos mais próximos) e o SVM (Máquinas de vetores de suporte). A figura 4.1 mostra um fluxograma com as etapas realizadas.

Esta metodologia foi estruturada para ser o mais geral possível e permitir a validação de qualquer sistema para classificação envolvendo a expressão facial.

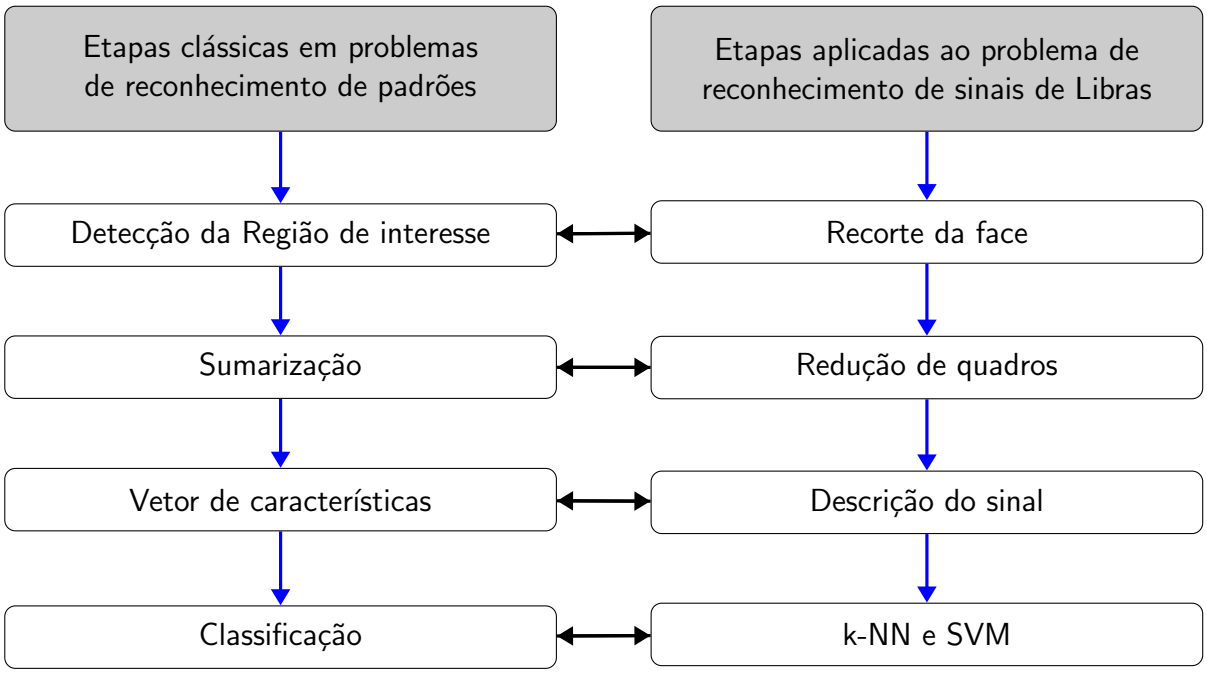


Figura 4.1: Metodologia para o reconhecimento automático de sinais de Libras.

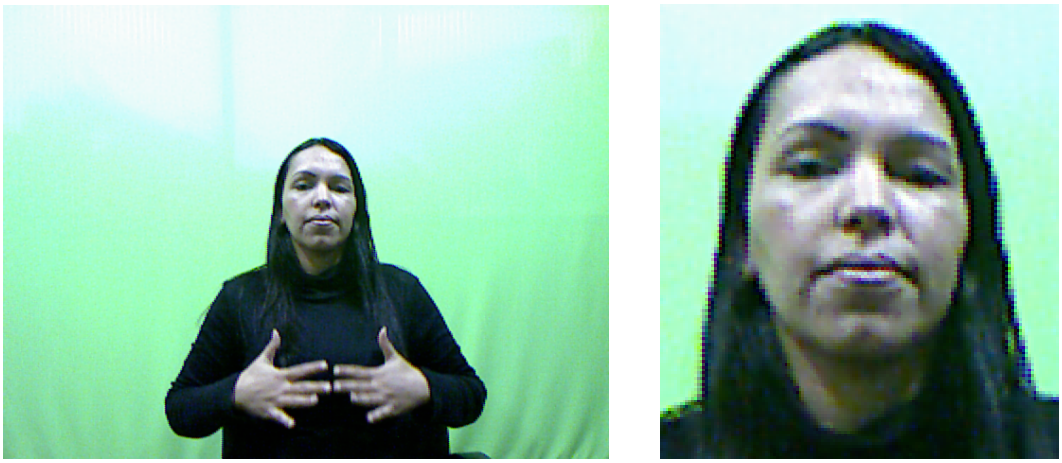
4.2 Detecção da Região de Interesse

A detecção da região de interesse ou segmentação (Carneiro et al., 2009)(Koroishi e Silva, 2015)(Almeida, 2014)(Gonçalves et al., 2016)(Bastos, 2015)(Santos et al., 2015)(Almeida et al., 2013)(Almeida et al., 2014) consiste em isolar o objeto de interesse em uma imagem. Segundo Carneiro et al. (2009), esta etapa é crítica, pois todas as etapas posteriores à segmentação dependem do seu resultado.

Como o objetivo do trabalho é a classificação de um sinal por meio de características presentes na expressão facial, o rosto é a região de interesse. O recorte do rosto na imagem foi realizado tendo como referência o pixel central do quadro, pois a distância do sinalizador ao sensor é fixa. Dessa forma, o algoritmo desenvolvido recebe os vídeos da base de dados, quadro a quadro, e retorna um vídeo com as imagens do rosto recortado a uma taxa de 30 quadros por segundo.

Para que toda a face seja incluída na imagem, o valor empírico do tamanho do quadro foi de 141x161 *pixels*. As figuras 4.2a e 4.2b mostram um quadro completo e a região de interesse detectada, respectivamente.

Há na literatura muitos algoritmos que fazem a detecção do rosto automaticamente, como o Algoritmo de Viola-Jones (Diniz et al., 2016)(Sousa et al., 2016). No entanto, a ferramenta aqui utilizada para o recorte da região de interesse já cumpriu o objetivo da etapa.



(a) Quadro completo (680 x 480 *pixels*) (b) Face detectada (141 x 161 *pixels*)

Figura 4.2: Recorte da região de interesse. Sinal Surpresa.
Fonte: Rezende et al. (2016)

4.3 Sumarização

A etapa de sumarização dos vídeos dos sinais é essencial para o trabalho devido a redução do custo computacional e por eliminar quadros com informações redundantes, tornando o processamento e a extração de características mais eficiente. A principal função da sumarização consiste em reduzir o tamanho dos vídeos com o mínimo de perda de informação possível (Almeida, 2014). Há várias técnicas de sumarização encontradas na literatura. Nesse trabalho optou-se por utilizar uma abordagem do problema clássico de otimização conhecido como Problema da Diversidade Máxima, apresentado em Kuo et al. (1993), utilizado para extrair os quadros mais relevantes em um vídeo.

O Problema da Diversidade Máxima é um problema de otimização que consiste em encontrar elementos tais que a diversidade entre eles seja maximizada. Neste caso, calcula-se a diversidade entre o quadro m e o quadro n do vídeo de cada sinal. Seu cálculo é baseado na distância temporal e na diferença de cores RGB entre os quadros. Para resolver este problema de otimização, utilizou-se o algoritmo desenvolvido em Freitas et al. (2014a) e Almeida et al. (2015) que implementaram uma solução empregando a estratégia evolutiva denominada MSES (*Memetic Self-Adaptive Evolution Strategies*).

Optou-se por selecionar 5 quadros para representar cada sinal. Verificou-se, através de testes visuais e pelos experimentos realizados em Almeida (2014) e Almeida et al. (2014), que 5 quadros eram suficientes para representar os sinais gravados, além de ser um bom limiar entre a representação e o tamanho do vetor de características. Como exemplo, na figura 4.3 estão os quadros que compõem uma das gravações do sinal Felicidade. Destes 104 quadros foram obtidos os 5 mais significativos, por meio do algoritmo de sumarização. Percebe-se que há realmente muitos quadros praticamente idênticos, justificando, assim, a aplicação dessa etapa. Como esse trabalho trata do parâmetro expressões não-manuais da Libras e as mãos não fizeram parte da região de interesse, a sumarização detectou apenas as mudanças mais significativas na face. No apêndice C encontram-se todos os quadros significativos selecionados de todas as gravações.

Figura 4.3: 104 quadros da 4^a gravação do sinal Felicidade. Os 5 quadros mais diversos como resultado da sumarização aplicada para essa gravação foram: 0 - 22 - 37 - 52 - 89 (em destaque).

4.4 Vetor de Características

O objetivo dessa etapa é obter uma representação robusta de cada sinal. Vários trabalhos realizaram a extração de características, tendo esta etapa como parte fundamental para a classificação, com o intuito de extrair características representativas (Carneiro et al., 2009)(Almeida, 2014)(Bastos, 2015)(Santos et al., 2015).

Para cada um dos 5 quadros retornados da etapa de sumarização, um descritor foi criado e o vetor de características resultante é composto pela concatenação destes 5 descritores.

$$Vetor = \left[\begin{array}{ccccc} D_1 & D_2 & D_3 & D_4 & D_5 \end{array} \right]$$

sendo D_1 o descritor do primeiro quadro significativo, D_2 o descritor do segundo quadro

significativo, até D_5 que é o descritor do quinto quadro significativo.

Como a base de sinais disponibiliza tanto pontos cartesianos quanto imagens, optou-se por usar as duas informações como características e verificar qual é a melhor representação para os dados. A figura 4.4 mostra os 121 pontos cartesianos da face dos 5 quadros significativos da 4^a gravação do sinal Felicidade e a figura 4.5 apresenta as respectivas imagens. Vale ressaltar que a referência dos pontos, $x = 0$ e $y = 0$, é o centro do quadro de gravação, que pode ou não coincidir com algum dos 121 pontos do rosto. Isto pode ser exemplificado pela figura 4.6 que apresenta um quadro significativo e a aproximação da posição dos pontos relativa a ele.

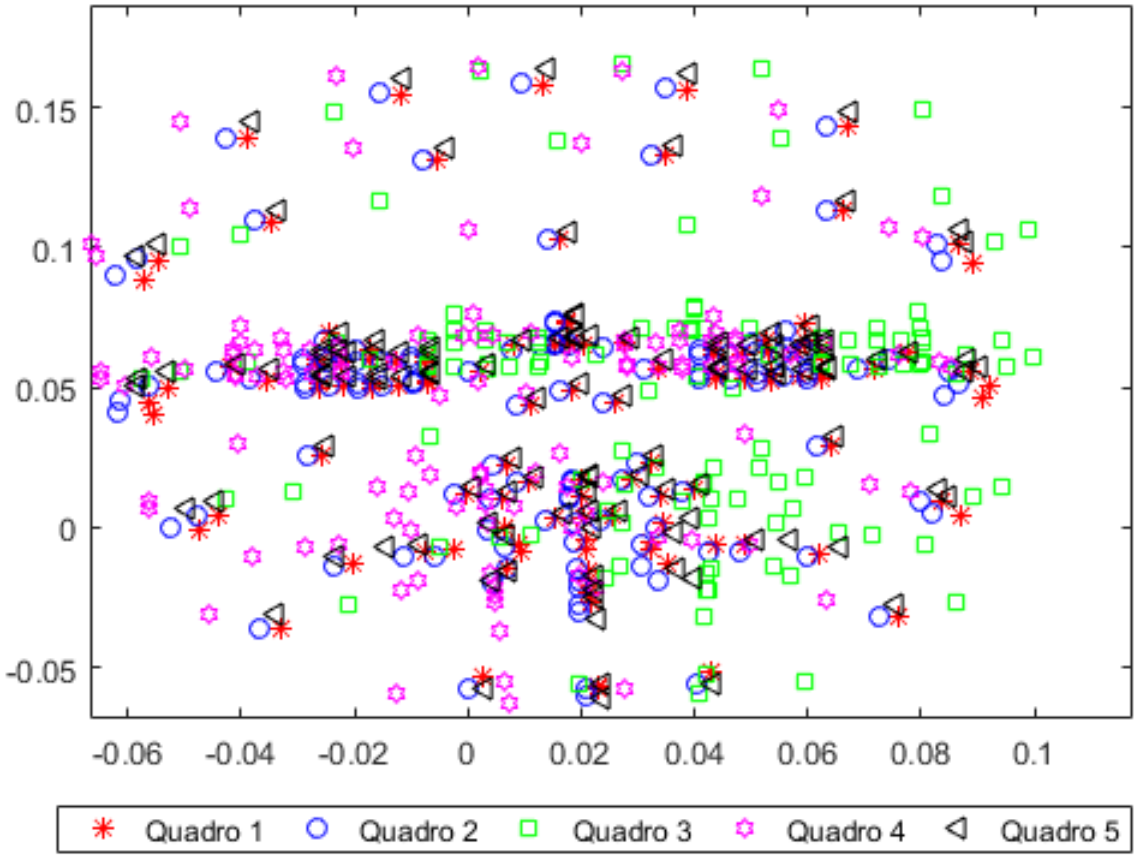


Figura 4.4: 121 pontos dos 5 quadros mais significativos da 4^a gravação do sinal Felicidade.

(a) 0 - Quadro 1 (b) 22 - Quadro 2 (c) 37 - Quadro 3 (d) 52 - Quadro 4 (e) 89 - Quadro 5

Figura 4.5: 5 quadros mais significativos da 4^a gravação do sinal Felicidade.

Figura 4.6: Aproximação dos 121 pontos correspondentes ao 3^o quadro significativo da 3^a gravação do sinal Zangado.

4.4.1 Vetor de Características: Pontos (x,y) da Face

Para descrever cada quadro dos vídeos, as coordenadas (x,y) dos 121 pontos do rosto foram concatenadas, obtendo-se um vetor de 242 posições com a seguinte representação:

$$D = \left[\underbrace{x_1 \quad y_1}_{ponto1} \quad x_2 \quad y_2 \quad x_3 \quad y_3 \quad \cdots \quad \underbrace{x_{121} \quad y_{121}}_{ponto121} \right]_{1x242}$$

A partir da associação dos descritores de cada um dos 5 quadros significativos do sinal (D_1, D_2, D_3, D_4 e D_5) obtêm-se o vetor de características para esta configuração. Neste caso, sua dimensão é 1x1210 e a representação final de cada sinal segue a estrutura:

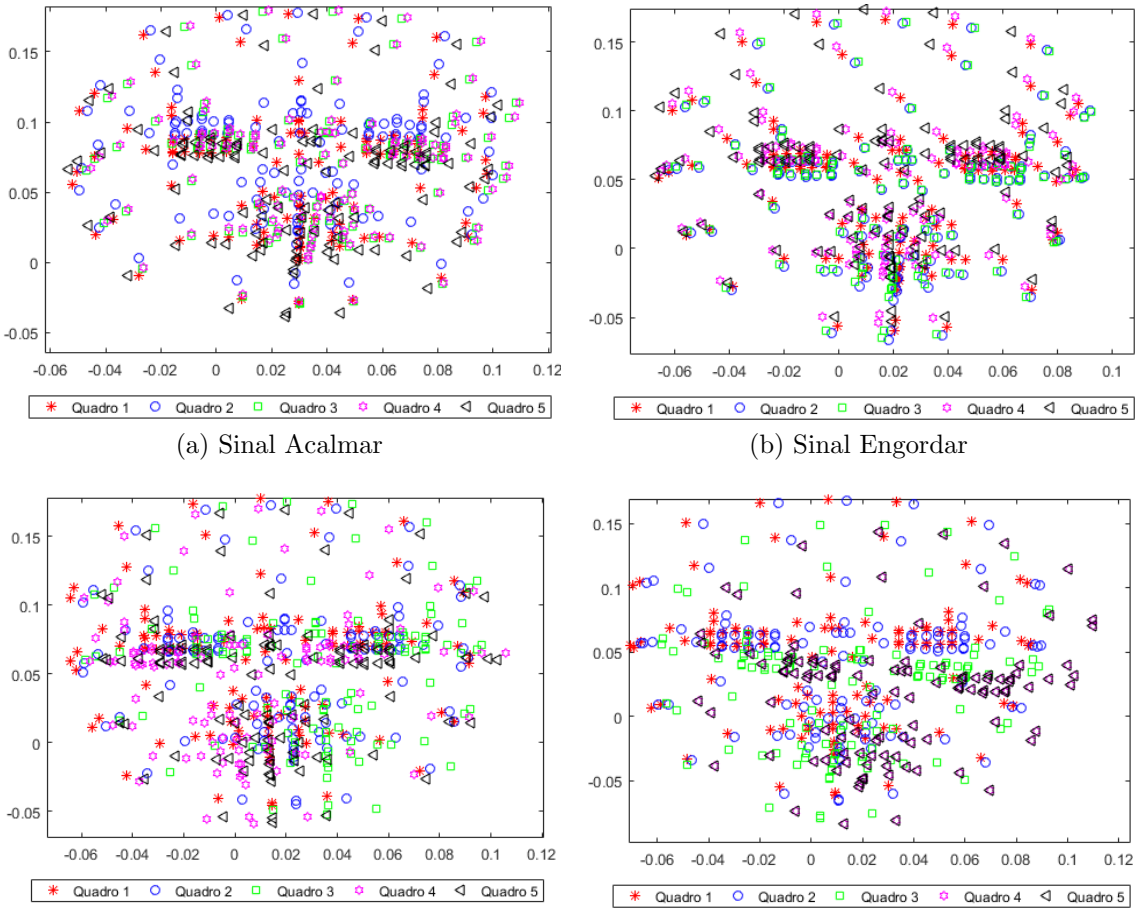
$$Vetor = \left[\; D_1 \quad D_2 \quad D_3 \quad D_4 \quad D_5 \; \right]_{1x1210}$$

Por fim, tem-se um conjunto de dados X (100 amostras e 1210 características) e seus rótulos Y_d, que são as saídas desejadas:

$$X_{100\times1210} = \begin{bmatrix} Vetor_{Acalmar/Gravacao1} \\ \vdots \\ Vetor_{Acalmar/Gravacao10} \\ \vdots \\ \vdots \\ Vetor_{Zangado/Gravacao1} \\ \vdots \\ Vetor_{Zangado/Gravacao10} \end{bmatrix} \qquad Yd_{100\times1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ 10 \\ \vdots \\ 10 \end{bmatrix}$$

Como os pontos (x,y) descrevem a posição das partes da face, eles podem sofrer algum deslocamento, alterando sua posição entre gravações de um mesmo sinal. Assim, procurou-se verificar a partir de 4 experimentos realizados, se a performance do classificador é alterada nestes casos.

- **Primeiro Experimento (EX.1):** implementação da metodologia descrita ao longo do trabalho a partir das informações dos sinais sem tratamento prévio. Desta forma, a classificação dos sinais foi realizada com os dados brutos disponíveis na base de dados como ilustrado na figura 4.7.



(a) Sinal Acalmar (b) Sinal Engordar

(c) Sinal Magro (d) Sinal Zangado

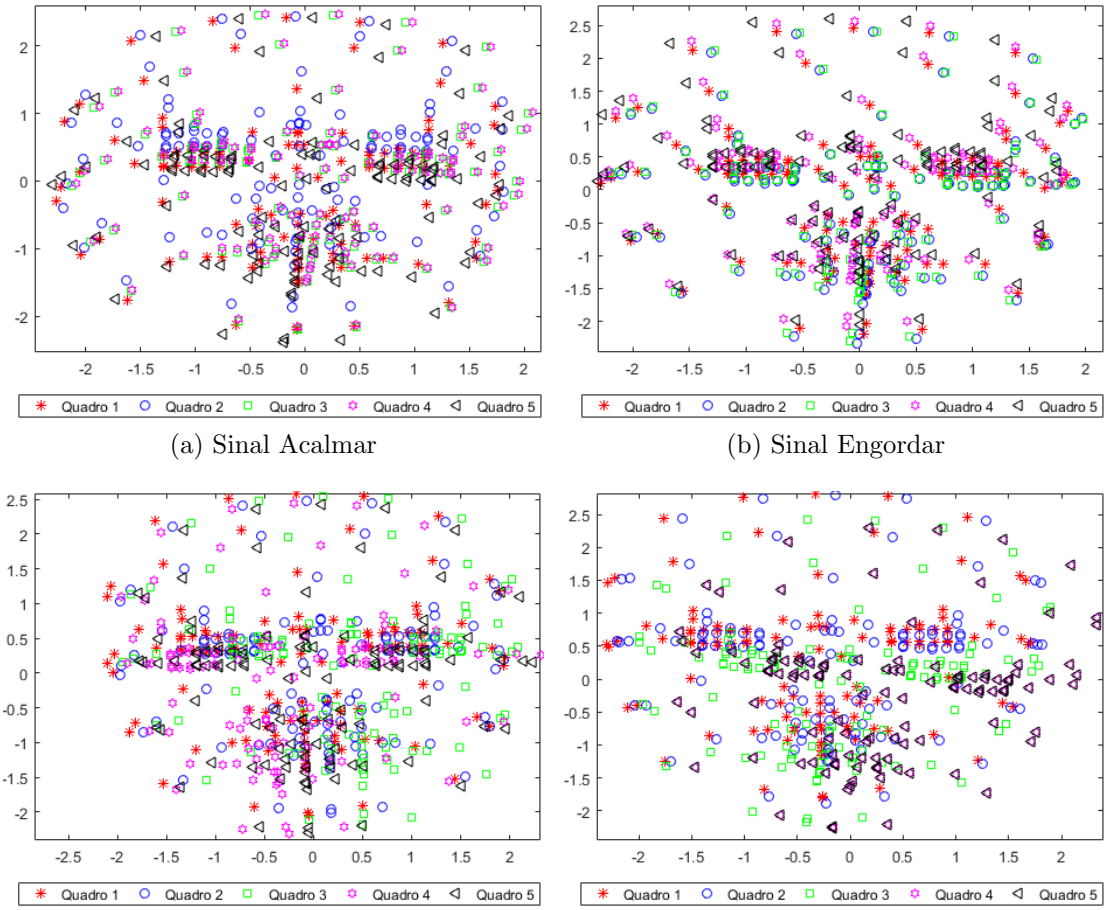
Figura 4.7: 121 pontos originais dos 5 quadros mais significativos da 1^a gravação dos sinais (a) Acalmar, (b) Engordar, (c) Magro e (d) Zangado.

- **Segundo Experimento (EX.2):** há tratamento prévio dos dados relativos aos sinais. Para cada gravação de cada sinal, ou seja, para cada amostra, a Normalização Z foi aplicada em cada coordenada, respeitando as equações 4.1 e 4.2. Dessa forma, a nova distribuição dos pontos tem média 0 e desvio-padrão igual a 1 (Figura 4.8).

$$x_{novo} = \frac{x - \bar{x}}{\sigma(x)} \tag{4.1}$$

$$y_{novo} = \frac{y - \bar{y}}{\sigma(y)} \tag{4.2}$$

sendo:
x e y correspondem às coordenadas dos pontos que serão atualizadas,
$\bar{x}$ e $\bar{y}$ são o valor médio de todos os x e y, e
$\sigma(x)$ e $\sigma(y)$ são o desvio-padrão de x e y daquela gravação, respectivamente.



(a) Sinal Acalmar (b) Sinal Engordar

(c) Sinal Magro (d) Sinal Zangado

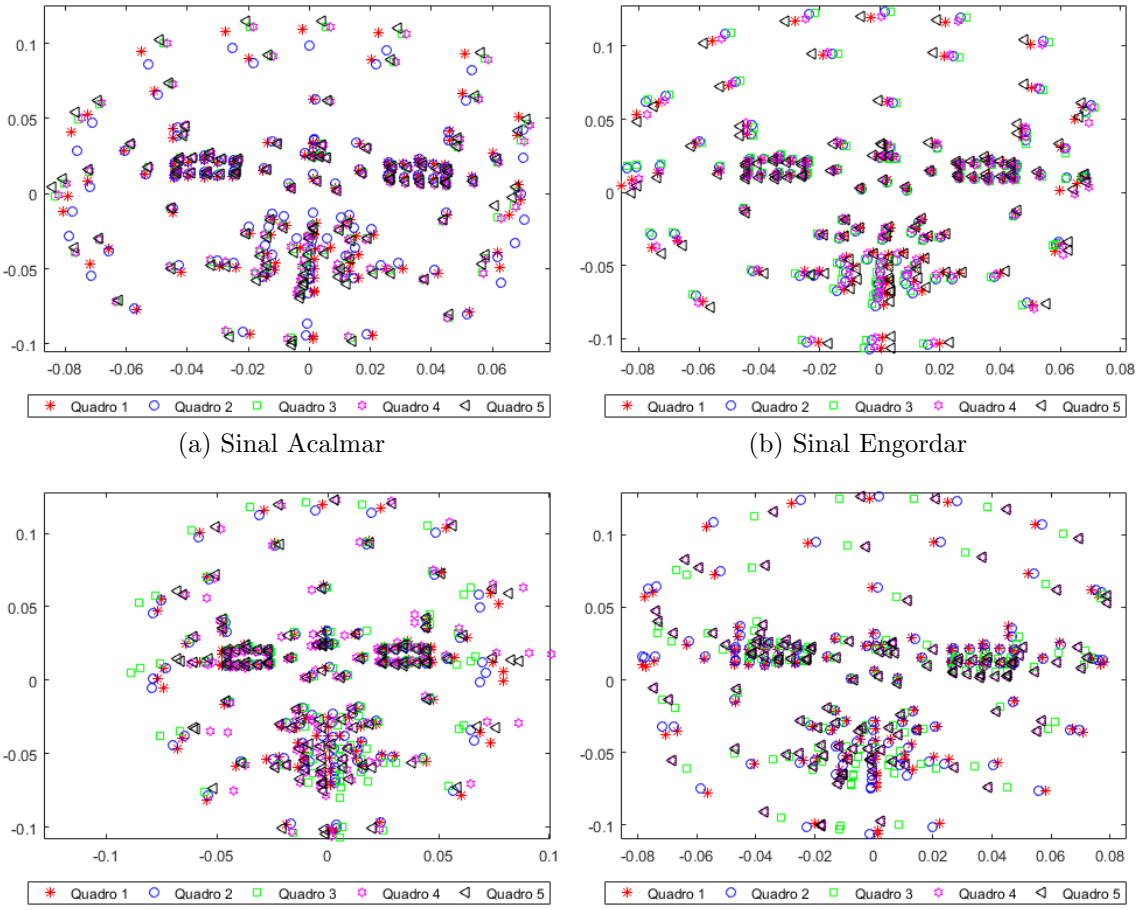
Figura 4.8: 121 pontos do rosto após normalização Z dos 5 quadros mais significativos da 1^a gravação dos sinais (a) Acalmar, (b) Engordar, (c) Magro e (d) Zangado.

- **Terceiro Experimento (EX.3):** os dados de cada quadro foram atualizados de acordo com as equações 4.3 e 4.4. Aplicando a normalização pelo centroide do 1^o quadro, cada ponto teve como referência o ponto médio do 1^o quadro da sua gravação (Figura 4.9).

$$x_{novoPontoP} = x_{pontoP} - \bar{x}_{quadro1^o} \tag{4.3}$$

$$y_{novoPontoP} = y_{pontoP} - \bar{y}_{quadro1^o} \tag{4.4}$$

sendo:
$x_{novoPontoP}$ e $y_{novoPontoP}$ correspondem às coordenadas atualizadas,
x_{pontoP} e y_{pontoP} correspondem às coordenadas do ponto que serão atualizadas, e
$\bar{x}_{quadro1^o}$ e $\bar{y}_{quadro1^o}$ são o valor médio do x e do y no 1^o quadro, respectivamente.



(a) Sinal Acalmar

(b) Sinal Engordar

(c) Sinal Magro

(d) Sinal Zangado

Figura 4.9: 121 pontos do rosto após EX.3 dos 5 quadros mais significativos da 1^a gravação dos sinais (a) Acalmar, (b) Engordar, (c) Magro e (d) Zangado.

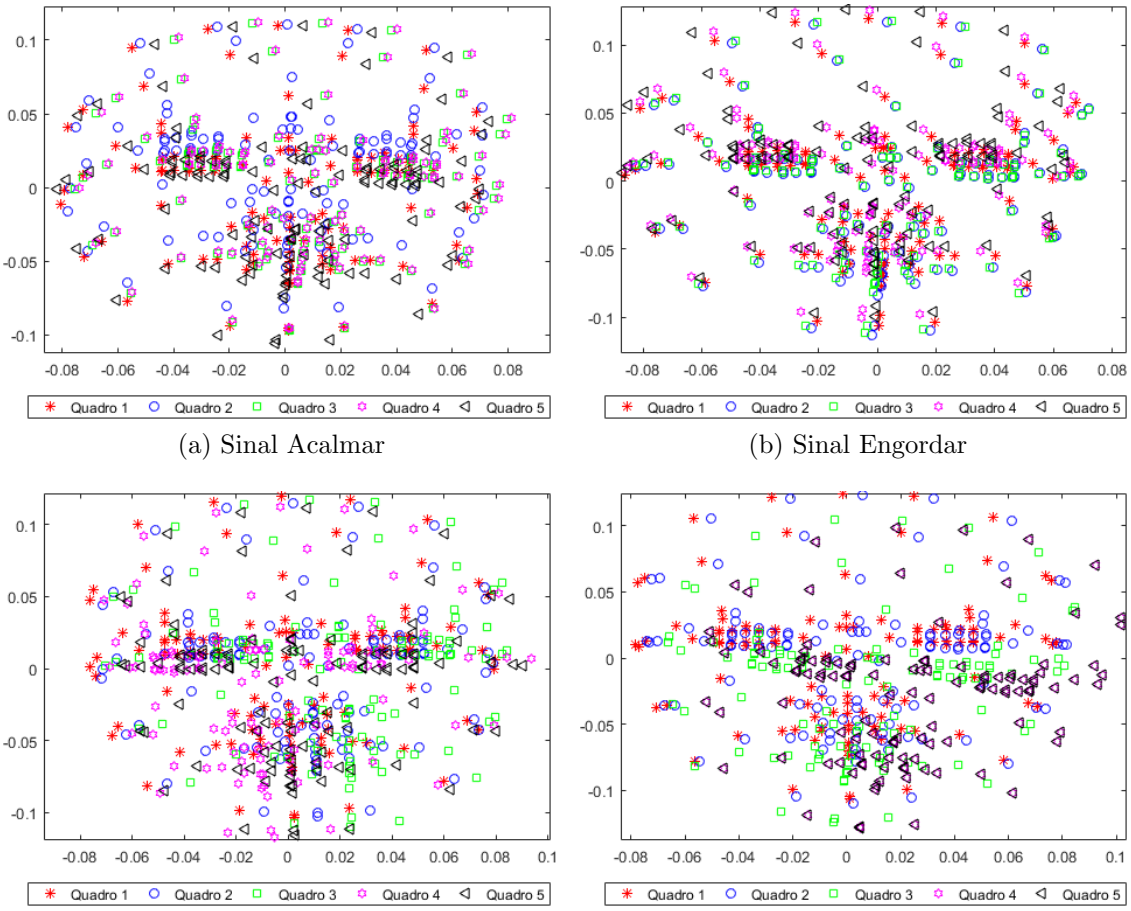
- **Quarto Experimento (EX.4):** os dados de cada quadro foram atualizados de acordo com as equações 4.5 e 4.6. Aplicando a normalização pelo centroide do quadro em questão, cada ponto teve como referência o ponto médio do seu respectivo quadro (Figura 4.10).

$$x_{novoPontoP} = x_{pontoP} - \bar{x}_{quadro} \tag{4.5}$$

$$y_{novoPontoP} = y_{pontoP} - \bar{y}_{quadro} \tag{4.6}$$

onde:
$\bar{x}_{quadro}$ e $\bar{y}_{quadro}$ são o valor médio de x e y no quadro em questão, respectivamente, e $centroide = (\bar{x}_{quadro}, \bar{y}_{quadro})$.

As figuras 4.7 a 4.10 mostram os 4 experimentos aplicados na primeira gravação dos sinais Acalmar, Engordar, Magro e Zangado. Percebe-se que a normalização Z apenas amplia o espaço que o sinal é executado, sendo que os experimentos 2 e 4 dão ênfase à trajetória global do sinal, ou seja, o movimento do rosto. Nos experimentos 2, 3 e 4, os pontos foram centralizados em $x = 0$ e $y = 0$, como apresentado nas figuras 4.8, 4.9 e 4.10. Vale ressaltar que o experimento 3, EX.3, enfatiza a trajetória dos pontos realçando a expressão facial, como ilustra a figura 4.9.



(a) Sinal Acalmar (b) Sinal Engordar

(c) Sinal Magro (d) Sinal Zangado

Figura 4.10: 121 pontos do rosto após EX.4 dos 5 quadros mais significativos da 1^a gravação dos sinais (a) Acalmar, (b) Engordar, (c) Magro e (d) Zangado.

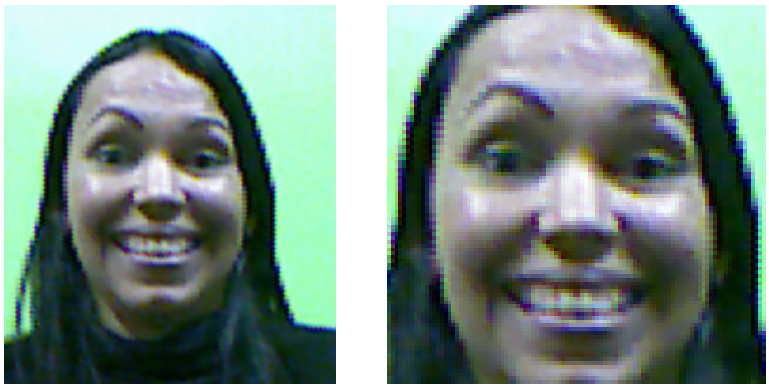
Ao final desta etapa, obtém-se 4 vetores de características diferentes para cada sinal. A hipótese a ser testada é: há alguma representação melhor que a dos dados brutos/originais?

4.4.2 Vetor de Características: Descritor de Textura LBP

De posse dos quadros que compõem os vídeos da base de dados, optou-se por utilizar o descritor de textura LBP, como descrito no Capítulo 3.

Dentre os vários operadores LBP, optou-se por utilizar o $LBP^{u2}_{P,R}$: LBP com padrões uniformes, vizinhança de P *pixels* e raio R. Buscando extrair as características mais representativas das imagens, variou-se o número de vizinhos (8 ou 12), o raio (1 ou 2) e o tamanho da janela/célula. Cada uma das combinações foram aplicadas nas imagens originais recortadas, buscando eliminar ainda mais as informações que não faziam parte da expressão facial, como exemplifica a figura 4.11. Ao final desta etapa, tinham-se 8 diferentes configurações de LBP. Através deste experimento, analisou-se qual seria a combinação que retornaria a melhor acurácia do classificador.

Para recortar as imagens, utilizou-se o algoritmo de Viola-Jones (Viola e Jones, 2004)



(a) Imagem original (b) Recorte do rosto

Figura 4.11: 5^o quadro da 1^a gravação do sinal Felicidade. Imagem (a) 141x161 pixels e (b) 100x100 pixels.

que tenta encontrar na imagem características que codificam a face (Diniz et al., 2016). Dos 500 quadros pertencentes ao conjunto de dados (10 sinais $\times$ 10 gravações $\times$ 5 quadros), o algoritmo teve dificuldade em detectar a face em 3, pois nestes o sinalizador não estava com o rosto diretamente voltado para o vídeo. Foram os quadros:

- 3 da 9^a gravação do sinal Aniquilar;
- 3 da 9^a gravação do sinal Magro; e
- 1 da 1^a gravação do sinal Surpresa.

Optou-se por realizar 3 análises distintas para recordar a face nos três casos que o algoritmo Viola-Jones não foi eficiente:

1. recortar a imagem à mão, como mostrado na figura 4.12, seguindo as mesmas proporções do algoritmo (100 $\times$ 100 *pixels*);

Figura 4.12: Recorte à mão.

2. substituir a imagem pelo quadro mais representativo das 10 gravações daquela posição, figura 4.13; e

Figura 4.13: Quadro substituído.

3. calcular o LBP médio dos demais quadros daquela gravação e atribuir ao quadro em questão, como exemplificado na figura 4.14.

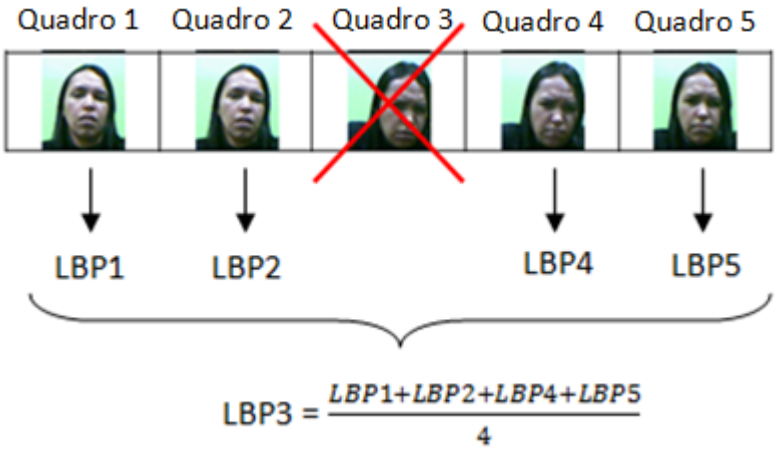


Figura 4.14: LBP médio.

Para cada um dos três quadros que não tiveram a face detectada, as três alternativas

para resolver o problema de recorte da face foram realizadas e as 8 configurações listadas na tabela 4.1 foram executadas para cada análise. Ao final da classificação, tem-se qual a melhor solução para os casos que o algoritmo Viola-Jones falhou. Vale ressaltar que nestes experimentos, utilizaram-se as imagens recortadas e as mesmas imagens particionadas em 16 células, como na figura 4.15. Estas configurações afetam diretamente o tamanho do vetor de características que depende do número de *bins* e do número de células da imagem para cada um dos 5 quadros significativos.

Tabela 4.1: Oito configurações testadas para o vetor de características da imagem recortada.

Configuração do LBP	**Número de *bins* por célula**	**Número de células por imagem (c)**	**Tamanho do vetor de características do sinal (*bins*×c×5)**
$LBP^{u2}_{8,1}$	59	1	295
$LBP^{u2}_{8,2}$	59	1	295
$LBP^{u2}_{12,1}$	135	1	675
$LBP^{u2}_{12,2}$	135	1	675
$LBP^{u2}_{8,1}$	59	16	4720
$LBP^{u2}_{8,2}$	59	16	4720
$LBP^{u2}_{12,1}$	135	16	10800
$LBP^{u2}_{12,2}$	135	16	10800

Figura 4.15: Imagem (em escala de cinza) particionada em 16 células de tamanho [25 25].

A figura 4.16 apresenta um resumo dos vetores de características formados. Buscou-se trabalhar com ambas as informações disponibilizadas pela base de dados: pontos cartesianos e imagens. No caso dos pontos, investigou-se se há alguma forma de representação melhor que a utilização dos dados brutos como vetor de características, e no caso das imagens, investigou-se qual seria a melhor configuração do LBP.

As figuras 4.17 a 4.20 ilustram cada umas quatro configurações do operador LBP aplicado a um dos quadros significativos dos sinais Acalmar, Engordar, Magro e Zangado, após a aplicação do algoritmo Viola-Jones. Existe uma diferença visual entre a estrutura

$$Pontos \rightarrow Concatenando\ as\ coordenadas \rightarrow \begin{cases} Ex.1 \\ Ex.2 \\ Ex.3 \\ Ex.4 \end{cases}$$

$$Imagem\ Recortada \rightarrow \begin{cases} LBP: \ quadro\ inteiro \\ LBP: \ quadro\ particionado \end{cases} \rightarrow \begin{cases} LBP^{u2}_{8,1} \\ LBP^{u2}_{8,2} \\ LBP^{u2}_{12,1} \\ LBP^{u2}_{12,2} \end{cases}$$

Figura 4.16: Diagrama com as variações possíveis para a composição dos vetores de características.

gerada pela aplicação de cada um dos operadores, além do LBP com 12 vizinhos gerar um número maior de *bins* que o LBP com 8 vizinhos, respeitando a equação 3.2 apresentada no Capítulo 3. Para todos os operadores, percebe-se uma diferença entre o tamanho das barras quando uma mesma configuração foi aplicada aos diferentes sinais. Este fato é justificado pela diferença na expressão facial de cada um dos casos e é o fator que diferenciará os sinais.

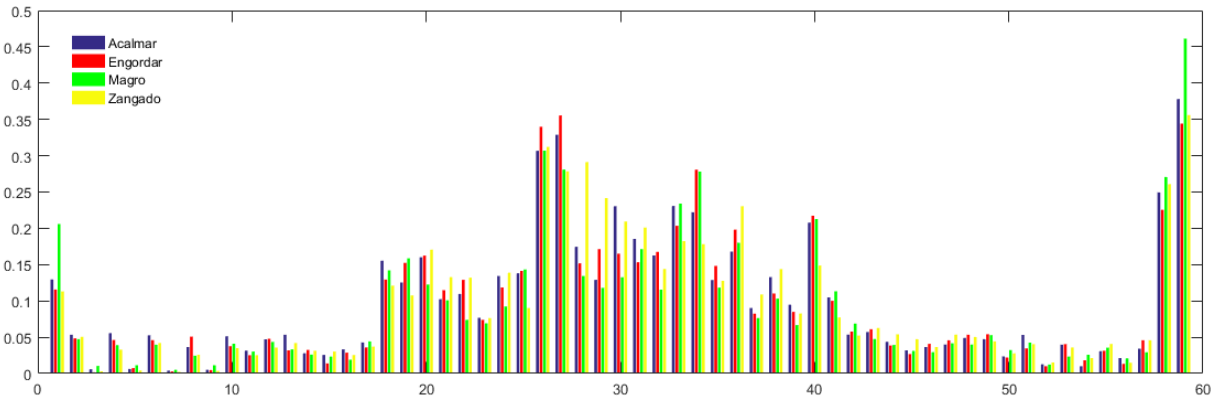


Figura 4.17: $LBP^{u2}_{8,1}$ aplicado ao 3^o quadro significativo da 1^a gravação dos sinais Acalmar, Engordar, Magro e Zangado.

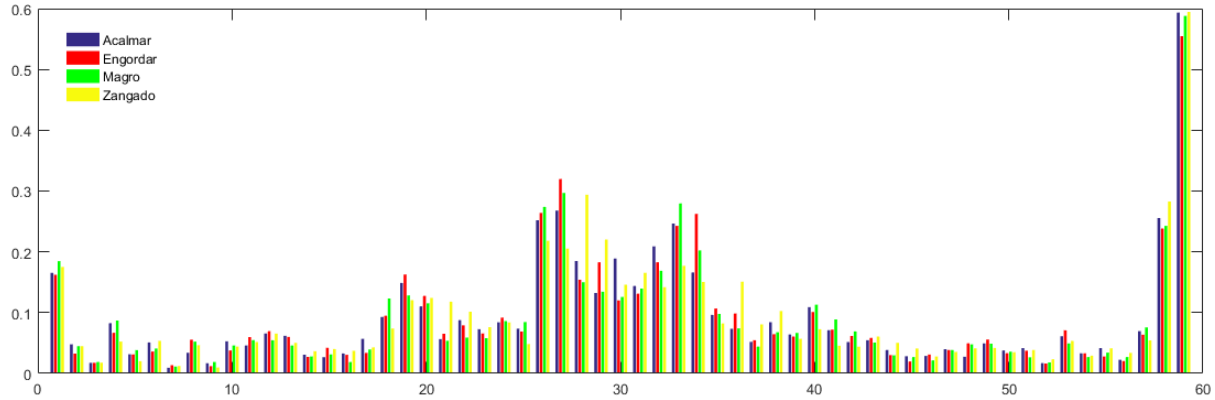


Figura 4.18: $LBP^{u2}_{8,2}$ aplicado ao 3^o quadro significativo da 1^a gravação dos sinais Acalmar, Engordar, Magro e Zangado.

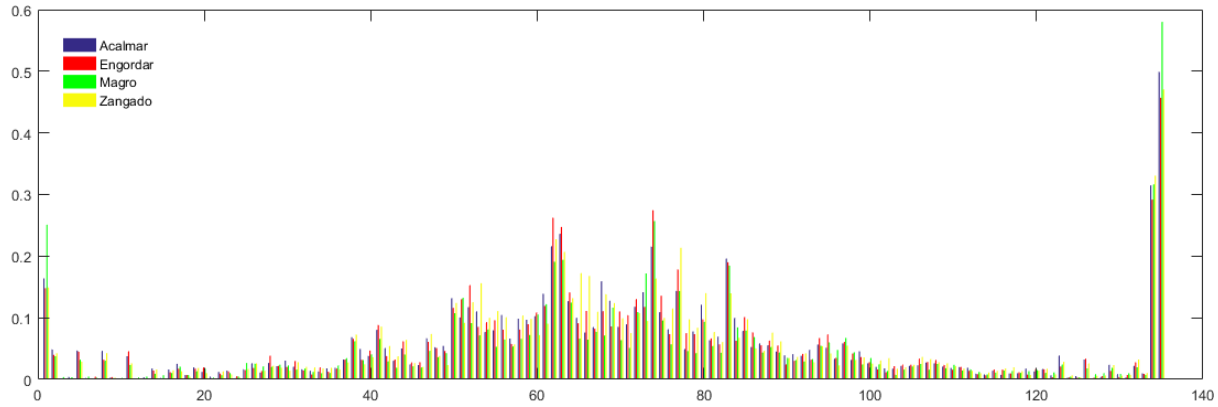


Figura 4.19: $LBP^{u2}_{12,1}$ aplicado ao 3^o quadro significativo da 1^a gravação dos sinais Acalmar, Engordar, Magro e Zangado.

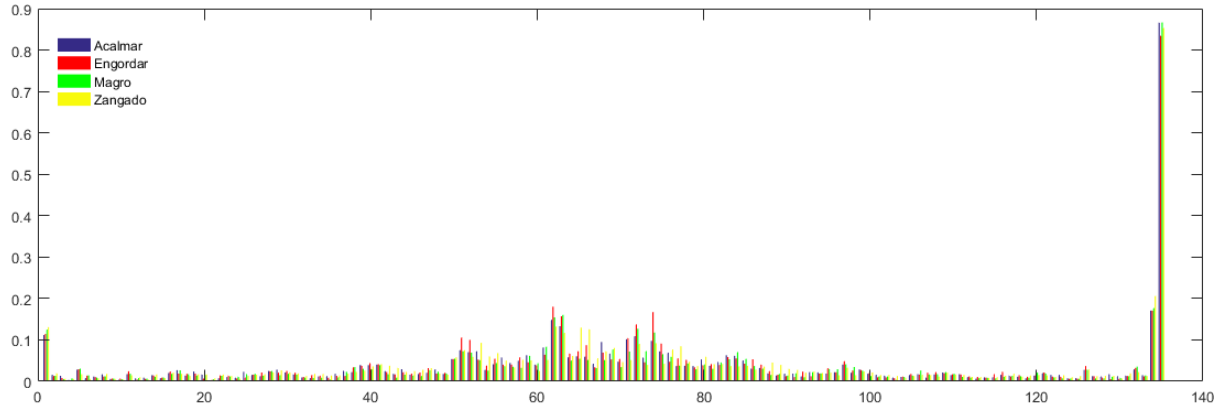


Figura 4.20: $LBP^{u2}_{12,2}$ aplicado ao 3^o quadro significativo da 1^a gravação dos sinais Acalmar, Engordar, Magro e Zangado.

4.5 Classificação

A classificação dos sinais é a etapa final desse trabalho. Como entrada desta etapa, tem-se os vetores de características dispostos da seguinte forma:

$$X_{100\times TAM} = \begin{bmatrix} Vetor_{Acalmar/Gravacao1} \\ : \\ Vetor_{Acalmar/Gravacao10} \\ : \\ : \\ Vetor_{Zangado/Gravacao1} \\ : \\ Vetor_{Zangado/Gravacao10} \end{bmatrix} \qquad Yd_{100\times 1} = \begin{bmatrix} 1 \\ : \\ 1 \\ : \\ : \\ 10 \\ : \\ 10 \end{bmatrix}$$

sendo:
TAM o tamanho do vetor de características variável em cada composição,
$X_{100\times TAM}$ entrada do classificador, e
Y_d são as saídas desejadas.

Na classificação dos sinais foi utilizada a técnica k-NN (k vizinhos mais próximos, do

inglês, k-Nearest Neighbors) por ser um classificador indicado para base de dados que tem poucas amostras. Foi utilizada também a SVM (Máquina de Vetores de Suporte, do inglês, Support Vector Machines) por ser considerada estado da arte em reconhecimento de padrões.

Em ambos os métodos implementados, uma parte dos dados é separada para treinamento e o modelo gerado a partir desses classifica os demais dados pertencentes ao conjunto de teste. Ao final desta etapa, a taxa de acerto da classificação é obtida.

4.5.1 k-NN

O classificador k-NN desenvolvido por Patrick e Fischer (1970) é uma ferramenta muito utilizada para a classificação de padrões (Pedroso e Salles, 2012)(Diniz et al., 2013)(Diniz et al., 2016)(Santos et al., 2015). O método k-NN cria uma superfície de decisão complexa, possibilitando uma maior adaptação à forma de distribuição das amostras do conjunto de treinamento (Pedrini e Schwartz, 2008).

Para determinar a classe de uma amostra m que não pertença ao conjunto de treinamento, o classificador k-NN procura os k-ésimos elementos do conjunto de treinamento que estejam mais próximos de m e atribui a amostra à classe que recebeu o voto majoritário em relação aos k vizinhos mais próximos. A figura 4.21 exemplifica esse método para duas classes. Se k for igual a 3, a amostra "estrela" é classificada como pertencente a classe B, mas se k for igual a 6, a amostra é classificada como pertencente a classe A que recebeu o voto majoritário.

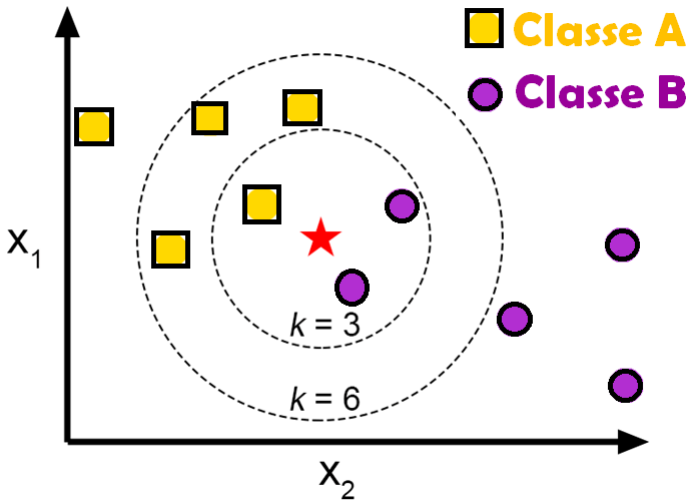


Figura 4.21: Exemplo de classificação utilizando k-NN com 3 e 6 vizinhos mais próximos. Fonte: DeWilde (2012) (Adaptado)

Diante deste conceito, o algoritmo de classificação foi implementado. Inicialmente, as 10 gravações de cada um dos 10 sinais foram aleatorizadas, de forma a impedir que a mesma gravação sempre pertença ou ao grupo de treinamento ou ao grupo de teste. 80% dos dados aleatorizados foram separados para treino e 20% para teste. Dessa forma, o grupo de treinamento possui 8 gravações de cada sinal totalizando 80 amostras e o grupo de teste 2 gravações, totalizando 20 amostras. A divisão 80%-20% foi empírica e teve

como objetivo gerar um modelo com um conjunto de treinamento que representasse de fato toda a amostra e tivesse uma performance satisfatória com os dados de teste.

De posse dos dados de treinamento, foi realizada uma validação cruzada buscando encontrar o valor de k que retornasse a maior taxa de acerto, o k_{best}. Dessa forma, os dados de treinamento foram divididos em 5-*folds* de mesmo tamanho (16 amostras para cada *fold*) e foram realizadas 5 iterações de cruzamento. Respeitando a porcentagem 80%-20%, 1-*fold* foi separado para teste e os 4 restantes para treinamento, conforme a figura 4.22.

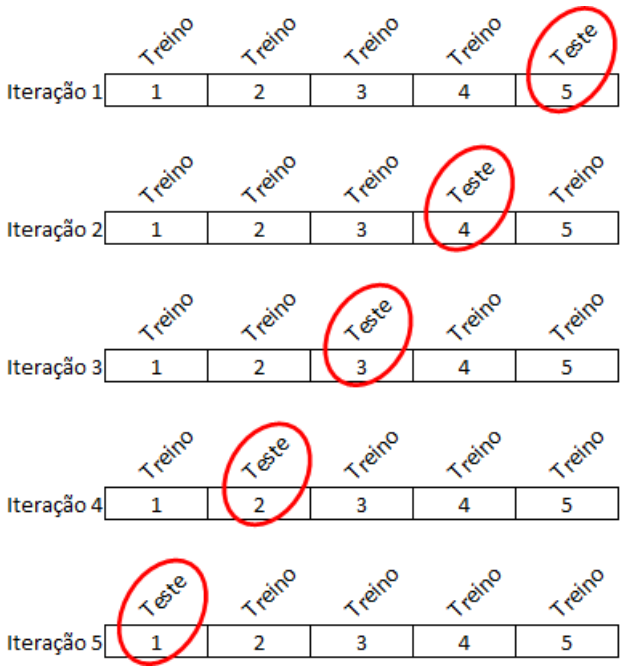


Figura 4.22: Processo para validação cruzada usando 5-*folds*.
Fonte: Rezende et al. (2016) (Adaptado)

Inicialmente, variou-se o k de 1 até 79, que é o valor máximo de vizinhos possível no caso do conjunto de treinamento. Entretanto, verificou-se que valores muito altos para o k não retornavam uma taxa de acerto satisfatória. Dessa forma, o critério adotado para variação do k baseou-se na equação 4.7. Como o número de amostras é igual a 100, os valores de k foram de 1 a 10.

$$1 \leq k \leq \sqrt{n^o \ de \ amostras} \tag{4.7}$$

Para cada valor de k, foram realizadas 5 iterações da validação cruzada e a acurácia média foi obtida. O k correspondente ao melhor resultado apresentado foi utilizado para o conjunto de teste.

Utilizando a métrica de distância euclidiana, o algoritmo foi executado 30 vezes, obtendo-se a acurácia de cada iteração , a acurácia média e o desvio-padrão do conjunto de dados de teste. O pseudo-código Algoritmo 1 exemplifica o código implementado.

Algoritmo 1: Classificação - kNN

Entrada: Amostras dos sinais
Saída: Acurácia média e σ das w iterações

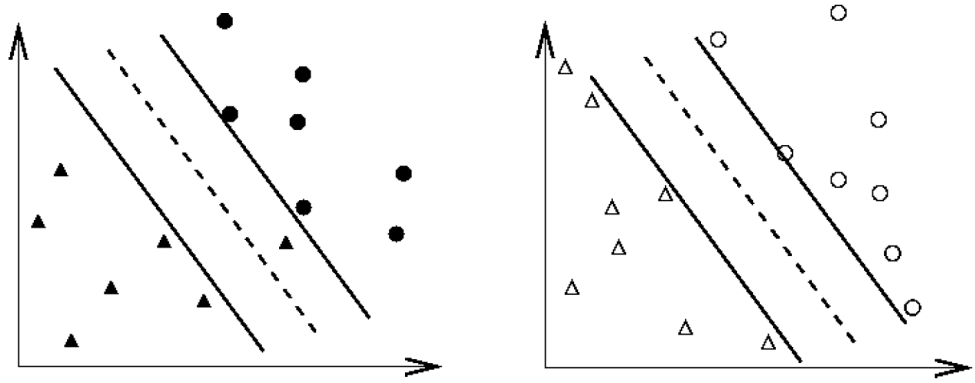
início
  para w = *1 até maxIterações* **faça**
    Aleatoriza as amostras de cada sinal
    treino $\leftarrow$ 80% dos dados
    teste $\leftarrow$ 20% dos dados
    para k = *1 até 10* **faça**
      para *iteracaoFold = 1 até 5* **faça**
        $testeV(iteracaoFold) \leftarrow$ ValidaçãoCruzada(dados=treino)
        $acc(k, iteracaoFold) \leftarrow$ k-nn$(testeV, k)$
      fim
    fim
    $[acc\ ind] \leftarrow max(acc)$
    $k_{best} \leftarrow ind$
    $acc_{teste}(w) \leftarrow$ k-nn$(teste, k_{best})$
  fim
  $acc_{med} \leftarrow mean(acc_{teste}(w))$
  $\sigma \leftarrow sd(acc_{teste}(w))$
fim
retorna $acc_{teste}(w), acc_{med}, \sigma$

4.5.2 SVM

O SVM apresentado por Cortes e Vapnik (1995) é um método muito utilizado em problemas de classificação e regressão (Pedroso e Salles, 2012)(Shan et al., 2005)(Estrela et al., 2013)(Almeida, 2014)(Almeida et al., 2014). Ele aprende na etapa de treino e seleciona dentro dos dados de treinamento pontos que formarão um vetor de suporte para classificação do conjunto de teste (Almeida, 2014). Este vetor de suporte é um hiperplano que otimiza a separação que maximiza a distância entre as classes, sendo usado como fronteira de decisão. A figura 4.23a ilustra o vetor de suporte obtido na etapa de treinamento e a figura 4.23b mostra o vetor de suporte classificando os dados de teste de um problema de classificação binária abordado em Hsu et al. (2016).

Para que fosse possível a comparação entre o k-NN e o SVM, a implementação do SVM seguiu a mesma metodologia do k-NN. Inicialmente os dados foram aleatorizados, com um grupo de treinamento contendo 8 gravações de cada sinal, o que totaliza 80 amostras e o grupo de teste 2 gravações, totalizando 20 amostras.

De posse do dados de treinamento (80% da base de dados), uma validação cruzada foi realizada para a escolha do parâmetro de custo C e o γ. Segundo Granzotto e Lopes (2015), o custo C determina um ponto de equilíbrio razoável entre a maximização da margem e a minimização do erro de classificação e o γ é responsável por ajustar os dados de treinamento ao modelo. Hsu et al. (2016) aconselha que o parâmetro C varie de 2^{-5} a 2^{15} e o γ de 2^{-15} a 2^{3}. Outra característica importante é a escolha do *kernel*. Optou-se pelo *kernel* RBF, pois segundo Hsu et al. (2016) é a melhor escolha quando o número de



(a) Superfície de separação para dados de treinamento

(b) Superfície de separação para dados de teste

Figura 4.23: Exemplo de superfície de separação gerada pelo SVM.
Fonte: Hsu et al. (2016)

características é muito maior que o número de amostras.

Como esse problema é muiticlasse e naturalmente o SVM resolve problemas de classificação binária, utilizou-se o pacote e1071 do *software* R, equivalente à biblioteca LIBSVM, que resolve problemas com SVM Multiclasse utilizando a técnica um-contra-um, colocando todos os subclassificadores binários e identificando a classe correta por um mecanismo de votação (Meyer, 2007). Além disso, este pacote realiza a variação dos parâmetros do classificador.

Após a validação cruzada e escolha dos melhores C e γ, um modelo foi obtido aplicando os melhores parâmetros no conjunto de treinamento. Este modelo, então, foi usado para classificar os dados de teste (20% da base de dados). O pseudocódigo Algoritmo 2 exemplifica o código implementado, o qual foi executado 30 vezes.

Algoritmo 2: Classificação - SVM

Entrada: Amostras dos sinais
Saída: Acurácia média e σ das w iterações

início
  para $w = 1$ *até* *maxIterações* **faça**
    Aleatoriza as amostras de cada sinal
    treino $\leftarrow$ 80% dos dados
    teste $\leftarrow$ 20% dos dados
    $[C, \gamma] \leftarrow$ TuneDeParametros(método=SVM, dados=treino,
      kernel=RBF, C = 2^{-15} a 2^3, $\gamma = 2^{-15}$ a 2^3)
    $model \leftarrow \text{SVM}(dados = treino, C, \gamma)$
    $acc_{teste}(w) \leftarrow \text{SVM}(dados = teste, model)$
  fim
  $acc_{med} \leftarrow mean(acc_{teste}(w))$
  $\sigma \leftarrow sd(acc_{teste}(w))$
fim
retorna $acc_{teste}(w), acc_{med}, \sigma$

4.6 Resumo

Uma metodologia que classificasse de forma satisfatória os sinais foi proposta. O estudo em relação às diversas variações dos vetores de características é um ponto importante por ser uma base de dados utilizada pela primeira vez nesse trabalho. Através das hipóteses a serem testadas, pode-se concluir qual será a melhor representação para este conjunto de dados. Teve-se o cuidado em padronizar a gravação de todos os sinais, aleatorizar a amostra para evitar viés nos dados, além de seguir os padrões estatísticos para a escolha dos melhores parâmetros: k, C e γ. Buscou-se uma metodologia mais geral possível que poderá ser aplicada a uma base de dados mais completa. Vale ressaltar que esse é um estudo exploratório de um problema muito mais complexo que abrange o reconhecimento automático dos mais de 10 mil sinais da língua (Almeida, 2014). A figura 4.24 apresenta de forma mais completa todas as etapas na metodologia proposta para a classificação dos sinais.

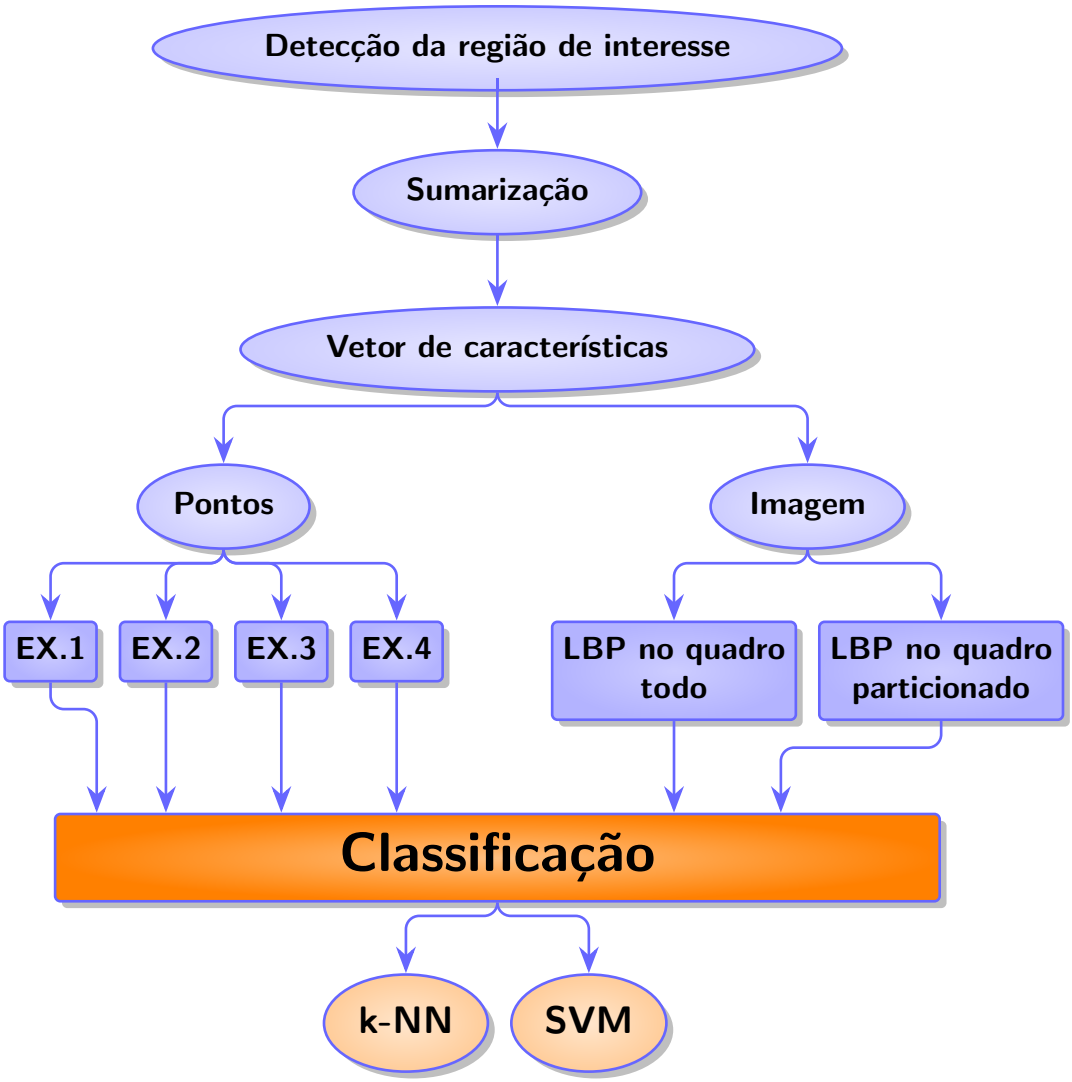


Figura 4.24: Etapas aplicada à base de dados experimental para o reconhecimento dos sinais de Libras.

Capítulo 5

Resultados e Discussões

Sumário

5.1 Introdução

O presente capítulo apresenta os resultados obtidos nas implementações propostas na metodologia. Buscou-se pelas configurações dos vetores de características e pelos parâmetros dos classificadores que retornaram a melhor taxa de acerto no reconhecimento dos sinais de Libras por meio da expressão facial. Para comparar as técnicas de extração de características e os classificadores, os testes estatísticos ANOVA (Análise de Variância) e *Tukey* (Montgomery, 2006) foram realizados.

O teste ANOVA informa a probabilidade de existência de não semelhança entre as distribuições dos dados, ou seja, ele verifica se existem diferenças entre as implementações comparadas. A hipótese nula H_0 indica que as implementações são semelhantes e a hipótese alternativa H_i indica que existe alguma diferença entre elas:

$$\begin{cases} H_0 : \tau_i = 0, \forall i \\ H_i : \exists \tau_i \neq 0 \end{cases}$$

onde τ_i é o deslocamento, da média global, da implementação i.

Caso o teste de ANOVA indique um *p-value* maior que o nível de significância α, a hipótese nula de igualdade entre os desempenhos das implementações não pode ser rejeitada, ou seja, o teste não encontrou diferença entre elas. Caso contrário e com o intuito de verificar diferenças pontuais entre elas, o teste *Tukey* é realizado. Ele compara todos contra todos e caso o intervalo de confiança não englobe o valor 0, alguma diferença é detectada.

O nível de significância α indica a probabilidade da ocorrência de um falso positivo, ou seja, a probabilidade de rejeição da hipótese nula sendo essa verdadeira. Em contrapartida, o nível de confiança $(1 - \alpha)$ indica a chance de se capturar um parâmetro verdadeiro da

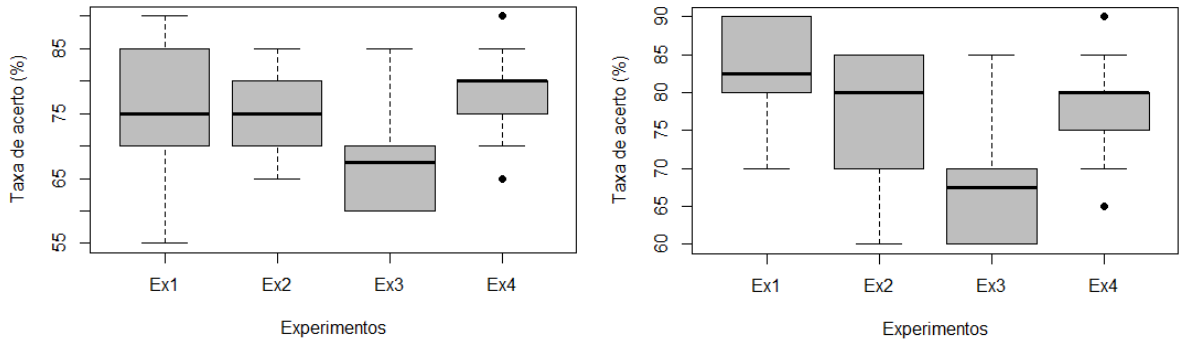
população. Para todas as análises realizadas, adotou-se $\alpha = 0.05$ e o nível de confiança dos testes é de 95%. Vale ressaltar que as premissas do teste ANOVA - normalidade, homoscedasticidade e independência - foram comprovadas através da análise gráfica dos resíduos.

5.2 Classificação dos Sinais

Nesta seção serão apresentados os resultados e análises obtidos nas implementações. Cada implementação do sistema descrito na metodologia foi executada 30 vezes, obtendo-se então 30 valores de taxa de acerto para cada caso.

5.2.1 Vetor de Características: Pontos (x,y) da Face

Foram realizados 4 experimentos utilizando os 121 pontos cartesianos do rosto: (i) sem nenhum tratamento prévio; (ii) normalizados de acordo com a regra Z-score; (iii) normalizados pelo centroide do primeiro quadro; e (iv) normalizados pelo centroide do quadro em questão. O intuito destes experimentos foi verificar se deslocamentos da face afetam o desempenho do classificador quando se trata de gravações distintas de um mesmo sinal. O vetor de características formado pela concatenação das coordenadas cartesianas foram classificados pelo k-NN e pelo SVM. As figuras 5.1a e 5.1b mostram suas taxas de acerto, respectivamente. De forma complementar, a tabela 5.1 apresenta o valor médio e o desvio padrão destes resultados.



(a) Classificando com k-NN (b) Classificando com SVM

Figura 5.1: Taxa de acerto das 30 execuções de cada um dos experimentos cujo vetor de características é composto pela concatenação dos pontos cartesianos.

Tabela 5.1: Taxa de acerto média e desvio padrão utilizando o vetor de características é composto pela concatenação dos pontos cartesianos (30 execuções).

	Ex.1		**Ex. 2**		**Ex. 3**		**Ex.4**	
k-NN	75,50%	σ=8,44	74,50%	σ=6,06	67,66%	σ=6,78	77,83%	σ=6,52
SVM	82,50%	σ=6,79	76,83%	σ=7,59	76,50%	σ=8,11	80,66%	σ=7,15

Aplicando o teste ANOVA nos resultados apresentados na figura 5.1a obteve-se um *p-value* $= 1,01 \times 10^{-6} < \alpha$, indicando que há diferença entre os 4 experimentos classificados com k-NN. Aplicando o teste *Tukey*, verificou-se que o experimento 1 (Ex.1), 2 (Ex.2) e 4 (Ex.4) são similares entre si dentro de um intervalo de confiança de aproximadamente 5% e superiores ao experimento 3 (Ex.3), como apresentado na figura 5.2.

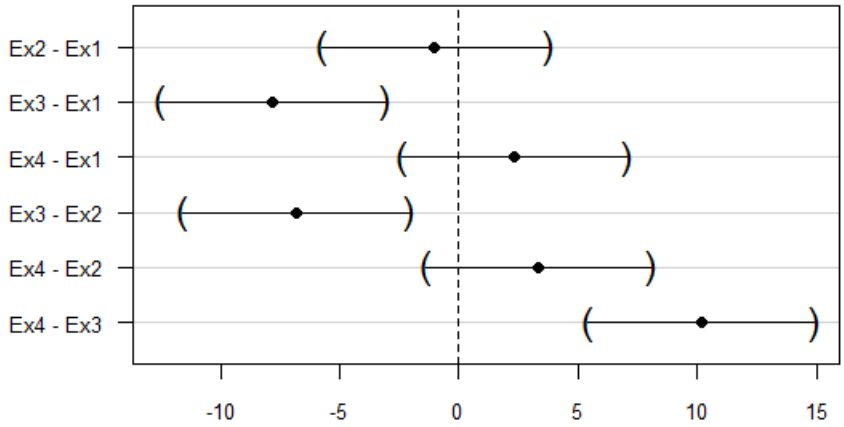


Figura 5.2: Intervalo de confiança para a comparação todos contra todos entre os 4 experimentos classificados com k-NN.

Analisando a classificação realizada pelo SVM, o ANOVA retornou um *p-value* menor que o nível de significância, indicando que há diferença entre os quatro experimentos que estão sendo comparados. Aplicando o teste *Tukey*, verifica-se que o experimento 1 (Ex.1), que utiliza os dados originais da base, é melhor que o experimento 2 (Ex.2), cujos valores da base foram normalizados de acordo com a regra Z-score, e melhor que o experimento 3 (Ex.3), que normaliza os dados pelo centroide do primeiro quadro. Vale ressaltar que o experimento 2 e 4 também superaram o experimento 3. Já o Ex.1 e Ex.2 em relação ao Ex.4, englobaram o valor 0 nos intervalos de confiança, como apresentado na figura 5.3, sendo similares dentro de um intervalo de confiança de aproximadamente 5%.

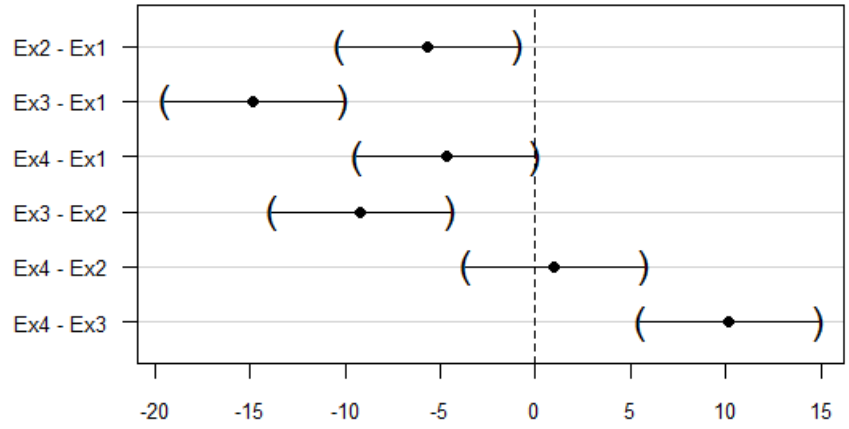


Figura 5.3: Intervalo de confiança para a comparação todos contra todos entre os 4 experimentos classificados com SVM.

Por fim, comparando os 4 experimentos a partir do resultado de ambos os classificado-

res, obteve-se um *p-value* $= 2,3 \times 10^{-12}$, ou seja, existe diferença entre as implementações, de acordo com o teste estatístico ANOVA. Comparando todos contra todos, obteve-se a análise apresentada na figura 5.4. k1 a k4 representam os 4 experimentos classificados com k-NN e S1 a S4 os mesmos classificados com SVM. Os itens destacados na figura 5.4 indicam as implementações não semelhantes, sendo S1 (dados brutos - Ex.1 + SVM) a implementação de melhor desempenho.

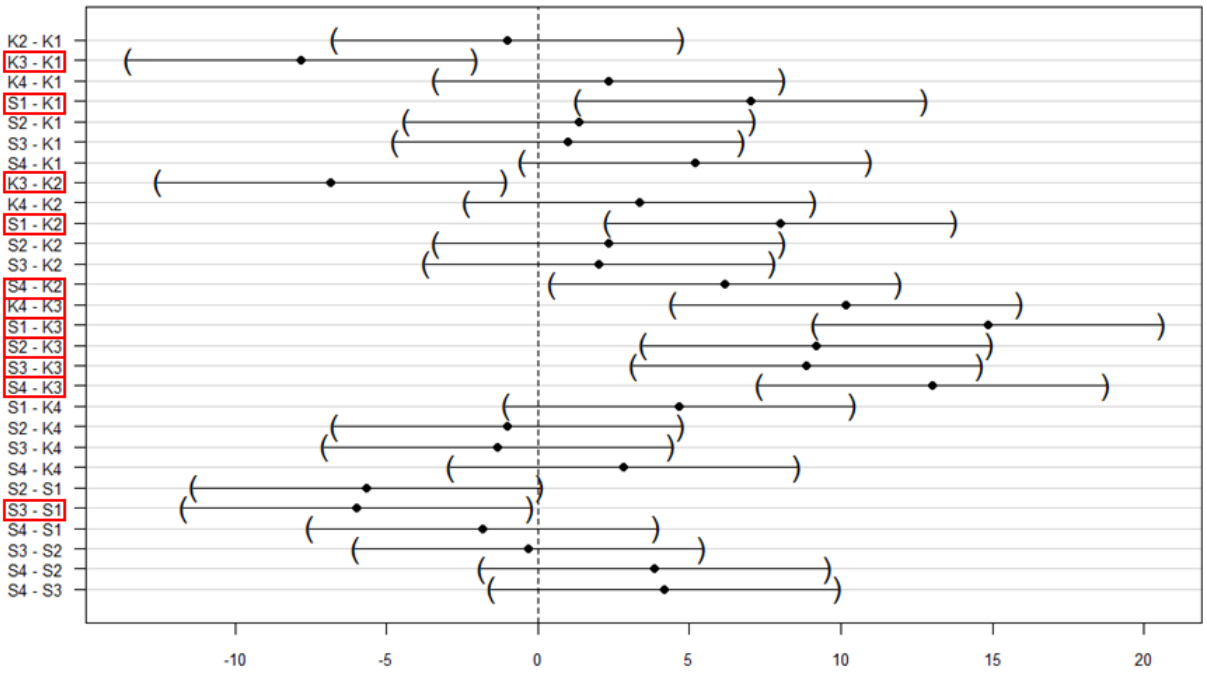


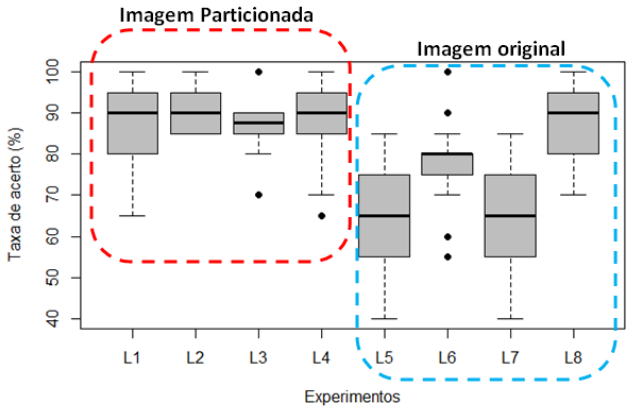
Figura 5.4: Intervalo de confiança para a comparação todos contra todos entre os 4 experimentos com os classificadores k-NN e SVM.

Diante dos resultados obtidos, a melhor configuração envolvendo o vetor de características composto pelas coordenadas (x,y) ocorre quando são utilizados os dados brutos (Ex.1) e o classificador SVM. Essa escolha deve-se ao fato do SVM com dados brutos ser melhor que os experimentos 1, 2 e 3 com k-NN e melhor que o SVM com os dados normalizados pelo centroide do primeiro quadro (Ex.3). Dessa forma, tem-se uma configuração que não necessita de tratamento prévio e, consequentemente, sem custo computacional adicional.

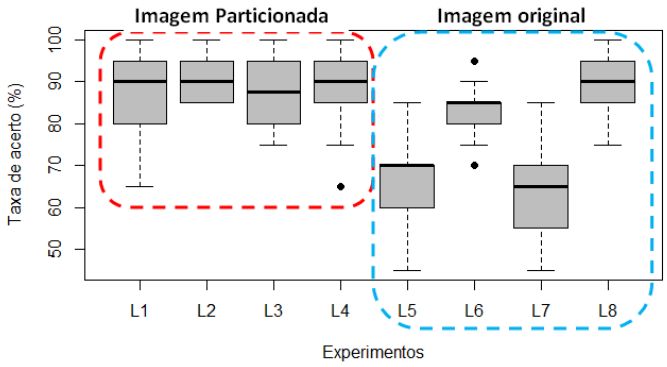
5.2.2 Vetor de Características: Descritor de Textura LBP

Além dos pontos, as informações dos quadros dos vídeos que compõem a gravação de cada sinal foram utilizadas para formar os vetores de características. O operador LBP foi aplicado nas imagens recortadas dos vídeos sumarizados. No entanto, o algoritmo Viola-Jones não detectou a face em 3 dos 500 quadros que formam a base de sinais de Libras. Dessa forma, o recorte do rosto nestes quadros foi feito de três formas distintas e analisou-se qual seria a forma mais adequada quando este erro for detectado. Para cada uma das análises, as configurações do LBP, descritas anteriormente na tabela 4.1 (Capítulo 4), foram aplicadas.

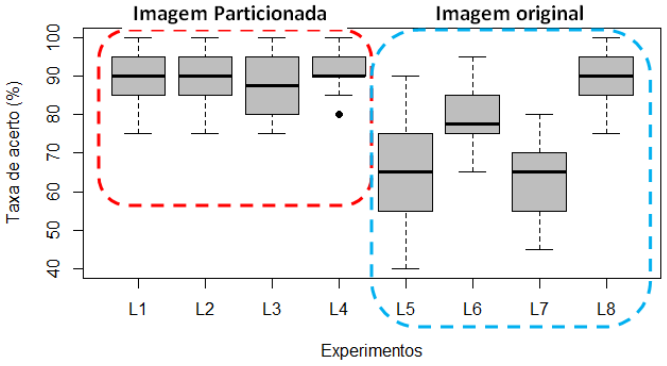
A primeira análise realizada foi recortar os quadros à mão repeitando o tamanho 100x100 do quadro a ser gerado. A segunda análise foi substituir o quadro cuja face não foi detectada pelo quadro mais representativo da sua posição. Por fim, testou-se a substituição pelo LBP médio dos outros 4 quadros significativos das suas respectivas gravações. A figura 5.5 apresenta os resultados obtidos nas análises quando o classificador aplicado foi o k-NN, sendo que os 4 primeiros *boxplots*, L1 a L4, são referentes à imagem dividida em janelas/células e L5 a L8 referentes à imagem original.



(a) Quadros recortados à mão



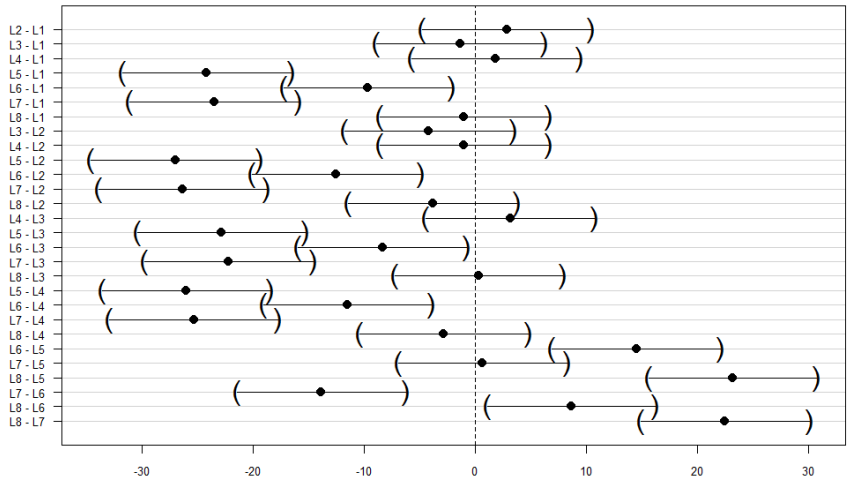
(b) Quadros substituídos



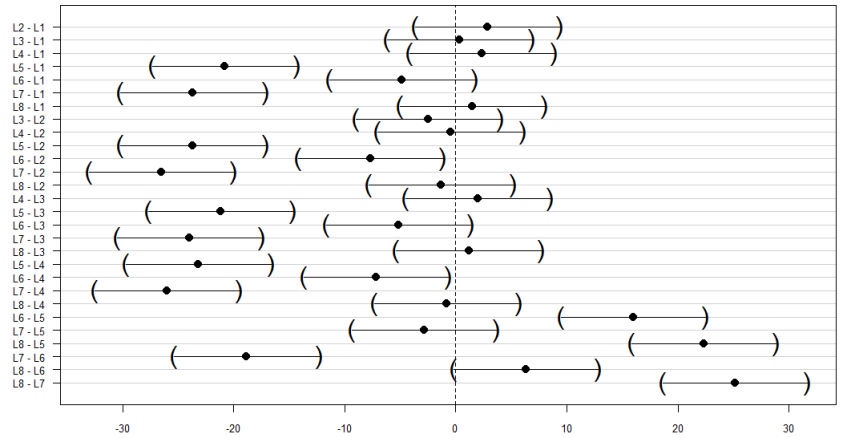
(c) LBP médio

Figura 5.5: 30 execuções de cada um dos operadores classificados com k-NN, sendo L1 e L5: $LBP^{u2}_{8,1}$, L2 e L6: $LBP^{u2}_{8,2}$, L3 e L7: $LBP^{u2}_{12,1}$, L4 e L8: $LBP^{u2}_{12,2}$.

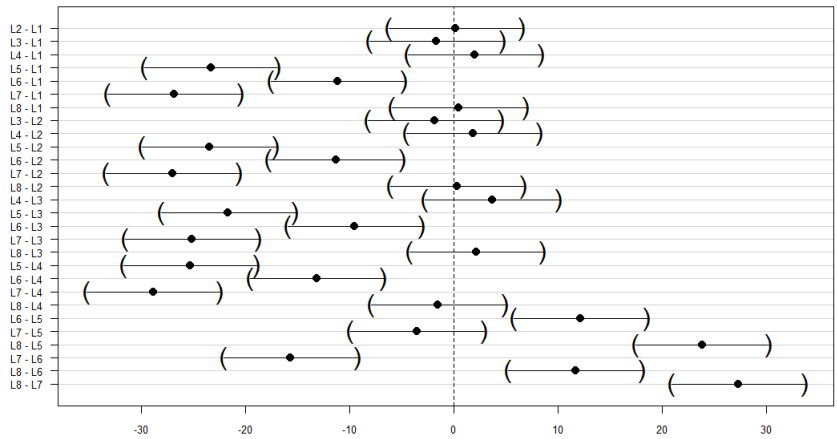
Ao aplicar o teste ANOVA nos 3 casos descritos, verificou-se que existe diferença entre as implementações, ou seja, $p\text{-}value < \alpha$. Para verificar as diferenças pontuais, o teste *Tukey* foi executado. De acordo com a figura 5.6 os operadores L1, L2, L3, L4 e L8 se destacam em relação aos demais em todas as análises.



(a) Quadros recortados à mão



(b) Quadros substituídos



(c) LBP médio

Figura 5.6: Intervalo de confiança das 30 execuções de cada um dos operadores classificados com k-NN.

Analisando os resultados dos operadores LBP aplicados aos quadros, tem-se que as configurações aplicadas à imagem particionada (L1 a L4) e a configuração L8 aplicada à imagem original foram os operadores com as maiores taxas de acerto, com o classificador k-NN. Entretanto, como mostra a tabela 5.2, o tamanho do vetor de características varia para cada uma destas configurações. Esta variação afeta diretamente o custo computacional das implementações, principalmente para o algoritmo de classificação SVM que faz a variação extensa dos parâmetros C e γ. Com isso, o melhor extrator de características, neste caso, foi o L8 ($LBP^{u2}_{12,2}$) aplicado na imagem original, pois este teve um bom desempenho com o menor número de características possível.

Tabela 5.2: Cinco configurações de melhor desempenho quando os operadores LBP foram aplicados juntamente com o classificador k-NN.

Experimento	**Configuração do LBP**	**Número de células por imagem (c)**	**Tamanho do vetor de características do sinal (*bins**c*5)**
L1	$LBP^{u2}_{8,1}$	16	4720
L2	$LBP^{u2}_{8,2}$	16	4720
L3	$LBP^{u2}_{12,1}$	16	10800
L4	$LBP^{u2}_{12,2}$	16	10800
L8	$LBP^{u2}_{12,2}$	1	675

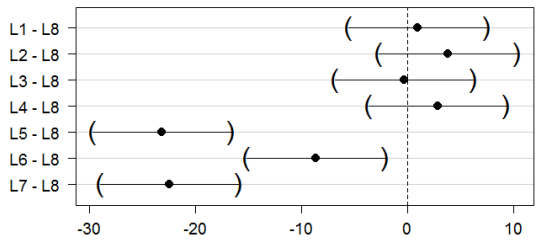
Para justificar esta escolha, o teste estatístico *Dunnet* foi aplicado. Como o teste de *Tukey*, ele também faz comparações múltiplas, entretanto, ele compara todos contra um. A figura 5.7 comprova que o L8 é melhor que L5, L6 e L7, e é similar aos experimentos realizados na imagem particionada. No entanto, ele forma um vetor de características bem menor.

Em relação aos experimentos que substituíram o algoritmo Viola-Jones nos casos que o mesmo não detectou a face, ambos tiveram o mesmo desempenho, como mostra a figura 5.8. Optou-se pelo LBP médio (Análise 3), pois ele independe de uma análise visual, que pode ser uma análise que tendencie o resultado. Além disso, ele possui a melhor taxa de acerto média e o menor desvio-padrão em relação aos outros dois experimentos, como apresentado na tabela 5.3.

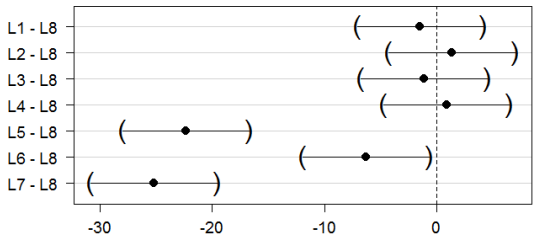
Tabela 5.3: Taxa de acerto média e desvio padrão das 30 execuções cujo vetor de características foi composto pelo operador L8 = $LBP^{u2}_{12,2}$.

	Análise 1	**Análise 2**	**Análise 3**
Média	86,66%	89,16%	89,83%
Desvio-padrão (σ)	8,74	6,44	5,79

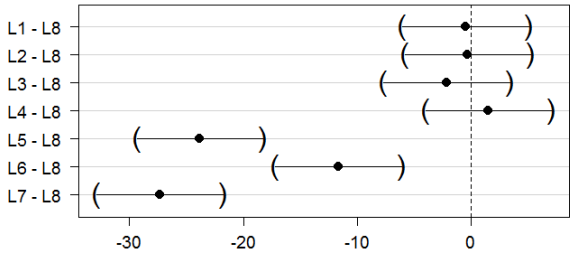
Tendo como base o vetor de características composto pelos pontos cartesianos, percebeu-se uma tendência do classificador SVM ter uma performance melhor que o k-NN. Com isso e tendo em vista o custo computacional dos diversos operadores LBP juntamente



(a) Análise 1



(b) Análise 2



(c) Análise 3

Figura 5.7: Comparação dos melhores operadores LBP aplicados a imagem da face recortada e classificados com k-NN.

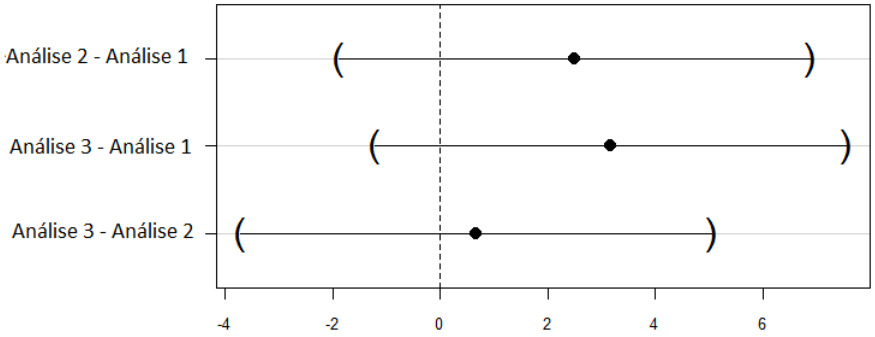


Figura 5.8: Comparação entre as 3 análises quando o operador L8 foi aplicado juntamente com o classificador k-NN, sendo Análise 1 a situação do quadro recortado à mão; Análise 2 quando o quadro foi substituído; e na Análise 3 calculou-se o LBP do quadro médio.

com a variação dos parâmetros do SVM, optou-se por testar apenas os operadores que retornaram os melhores resultados utilizando o classificador k-NN. Neste caso também não houve diferenças significativas entre as implementações testadas dentro de um intervalo de confiança de 3%, como ilustra a figura 5.9.

A tabela 5.4 apresenta a taxa de acerto média e o desvio-padrão dos resultados obtidos pelos experimentos classificados com SVM. Tanto pela análise gráfica quanto pelos valores numéricos encontrados, verifica-se um desempenho muito similar dos experimentos. Dessa forma, optou-se novamente pela análise 3, uma vez que ela apresenta uma boa taxa de acerto e é o caso que não necessita da análise visual.

Comparando os resultados quando o vetor de características foi obtido pelo operador $LBP^{u2}_{12,2}$ classificado pelo k-NN (Figura 5.8) e pelo SVM (Figura 5.9), concluiu-se que a

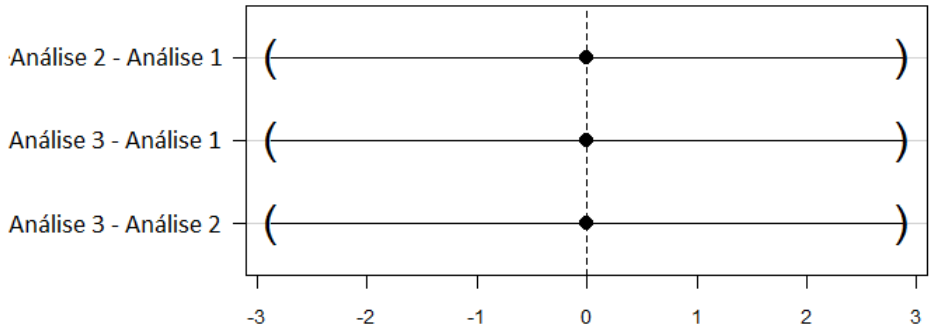


Figura 5.9: Comparação entre as 3 análises quando o operador L8 foi aplicado juntamente com o classificador SVM, sendo Análise 1 a situação do quadro recortado à mão; Análise 2 quando o quadro foi substituído; e na Análise 3 calculou-se o LBP do quadro médio.

Tabela 5.4: Taxa de acerto média e desvio padrão das 30 execuções cujo vetor de características foi composto pelo operador L8 = $LBP^{u2}_{12,2}$ em ambos experimentos e o classificador foi o SVM.

	Análise 1	**Análise 2**	**Análise 3**
Média	94,50%	95,33%	95,33%
Desvio-padrão (σ)	6,06	4,72	4,90

melhor configuração é quando os experimentos foram classificados com SVM como mostra a figura 5.10 obtida pela aplicação do teste *Tukey*. Em todos os casos, o classificador SVM obteve melhores taxas de acerto.

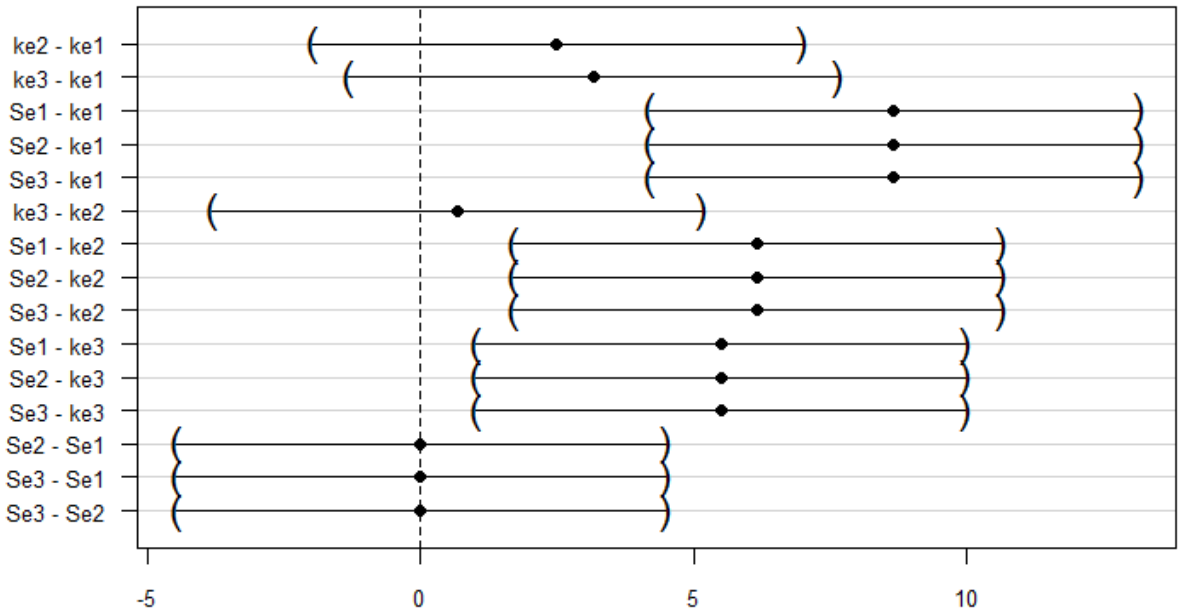


Figura 5.10: Comparação entre os classificadores quando o operador L8 foi aplicado, sendo Ke1 e Se1 os resultados do quadro recortado à mão e classificado com k-NN e SVM, respectivamente; Ke2 e Se2 quando o quadro foi substituído e classificado com k-NN e SVM, respectivamente; e no Ke3 e Se3 calculou-se o LBP do quadro médio e classificou com k-NN e SVM, respectivamente.

5.3 Análise dos Resultados

Cada etapa desse trabalho foi estruturada de forma a obter a melhor performance possível, ou seja, ter a maior taxa de acerto no reconhecimento dos sinais de Libras. Uma metodologia generalizada foi estruturada e cada um dos estágios foi implementado de modo que seu resultado não fosse tendenciado por um viés dos dados. Por isso as amostras são aleatorizadas antes da classificação e houve a variação dos parâmetros dos classificadores. Verificou-se que o SVM obteve melhores resultados de classificação que o k-NN e este desempenho era esperado pela sua capacidade em resolver problemas de classificação e de generalização.

Outro ponto importante foi a performance obtida com o uso do vetor de características composto pelas coordenadas cartesianas. As taxas de acerto não foram expressivas neste caso. Este desempenho pode ser atribuído a pouca variedade na trajetória de alguns pontos e no neste caso, um agrupamento dos parâmetros seria o mais indicado. Entretanto, os resultados obtidos com a aplicação da técnica de Visão Computacional LBP foram promissores.

O melhor resultado obtido quando o vetor de características era composto pela concatenação das coordenadas cartesianas foi com os dados brutos e classificador SVM. Já em relação ao vetor de características obtido pela aplicação do descritor LBP, a melhor configuração foi com o operador $LBP^{u2}_{12,2}$, utilizando o LBP médio para os casos que o algoritmo Viola-Jones não detectou a face e classificando-se com SVM. O apêndice D apresenta os resultados e os respectivos parâmetros de cada uma das 30 execuções destas implementações.

O teste ANOVA que comparou estes dois casos indicou diferenças entre as implementações ($p\text{-}value = 2.05 * 10^{-11}$). De acordo com a figura 5.11, o teste *Tukey* indicou que a melhor representação para os sinais é por meio da utilização das imagens como características.

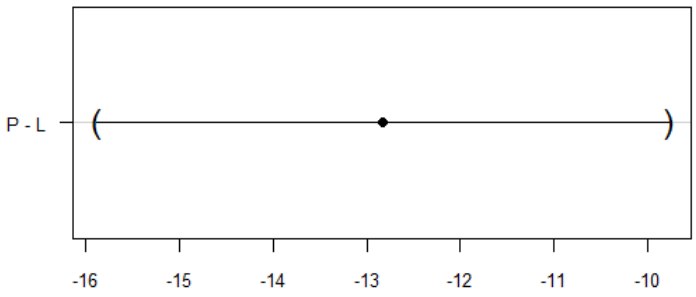


Figura 5.11: Comparação entre o vetor de características obtido pelos pontos (P) e o vetor de características obtido pelo descritor LBP (L). Ambos classificados com SVM.

O critério de comparação entre as implementações realizadas nesse trabalho foi a taxa média de acerto estimada em relação ao conjunto de teste. Entretanto, buscou-se uma métrica para avaliar cada classe. Dentre as várias presentes na literatura, optou-se pela *F-measure* que considera o desempenho para a classe positiva (Castro, 2011). Ela é calculada a partir de duas métricas: precisão e *recall* ou taxa de verdadeiros positivos (TPr), como mostra a equação 5.1. De acordo com Castro (2011), a variável β é utilizada para ajustar

a importância relativa entre precisão e *recall* e pode ser tipicamente adotada como igual a 1.

$$F - measure = \frac{(1+\beta) * Recall * Precisao}{\beta^2 * Recall + Precisao}$$

$$\beta = 1 \Longrightarrow F - measure = \frac{2 * Recall * Precisao}{Recall + Precisao} \tag{5.1}$$

sendo:

$$Precisao = \frac{VP}{VP + FP} \tag{5.2}$$

$$Recall = TPr = \frac{VP}{VP + FN} \tag{5.3}$$

sabendo que: VP, verdadeiro positivo, é o número de classificações corretas do sinal; FP, falso positivo, é o número de amostras classificadas erroneamente como sendo da classe em questão e FN, falso negativo, é o número de amostras da classe em questão classificadas erroneamente como sendo de outras classes.

Com base nas tabelas 5.5 e 5.6, obteve-se o resultado da métrica *F-measure* para a avaliação de cada classe nas implementações que estão sendo comparadas. O valor máximo da diagonal principal destas tabelas, que representam as matrizes de confusão, é 60, pois as implementações foram executadas 30 vezes e 20% dos dados de cada sinal compunha o conjunto de teste. A tabela 5.7 apresenta os valores de precisão, *recall* e *F-measure* das melhores implementações.

Tabela 5.5: Matriz de confusão do sistema para a melhor classificação obtida com o vetor de característica obtido pela concatenação dos pontos cartesianos e classificado com SVM. Sinais: Acalmar (Aca), Acusar (Acu), Aniquilar (Ani), Apaixonado (Apa), Engordar (Eng), Felicidade (Fel), Magro (Mag), Sortudo (Sor), Surpresa (Sur) e Zangado (Zan).

		Predição									
		Aca	**Acu**	**Ani**	**Apa**	**Eng**	**Fel**	**Mag**	**Sor**	**Sur**	**Zan**
Saída Real	**Aca**	60	0	0	1	0	1	0	0	8	0
	Acu	0	44	2	0	0	0	0	1	0	13
	Ani	0	5	55	0	0	0	0	1	0	5
	Apa	0	0	0	52	0	3	0	1	5	0
	Eng	0	0	0	0	60	9	1	0	0	1
	Fel	0	0	0	1	0	33	3	0	4	0
	Mag	0	0	0	0	0	2	50	0	0	0
	Sor	0	0	0	0	0	0	2	57	0	0
	Sur	0	0	0	6	0	12	4	0	43	0
	Zan	0	11	3	0	0	0	0	0	0	41

Ao analisar a tabela 5.7, percebe-se que o vetor de características obtido pelo operador LBP teve um resultado melhor para a métrica *F-measure* quando comparado com o resultado gerado pelo vetor composto pelas coordenadas cartesianas. Dessa forma, os resultados corroboram com a análise da taxa de acerto realizada anteriormente. Com isso, a melhor configuração para a classificação de sinais de Libras por meio da expressão facial é utilizando o vídeo dos sinais, extraindo as características através do operador LBP e classificando os padrões por meio da técnica SVM.

Tabela 5.6: Matriz de confusão do sistema para a melhor classificação obtida com o vetor de característica obtido pelo LBP e classificado com SVM. Sinais: Acalmar (Aca), Acusar (Acu), Aniquilar (Ani), Apaixonado (Apa), Engordar (Eng), Felicidade (Fel), Magro (Mag), Sortudo (Sor), Surpresa (Sur) e Zangado (Zan).

		Predição									
		Aca	**Acu**	**Ani**	**Apa**	**Eng**	**Fel**	**Mag**	**Sor**	**Sur**	**Zan**
Saída Real	**Aca**	60	0	0	0	0	0	0	0	0	0
	Acu	0	50	2	0	0	0	0	0	0	7
	Ani	0	0	58	0	0	0	0	0	0	0
	Apa	0	0	0	60	0	0	0	0	0	0
	Eng	0	0	0	0	60	0	4	0	0	0
	Fel	0	0	0	0	0	56	0	0	1	0
	Mag	0	0	0	0	0	0	56	0	0	0
	Sor	0	0	0	0	0	0	0	60	0	0
	Sur	0	0	0	0	0	4	0	0	59	0
	Zan	0	10	0	0	0	0	0	0	0	53

Tabela 5.7: Precisão, *Recall* e *F-measure* para cada uma das classes em cada uma das implementações.

	Pontos + SVM			**LBP + SVM**		
	Precisão	**Recall**	**F-measure**	**Precisão**	**Recall**	**F-measure**
Acalmar	0.8571	**1**	0.9230	**1**	**1**	**1**
Acusar	0.7333	0.7333	0.7333	**0.8474**	**0.8333**	**0.8402**
Aniquilar	0.8333	0.9166	0.8729	**1**	**0.9666**	**0.9830**
Apaixonado	0.8524	0.8666	0.8594	**1**	**1**	**1**
Engordar	0.8450	**1**	0.9159	**0.9375**	**1**	**0.9677**
Felicidade	0.5500	0.5500	0.5500	**0.9824**	**0.9333**	**0.9572**
Magro	0.9615	0.8833	0.9207	**1**	**0.9333**	**0.9654**
Sortudo	0.9661	0.9500	0.9579	**1**	**1**	**1**
Surpresa	0.6615	0.7166	0.6879	**0.9365**	**0.9833**	**0.9593**
Zangado	0.7454	0.6833	0.7130	**0.8412**	**0.8833**	**0.8617**

Por fim, verificou-se quais gravações eram classificadas erroneamente para cada sinal em cada uma das implementações, como mostra a tabela 5.8. Percebe-se que a maioria das amostras cujo vetor de características foi obtido pela aplicação do operador LBP também aparece quando o vetor de características foi composto pela concatenação dos pontos cartesianos, ambos classificados com SVM. Além disso, na configuração obtida pelos pontos (x,y) da base de dados experimental há um número muito maior de amostras que não foram classificadas corretamente, confirmando que o vetor de características obtido pelo operador LBP é a melhor representação para essa base de dados. Entretanto, verifica-se que existem amostras que são classificadas erradamente em apenas uma das implementações. Este fato sugere, futuramente, o desenvolvimento de um sistema híbrido utilizando as informações dos pontos e do operador LBP. Vale ressaltar nos dados apresentados na tabela 5.8 não foi computado a quantidade de vezes que cada amostra foi classificada erradamente. Esta análise depende da frequência com que cada amostra é sorteada para compor o conjunto de teste e nesse trabalho esta frequência é aleatória em cada iteração.

Tabela 5.8: Gravações classificadas erroneamente para cada sinal em cada uma das implementações.

	Pontos + SVM	**LBP + SVM**
Acalmar	-	-
Acusar	1 / 2 / 3 / 4 / 5 / 7 / 10	4 / 5 / 6 / 10
Aniquilar	5 / 6 / 7 / 10	6
Apaixonado	1 / 9 / 10	-
Engordar	-	-
Felicidade	1 / 2 / 6 / 8 / 9 / 10	5 / 9
Magro	1 / 10	7
Sortudo	4 / 10	-
Surpresa	3 / 4 / 8 / 9	8
Zangado	2 / 3 / 4 / 7 / 8 / 9	4 / 10

As tabelas 5.5, 5.6 e 5.8 mostram que os sinais Acusar, Aniquilar, Felicidade, Surpresa e Zangado tiveram uma alta taxa de erro na classificação. Ao analisar as imagens de cada sinal, presentes no apêndice C, verifica-se que a expressão dos sinais Acusar, Aniquilar e Zangado são muito similares e as expressões do sinal Felicidade se confundem com as do sinal Surpresa. No entanto, estes sinais são diferentes entre si quando os parâmetros fonológicos da língua relativos às mãos são considerados. Como apenas a expressão facial foi analisada nesse trabalho foi natural os erros encontrados. Vale ressaltar que a expressão facial é um dos parâmetros principais da língua, mas para que seja possível o reconhecimento dos sinais de Libras de forma completa é necessário complementar a informação da face com o movimento das mãos.

O operador LBP teve um rendimento melhor que os pontos. Como são poucas amostras e muitas características, provavelmente o classificador encontrou padrões espúrios que permitiram separar as classes, justificando esta diferença de performance. O LBP é sensível a cor de pele, rosto do sinalizador e características da face, sendo que o extrator de pontos é indiferente a estas variáveis, tornando-o um método vantajoso ao lidar com estes fatores. O ideal para trabalhos nessa linha de pesquisa é que a base de dados fosse gravada por mais de um sinalizador e o método de extração de características fosse associado exclusivamente a expressão facial.

Capítulo 6

Conclusões e Propostas de Continuidade

Sumário

A língua brasileira de sinais, oficializada em 2002, é uma das formas de comunicação da comunidade surda. Desenvolver um sistema de reconhecimento desta língua é desafiador pelo fato dos trabalhos desta linha de pesquisa atuarem apenas em resolver problemas pontuais e pelo fato de ainda não existir um sistema de reconhecimento robusto para a classificação dos elementos que compõem a língua, uma vez que uma das suas principais características é ser visual.

Essa dissertação abordou um dos parâmetros fonológicos da Libras, a expressão facial, para o reconhecimento de sinais da língua. Primeiramente uma base experimental foi gravada com o intuito de validar a metodologia proposta. Em cada gravação de cada sinal o rosto foi detectado e recortado, e o vídeo com as imagens da face passou pelo processo de sumarização. Com isso, obtiveram-se 5 quadros que representam cada gravação. Em seguida, dois vetores de características foram construídos: um com as coordenadas (x,y) dos 121 do rosto de cada um dos quadros significativos e outro com a aplicação do descritor de textura LBP também aplicado a cada um dos quadros. Por fim, as informações extraídas da base de dados experimental foram classificadas tanto com o classificador k-NN quanto com o SVM.

Por meio desta metodologia foi possível atingir uma taxa de acerto média de 95% para o vetor de características extraído da aplicação do operador LBP, $LBP^{u2}_{12,2}$, em cada um dos quadros significativos, classificando com SVM. Foi possível com esta configuração obter um descritor de cada sinal com uma dimensionalidade menor quando comparado com qualquer um dos demais testes feitos nesse trabalho. Esta característica do sistema resultou na implementação de menor tempo computacional. O critério de escolha desta configuração foi a taxa média de acerto. No entanto, verificou-se pelo cálculo da métrica *F-measure* que o desempenho em cada classe nesta configuração também atingiu níveis satisfatórios.

Conclui-se, então, que o objetivo de obter uma abordagem que utilize técnicas de Inteligência Computacional com uma boa taxa de acerto no reconhecimento automático de sinais de Libras por meio da expressão facial foi atingido. Este é um problema muito complexo e com muitas variáveis a serem controladas e uma metodologia foi idealizada inicialmente a partir de decisões tomadas para auxiliar na classificação dos sinais. O

reconhecimento de Libras tendo como base apenas o parâmetro expressão não-manual é somente uma parte do problema. Esta informação deve ser agregada ao gesto das mãos e movimento do corpo para o completo reconhecimento do sinal.

Uma das limitações desse trabalho e que tem sido recorrente em trabalhos dessa natureza é o número reduzido de amostras que compõem a base de dados experimental. Dentro das limitações de tempo e recursos, apenas 10 sinais foram gravados. Tendo como referência o conjunto de dados criado em Almeida (2014), preocupou-se na quantidade de amostras de cada sinal. Por isso, cada sinal foi gravado 10 vezes ao invés de 5, como em Almeida (2014). Outro ponto importante nesse estudo foi a aplicação do algoritmo de sumarização. Optou-se pelos 5 quadros mais significativos de cada sinal, no entanto esta análise foi visual e com base nos estudos realizados em Almeida (2014).

Verificou-se que o desempenho do classificador k-NN não foi superior aos resultados do SVM em nenhuma das implementações. Entretanto, os seus resultados estiveram acima de 86% com o vetor de características constituído pelo operador LBP, sendo este um resultado promissor. Vale ressaltar que ambos os classificadores tiveram um desempenho menor quando o vetor de pontos foi utilizado. Neste caso, tem-se vetores de 1210 características e no vetor resultante da aplicação do $LBP^{u2}_{12,2}$ a dimensionalidade diminuiu para 675. De acordo com Bishop (1995), a partir de uma determinada quantidade de atributos o desempenho do classificador tende a diminuir, mesmo que esses atributos sejam significativos. Este fenômeno é denominado mal da dimensionalidade e pode ser evitado por meio de uma seleção de características.

Ainda assim, há muito a se fazer para que um sistema de reconhecimento automático de sinais atenta a seu principal público: a comunidade surda. O foco principal desse trabalho foi a importância da expressão facial na identificação dos sinais da Libras e os resultados obtidos demonstraram a potencialidade da metodologia proposta e quão importante são as expressões não-manuais na identificação de um sinal.

6.1 Propostas para Trabalhos Futuros

Sugere-se como propostas de continuidade desse trabalho:

- Criação de uma base de dados de sinais de Libras robusta, que permita a validação de sistemas de reconhecimento automático dos sinais;
- Desenvolver um algoritmo que analise quantos quadros são significativos para descrever os sinais;
- Realizar uma análise comparativa dentre os diversos descritores de imagem existentes na literatura para a classificação de imagens de representam sinais dinâmicos;
- Realizar uma validação cruzada para separar os dados de treinamento e os de teste, certificando que uma determinada amostra participará de ambos os conjuntos;
- Realizar a seleção de características para redução da dimensionalidade do vetor de características;

- Aplicar outros classificadores, como o *Random Forest*;
- Implementar um sistema de classificação híbrido, que junte ambas as informações da base de dados: pontos e imagens;
- Desenvolver um sistema de reconhecimento de sinais de Libras que aborde todos os parâmetros da língua.

Referências Bibliográficas

J. C. Ahlberg. Candide-3 - an updated parameterized face. *Technical report*, (LiTH-ISY-R2326), 2001.

T. Ahonen, A. Hadid, e M. Pietikainen. Face recognition with local binary patterns. In *European conference on computer vision - ECCV*, p. 469–481. Springer, 2004.

S. G. M. Almeida. *Extração de Características em Reconhecimento de Parâmetros Fonológicos da Língua Brasileira de Sinais utilizando Sensores RGB-D*. Tese de Doutorado, Universidade Federal de Minas Gerais, Programa de Pós Graduação em Engenharia Elétrica, Belo Horizonte, Minas Gerais, Brasil, 2014.

S. G. M. Almeida, F. G. Guimarães, e J. A. Ramírez. A methodology for feature extraction in brasilian sign language recognition. In *Proceedings of the IX Workshop de Visao Computacional, WVC*, 2013.

S. G. M. Almeida, F. G. Guimarães, e J. A. Ramírez. Feature extraction in brazilian sign language recognition based on phonological structure and using rgb-d sensors. *Expert Systems with Applications*, 41(16):7259–7271, 2014.

S. G. M. Almeida, A. R. R. Freitas, e F. G. Guimarães. Um método para sumarização de vídeos baseado no problema da diversidade máxima e em algoritmos evolucionários. In *XII Simpósio Brasileiro de Automação Inteligente (SBAI)*, p. 1298 – 1303, Natal, Rio Grande do Norte, Brasil, 2015.

V. Amaral e C. E. Thomaz. Extração e comparação de características locais e globais para o reconhecimento automático de imagens de faces. In *VIII Workshop de Visão Computacional (WVC)*, Goiânia, Goias, Brasil, 2012.

I. L. O. Bastos. Reconhecimento de sinais da libras utilizando descritores de forma e redes neurais artificiais. Dissertação de Mestrado, Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia e Universidade Estadual de Feira de Santana, Salvador, Bahia, Brasil, 2015.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford university press, Grã-Bretanha, Inglaterra, 1995. ISBN 0198538642.

CadavidConcepts. Face model diagram. http://www.nuicapture.com/img/face_model_diagram.pdf.

F. C. Capovilla, W. D. Raphael, e A. C. L. Maurício. *Dicionário Enciclopédico Ilustrado Trilíngue da Língua Brasileira de Sinais (Libras) baseado em Linguística e Neurociências Cognitivas, Volume I: Sinais de A a H.*, Volume 1. Edusp, Brasil, 2 ed., 2012a. ISBN 9788531413315.

F. C. Capovilla, W. D. Raphael, e A. C. L. Maurício. *Dicionário Enciclopédico Ilustrado Trilíngue da Língua Brasileira de Sinais (Libras) baseado em Linguística e Neurociências Cognitivas, Volume II: Sinais de I a Z.*, Volume 2. Edusp, Brasil, 2 ed., 2012b. ISBN 9788531413315.

A. T. S. Carneiro, P. C. Cortez, e R. C. S. Costa. Reconhecimento de gestos da libras com classificadores neurais a partir dos momentos invariantes de hu. *Interaction South America 2009*, p. 190 – 195, 2009.

C. L. Castro. *Novos critérios para seleção de modelos neurais em problemas de classificação com dados desbalanceados*. Tese de Doutorado, Universidade Federal de Minas Gerais, Programa de Pós Graduação em Engenharia Elétrica, Belo Horizonte, Minas Gerais, Brasil, 2011.

S. Chao, Z. Tianzhu, B. Bing-Kun, X.Changsheng, e M. Tao. Discriminative exemplar coding for sign language recognition with kinect. *IEEE Transactions on Cybernetics*, 43(5):1418–1428, 2013.

C. Cortes e V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

N. Dalal e B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Volume 1, p. 886–893. IEEE, 2005.

B. DeWilde. Classification of hand-written digits (3). http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/, 2012.

A. Dhall, R. Goecke, S. Lucey, e T. Gedeon. Static facial expressions in tough conditions: Data, evaluation protocol and benchmark. p. 2106–2112, 2011.

A. Dhall, R. Goecke, S. Lucey, e T. Gedeon. Collecting large, richly annotated facial-expression databases from movies. *IEEE Multimedia*, 19(3):34 – 41, 2012a.

A. Dhall, J. Joshi, I. Radwan, e R. Goecke. In *Asian Conference on Computer Vision*, p. 613–626. Springer, 2012b.

J. B. Dias, K. P. Souza, e H. Pistori. Conjunto de treinamento para algoritmos de reconhecimento de libras. In *II Workshop de Visão Computacional*, São Carlos, São Paulo, Brasil, 2006.

F. A. Diniz, F. M. M. Neto, F. C. L. Júnior, e L. M. O. Fontes. Redface: um sistema de reconhecimento facial baseado em técnicas de análise de componentes principais e autofaces: comparação com diferentes classificadores. *Revista Brasileira de Computação Aplicada*, 5(1):42 – 54, 2013.

F. A. Diniz, T. R. Silva, e F. E. S. Alencar. Um estudo empírico de um sistema de reconhecimento facial utilizando o classificador knn. *Revista Brasileira de Computação Aplicada*, 8(1):50 – 63, 2016.

E. Doi, T. Inui, T.-W. Lee, T. Wachtler, e T. J. Sejnowski. Spatiochromatic receptive field properties derived from information-theoretic analyses of cone mosaic responses to natural scenes. *Neural computation*, 15(2):397–417, 2003.

M. Duduchi e F. C. Capovilla. Buscasigno: a construção de uma interface computacional para o acesso ao léxico da língua de sinais brasileira. In *Proceedings of VII Brazilian symposium on Human factors in computing systems*, p. 21–30, Natal, Rio Grande do Norte, Brasil, 2006. ACM.

P. Ekman e W. V. Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124 – 129, 1971.

B. N. S. Estrela, G. Cámara-Chávez, M. F. M. Campos, W. R. Schwartz, e E. R. Nascimento. Sign language recognition using partial least squares and rgb-d information. In *Proceedings of the IX Workshop de Visao Computacional, WVC*, 2013.

B. Fasel e J. Luettin. Automatic facial expression analysis: a survey. *Patter Recognition*, 36(1):259 – 275, 2003.

A. R. R. Freitas, F. G. Guimarães, R. C. P. Silva, e M. J. F. Souza. Memetic self-adaptive evolution strategies applied to the maximum diversity problem. *Optimization Letters*, 8(2):705–714, 2014a.

F. A. Freitas, S. M. Peres, C. A. M. Lima, e F. V. Barbosa. Grammatical facial expressions recognition with machine learning. In *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference*, p. 180 – 185, Palo Alto: The AAAI Press, 2014b.

F. A. Freitas, S. M. Peres, C. A. M. Lima, e F. V. Barbosa. Grammatical facial expressions data set. https://archive.ics.uci.edu/ml/datasets/Grammatical+Facial+Expressions#, 2014c.

V. Ghosal, P. Tikmani, e P. Gupta. Face classification using gabor wavelets and random forest. In *Computer and Robot Vision, 2009. CRV'09. Canadian Conference on*, p. 68–73. IEEE, 2009.

L. C. Gonçalves, R. B. Andrade, R. D. Campos, . A. Romero, e E. F. Saad. Redes neurais artificiais e processamento de imagem no reconhecimento de libras, usando kinect. *Jornal de Engenharia, Tecnologia e Meio Ambiente - JETMA*, 1(1):32 – 37, 2016.

M. H. Granzotto e L. C. O. Lopes. Desenvolvimento de sistema de detecção de falhas baseado em aprendizado estatístico de máquinas de vetores de suporte. *Blucher Chemical Engineering Proceedings*, 1(2):11819–11828, 2015.

D. Hamester, P. Barros, e S. Wermter. Face expression recognition with a 2-channel convolutional neural network. In *2015 International Joint Conference on Neural Networks - IJCNN*, p. 1–8. IEEE, 2015.

R. M. Haralick, I. Dinstein, e K. Shanmugam. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610 – 621, 1973.

C. Hsu, C. Chang, e C. Lin. A practical guide to support vector classification. 2016.

M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179 – 187, 1962.

A. K. Jain, R. P. W. Duin, e J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.

E. G. L. Júnior, L. H. Silva, C. J. P. Passarinho, e R. A. L. Rabêlo. Um robusto reconhecimento facial por filtro de gabor curvo e entropia. *Revista de Sistemas e Computação*, 6(1):80 – 89, 2016.

T. Kadir, R. Bowden, E. J. Ong, e A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *British Machine Vision Conference - BMVC*, p. 1–10, 2004.

T. Kanade, J. F. Cohn, e Y. Tian. Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, p. 46–53. IEEE, 2000.

G. O. Koroishi e B. V. L. Silva. Reconhecimento de sinais da libras por visão computacional. *Revista Mecatrone*, 1(1):1 – 9, 2015.

C. C. Kuo, F. Glover, e K. S. Dhir. Analyzing and modeling the maximum diversity problem by zero-one programing. *Decision Sciences*, 24(6):1171–1185, 1993.

J. Li e N. M. Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10):1771–1787, 2008.

Y. Liu, Y. Cao, Y. Li, M. Liu, R. Song, Y. Wang, Z. Xu, e X. Ma. Facial expression recognition with pca and lbp features extracting from active facial patches. In *Real-time Computing and Robotics (RCAR), IEEE International Conference on*, p. 368–373. IEEE, 2016.

M. Lyons, S. Akamatsu, M. Kamachi, e J. Gyoba. Coding facial expressions with gabor wavelets. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, p. 200 – 205. IEEE, 1998.

D. Meyer. Support vector machines - the interface to libsvm in package e1071. 2007.

D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2006. ISBN 0470088109.

M. Musci, R. Q. Feitosa, M. L. F. Velloso, e T. Novack. Padrões binários locais na classificação de imagens de sensoriamento remoto. In *Anais XV Simpósio Brasileiro de Sensoriamento Remoto - SBSR*, p. 7651 – 7658, Curitiba, Paraná, Brasil, 2011.

S. Noushath, G. H. Kumar, e P. Shivakumara. (2d) 2lda: An efficient approach for face recognition. *Pattern Recognition*, 39(7):1396 – 1400, 2006.

T. Ojala, M. Pietikinen, e D. Harwood. A comparative study of texture measures with classification based on featured distribution. *Pattern Recognition*, 29(1):51–59, 1996.

T. Ojala, M. Pietikinen, e T. Maenpaa. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. *IEEE Transitions on Pattern Analysis and Machine Intelligence*, 24(7):971 – 987, 2002.

OMS. Deafness and hearing loss. http://www.who.int/mediacentre/factsheets/fs300/en/, Mar. 2015.

E. Patrick e F. Fischer. A generalized k-nearest neighbor rule. *Information and control*, 16(2):128 – 152, 1970.

H. Pedrini e W. R. Schwartz. *Análise de Imagens Digitais: princípios, algoritmos e aplicações*. Thomson Learning, São Paulo, Brasil, 2008. ISBN 9788522105953.

F. J. C. Pedroso e E. O. T. Salles. Reconhecimento de expressões faciais baseado em modelagem estatística. In *Anais do XIX Congresso Brasileiro de Automática*, p. 631–638, 2012.

D. A. Ramos. Metodologia de busca de similaridade de genes por matriz de co-ocorrência. Dissertação de Mestrado, Pós-Graduação em Bioinformática da Universidade Federal do Paraná, Curitiba, Paraná, Brasil, 2012.

T. M. Rezende, C. L. Castro, e S. G. M. Almeida. An approach for brazilian sign language (bsl) recognition based on facial expression and k-nn classifier. In F. A. M. Cappabianco, F. A. Faria, J. Almeida, e T. S. Körting, editors, *Electronic Proceedings of the 29th Conference on Graphics, Patterns and Images (SIBGRAPI'16)*, São José dos Campos, SP, Brasil, october 2016. URL http://gibis.unifesp.br/sibgrapi16.

J. R. Santos, M. G. F. Costa, e C. F. F. C. Filho. Reconhecimento das configurações de mão de libras baseado na análise de discriminante de fisher bidimensional, utilizando imagens de profundidade. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, Manaus, Amazonas, Brasil, 2015.

C. Shan, S. Gong, e P. McOwan. Robust facial expression recognition using local binary patterns. In *IEEE International Conference on Image Processing 2005*, Volume 2, p. 367–370. IEEE, 2005.

C. Shan, S. Gong, e P. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, (27):803–816, 2009.

A. L. Sousa, S. W. S. Costa, Y. Pires, e F. P. Araújo. Reconhecimento de expressões faciais e emocionais como método avaliativo de aplicações computacionais. In *Anais do Encontro Regional de Computação e Sistemas de Informação*, p. 178–187, Manaus, Amazonas, Brasil, 2016. Universidade do Estado do Pará.

R. N. Valverde, F. G. Pereira, M. C. P. Santos, e R. F. Vassalo. Reconhecimento de gestos em 3d com kinect para interação homem-computador. In *Anais do XIX Congresso Brasileiro de Automática - CBA*, p. 4084–4089, Campina Grande, Paraíba, Brasil, 2012.

P. Viola e M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2(57):137–154, 2004.

G. Zhao e M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.

Apêndice A

Publicações

A.1 XII Simpósio de Mecânica Computacional - 2016

Reconhecimento de Expressões Faciais da Língua Brasileira de Sinais (LIBRAS) utilizando os classificadores k-NN e SVM.

RECONHECIMENTO DE EXPRESSÕES FACIAIS EM SINAIS DA LÍNGUA BRASILEIRA DE SINAIS (LIBRAS) UTILIZANDO OS CLASSIFICADORES K-NN E SVM

Tamires Martins Rezende, tamiresrezende@ufmg.br
Cristiano Leite de Castro, crislcastro@gmail.com
Felipe Augusto Oliveira Mota, felipemota@ufmg.br
Ciniro Aparecido Leite Nametala, ciniro@gmail.com
Ramon Santos Corrêa, ramonscorrea36@ufmg.br
Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais - Av. Antônio Carlos, nº 6627, 31270-901, Belo Horizonte, MG, Brasil.

Sílvia Grasiella Moreira Almeida, silvia.almeida@ifmg.edu.br
IFMG - Campus Ouro Preto - Rua Pandiá Calógeras, nº 898, Bauxita, 35400-000, Ouro Preto, MG, Brasil.

Resumo.*O reconhecimento automático de expressões faciais via imagens é um problema complexo que requer a aplicação de técnicas de Inteligência Computacional. Essas técnicas buscam reproduzir alguns aspectos do comportamento humano, tal como a capacidade de aprendizado. Diante desse aspecto, este trabalho traz resultados de uma metodologia para o reconhecimento de expressões faciais via sensores RGB-D. O objetivo é ser capaz de diferenciar fisionomias para posterior incorporação em um sistema reconhecedor da Libras. A metodologia proposta foi avaliada com 7 dos 34 sinais que compõem o dataset utilizado, sendo que cada sinal foi capturado 5 vezes. Optou-se pelos sinais (Justo/Amar/Angustiado/Comemorar/Rancor/Engordar/Brigar) cuja expressão facial se alterava ao longo de sua execução e as seguintes etapas foram executadas para cada sinal: (i) detecção e recorte da região de interesse (rosto), (ii) sumarização do vídeo utilizando o conceito da maximização da diversidade, (iii) criação do descritor, (iv) criação do vetor de características e (v) classificação com k-NN (k-vizinhos-mais-próximos) e SVM (Máquinas de Vetores de Suporte) Multiclasse. Obteve-se uma acurácia máxima de 85,71% com o SVM Multiclasse. Após essa etapa e com o objetivo de aumentar o conjunto de dados aplicou-se o método SMOTE (Synthetic Minority Over-sampling Technique) para geração de amostras sintéticas, mas a taxa de acerto na classificação continuou a mesma devido a ocorrência de sobreposição de amostras que o método está sujeito. Os resultados alcançados no estudo realizado mostram que o modelo proposto teve um desempenho considerável, possibilitando a construção de um sistema automático de reconhecimento útil aos usuários da língua.*

Palavras-chave: *Inteligência Computacional, Libras, k-NN, SVM Multiclasse, SMOTE.*

1. INTRODUÇÃO

A Visão Computacional procura auxiliar a resolução de problemas altamente complexos, buscando imitar a cognição humana e a habilidade do ser humano em tomar decisões de acordo com as informações contidas, por exemplo, em uma imagem (Pedrini e Schwartz, 2008). Como ramo da Visão Computacional, tem-se o Reconhecimento de Padrões, que pode ser definido como uma área de pesquisa que busca classificar dados de entradas de acordo com a semelhança dos seus termos, agrupando-os em classes. Aplicações neste ramo são inúmeras, tais como reconhecimento de caracteres, reconhecimento de expressões faciais e análise de expressão gênica e, basicamente, para estes tipos de aplicações a realização de tarefas possui as etapas: (i) Aquisição da imagem/vídeo; (ii) Segmentação da região/objetivo de interesse; (iii) Extração das características sobre a região de interesse; (iv) Seleção de características; e (v) Classificação das imagens/vídeos.

Na literatura, encontram-se muitos trabalhos de reconhecimento de expressões faciais relacionados com a emoção, tais como o artigo de Pedroso e Salles (2012) que propôs um sistema de reconhecimento de expressões de raiva, felicidade, tristeza, surpresa, medo, nojo e neutra, fazendo a localização da face através do algoritmo de Viola-Jones, extraindo as características pelo método estatístico AAM (*Active Appearance Model*) e classificando com k-NN e SVM. Outro exemplo deste tipo de aplicação é o trabalho de Oliveira e Jaques (2013), que apresenta um sistema computacional que classifica as emoções chamadas de básicas (raiva, medo, repulsa, surpresa, alegria e tristeza), por meio das expressões faciais do usuário captadas por uma webcam. Diferentemente dos trabalhos citados, o foco deste trabalho não está no reconhecimento da emoção propriamente dita, mas sim, no reconhecimento das expressões faciais que estão associadas a determinados sinais da Libras. A ideia é, portanto, propor uma metodologia que seja capaz de diferenciar fisionomias para posterior incorporação a um sistema reconhecedor de Libras.

A Libras é reconhecida oficialmente no Brasil desde 2002, por meio da Lei n^o 10.436, de 24 de abril de 2002. Para determinar o significado de um sinal, menor unidade da língua de sinais, torna-se importante a localização das mãos em relação ao corpo, a expressão facial, a movimentação que se faz ou não na hora de produzir o sinal, a orientação da palma da mão, entre outras características. Além das caracteísticas citadas, há um importante parâmetro para diferenciar sinais, denominado Expressão Não-Manual, ou seja, expressões formadas pelo movimento da face, dos olhos, da cabeça ou do tronco que compõem a construção sintática da linguagem (Almeida, 2014).

Para testar a metodologia proposta no artigo, utilizou-se a base de dados criada por Almeida (2014) em sua tese. Foram escolhidos 7 sinais, nos quais a expressão facial se alterava ao longo de sua execução, e cada sinal passou pelas etapas de detecção da região de interesse (rosto) e recorte da mesma, sumarização do vídeo contendo apenas o rosto, criação do descritor, criação do vetor de características e classificação. Após a classificação aplicou-se um método de geração de amostras sintéticas com o objetivo de aumentar o conjunto de dados e estes foram novamente submetidos a etapa de classificação. Obteve-se uma acurácia máxima de 85,71% com classificador SVM Multiclasse e a taxa de acerto continuou a mesma após geração de amostras sintéticas.

O artigo está organizado da seguinte forma: A Seção 2.apresenta as características do banco de dados utilizado nesse artigo. Em sequência, a metodologia de trabalho é apresentada na seção 3.Na seção 4são expostos os resultados encontrados e a conclusão do trabalho encontra-se na seção 5.

2. BANCO DE DADOS DE LIBRAS

O *dataset* utilizado nesse artigo foi criado para a tese de Almeida (2014). A principal contribuição deste trabalho foi a extração de características de sinais relacionados à estrutura fonológica da Língua Brasileira de Sinais a partir de vídeos RGB-D e o reconhecimento automático destes parâmetros através de um sistema computacional (Almeida, 2014). As etapas seguidas no trabalho de Almeida (2014) foram:

- Escolha dos sinais: a língua possui mais de 10 mil verbetes e diante da constante mutação e expansão da língua, houve a necessidade de selecionar apenas alguns sinais para o reconhecimento.
- Gravação dos sinais selecionados: utilizando um sensor RGB-D (*Kinect*) operado por meio do software *nuiCaptureAnalyze* obteve-se simultaneamente o vídeo de intensidade RGB, o vídeo de profundidade, o vídeo do esqueleto e os dados de 20 pontos do corpo humano.
- Extração de características: esta etapa envolveu a sumarização de vídeos (com o intuito de reduzir o tamanho dos vídeos, eliminando informações redundantes), a detecção da região de interesse (neste caso, as mãos) e a extração de descritores robustos capazes de diferenciar os sinais manuais.
- Reconhecimento do sinal: classificação das amostras de teste como pertencentes a algum grupo, algum sinal.

Foram selecionados 34 sinais de Libras sendo que cada sinal foi capturado cinco vezes. Para distinguir os sinais, escolheu-se quatro parâmetros: o ponto de articulação, a configuração das mãos, o movimento e a orientação da palma da mão

(Almeida, 2014) de cada uma das mãos.

Dos 34 sinais que compõem o *dataset* original, sete foram escolhidos para esse trabalho. Esta escolha teve como base a alteração da expressão/posição facial durante sua execução. A Figura 1, tendo como base as imagens de Capovilla *et al.* (2012a) e Capovilla *et al.* (2012b), ilustra estes sinais.

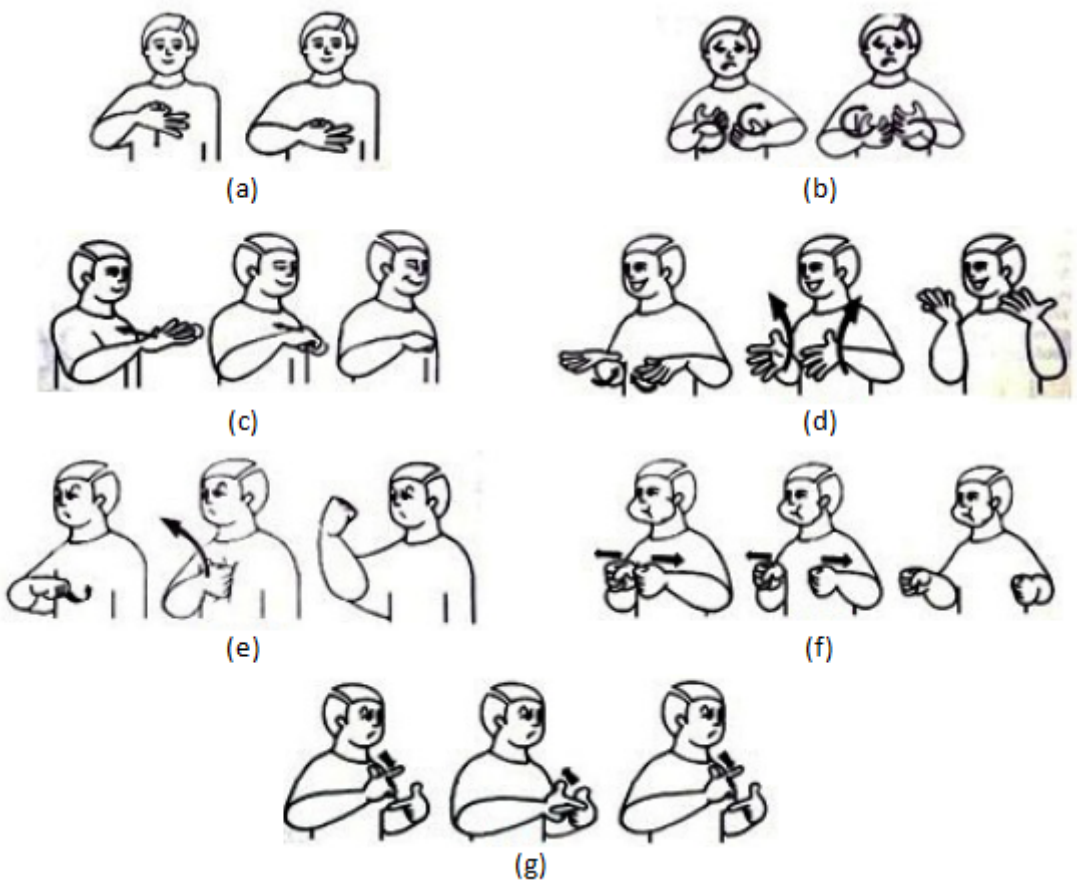


Figura 1. Sinais: (a) Justo, (b) Angustiado, (c) Amar, (d) Comemorar, (e) Rancor, (f) Engordar, (g) Brigar

3. METODOLOGIA

As etapas seguidas neste trabalho foram definidas após um estudo minucioso para que o modelo seja adequado a estrutura de dados que se tem e alcance uma acurácia (taxa de acerto) satisfatória:

1. Detecção da Região de Interesse: como o objetivo do trabalho é a detecção da expressão facial, tem-se apenas o rosto como região de interesse, de forma que na sumarização (próximo passo) apenas as mudanças na expressão facial são detectadas. O recorte do rosto foi realizado tendo como referência o pixel central do quadro, pois todas gravações foram feitas numa mesma posição e na parte central do vídeo. A Figura 2 mostra um quadro completo e a Fig. 3 ilustra a região de interesse detectada.

2. Sumarização: esta etapa tem várias vantagens para o trabalho, seja na redução de custo computacional, eliminação de quadros redundantes (informação desnecessária) e até tornar a extração de características mais eficiente. Diante das várias técnicas de sumarização encontradas na literatura, nesse artigo optou-se por utilizar uma abordagem do problema clássico de otimização conhecido como Problema da Diversidade Máxima, apresentado em Kuo *et al.* (1993), para extrair os quadros mais relevantes em um vídeo, baseando-se nas diferenças existentes entre eles.
   Dessa forma, é criado um vídeo com as imagens obtidas no passo anterior (Detecção da Região de Interesse). Este vídeo que tem uma taxa de 30 quadros por segundo será sumarizado. Optou-se pelos 5 quadros mais significativos

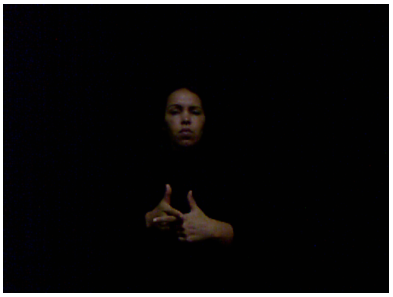


Figura 2. Frame completo do sinal Brigar

Figura 3. Rosto Detectado

(Fig. 4) tendo como base os teste feitos por Almeida (2014) em sua tese. Vale ressaltar que a sumarização permitiu obter vetores de caraterísticas de tamanhos iguais para todos os sinal.

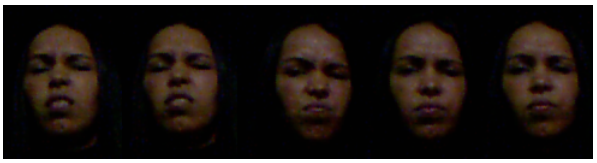


Figura 4. 5 quadros significativos do sinal Angustiado

3. Criação do Descritor: O objetivo dessa etapa foi obter uma representação de cada sinal que seja robusta e invariante a transformações. Com os 5 quadros retornados da etapa anterior (Sumarização), obtêm-se as coordenadas (*x,y*) dos 121 pontos do rosto de cada quadro de cada sinal (Fig. 5 - pontos vermelhos). Estas coordenadas são obtidas pelo software *nuiCaptureAnalyze* (*http://nuicapture.com/*) que opera o *Kinect* (*https://dev.windows.com/en-us/kinect*). A dimensão do descritor de cada quadro é 1x242 e tem a seguinte representação:

$$D = \left[\begin{array}{cccc} (x_1, y_1) & (x_2, y_2) & \ldots & (x_{121}, y_{121}) \end{array} \right]_{1x242}$$

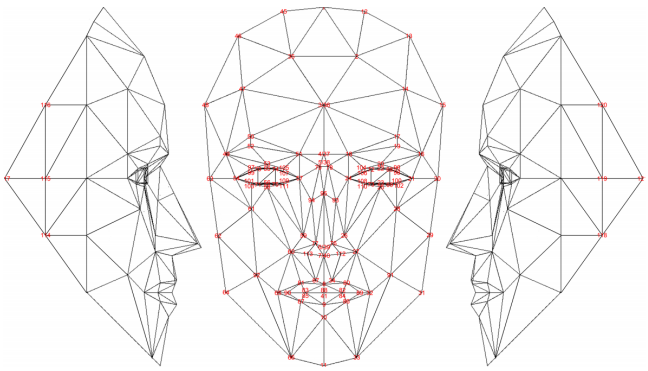


Figura 5. 121 pontos referentes ao rosto

4. Vetor de Características: Para cada sinal, o vetor de características será composto pela concatenação do descritores de cada um dos 5 quadros que o compõe. Sua dimensão será de 1x1210 e sua representação final é:

$$Vetor = \left[\begin{array}{ccccc} D_1 & D_2 & D_3 & D_4 & D_5 \end{array} \right]_{1x1210}$$

5. Classificação I: Na classificação dos sinais, utilizou-se o classificadores k-NN (k vizinhos mais próximos) (Patrick e Fischer, 1970) e SVM (Máquinas de Vetores de Suporte) Multiclasse (Chang e Lin, 2014). Optou-se por estes classificadores, pois o k-NN é indicado para *datasets* que tem poucas amostras e o SVM Multiclasse é próprio para problemas com mais de duas classes e é considerado como o estado da arte na tarefa de reconhecimento de padrões. Para determinar a classe de um elemento que não pertença ao conjunto de treinamento, o classificador k-NN procura k elementos do conjunto de treinamento que estejam mais próximos deste elemento desconhecido, ou seja, que tenham a menor distância e atribui a amostra a classe que recebeu o voto majoritário em relação aos k vizinhos mais próximos. Há várias métricas de distância (Euclidean, Cityblock, Chebychev, Correlation, Cosine, Hamming, Jaccard, Minkowski, Seuclidean e Spearman) e todas elas foram testadas buscando encontrar a acurácia máxima.
   Já o SVM encontra um hiperplano que otimiza a separação das classes, conhecido como hiperplano ótimo ou ideal, que maximiza a distância entre as classes, sendo usado como fronteira de decisão.
   Como entrada para os classificadores, tem-se a matriz $X_{35x1210}$ (35 amostras e 1210 características) e a saída desejada é a matriz Y_{35x1} (35 amostras e 1 saída).

$$X_{35x1210} = \left[\begin{array}{c} Vetor_{Justo/Amostra1} \\ Vetor_{Justo/Amostra2} \\ Vetor_{Justo/Amostra3} \\ Vetor_{Justo/Amostra4} \\ Vetor_{Justo/Amostra5} \\ Vetor_{Amar/Amostra1} \\ \cdots \\ Vetor_{Brigar/Amostra4} \\ Vetor_{Brigar/Amostra5} \end{array} \right] \qquad Y_{35x1} = \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ \cdots \\ 7 \\ 7 \end{array} \right]$$

6. Geração de Dados Sintéticos: Em busca de melhores resultados e sabendo-se da dificuldade/custo de gerar novas amostras, optou-se por gerar novos dados sinteticamente através do método SMOTE (*Synthetic Minority Over-sampling Technique*) (Chawla *et al.*, 2002). O algoritmo SMOTE cria dados artificiais, baseados nas semelhanças, no espaço de características, entre os exemplos existentes da classe minoritária, mas nesse trabalho ele é utizado para aumentar as amostras, tendo em vista que os dados já são balanceados. A lógica do algoritmo é a seguinte:

   - Define-se um valor para k (número de pontos vizinhos para cada amostra x_i).
   - Para cada amostra, calculam-se as distâncias euclidianas entre x_i e as demais amostras, sendo os k-vizinhos-próximos os de menor magnitude.
   - Escolha randomicamente um dos k-vizinhos-próximos e faça: $x_{novo} = x_i + (x_k - x_i).\delta,\ onde\ \delta = [0, 1]$. A Figura 6 exemplifica esse passo.

   Sabendo-se que cada sinal tem 5 amostras, apenas as amostras para treinamento passaram pelo método SMOTE.

7. Classificação II: Nesta última etapa, aplicou-se apenas o SVM Multiclasse, sendo que o conjunto de treinamento é composto por dados originais do *dataset* e os dados sintéticos gerados na etapa anterior (Geração de Dados Sintéticos) e no teste foram utilizadas amostras originadas do *dataset*.

Todas as etapas descritas foram feitas para todos os sinais de todas as amostras, sendo que o número de quadros varia em cada captura feita, como mostra a Tab. 1.

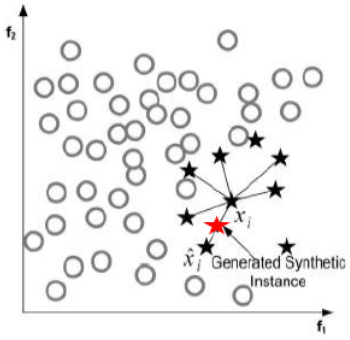


Figura 6. Método SMOTE - k vizinhos mais próximo e nova amostra gerada

Tabela 1. Número de quadros de cada sinal em cada amostra

Sinal	Amostra				
	01	02	03	04	05
Justo	46	20	46	28	36
Amar	59	86	84	57	94
Angustiado	92	95	63	67	63
Comemorar	63	40	41	56	51
Rancor	53	47	45	59	71
Engordar	56	35	46	44	42
Brigar	30	51	72	72	67

4. RESULTADOS

Para as etapas de treinamento e teste utilizando o kNN, fez-se uso da técnica *Leave-one-out* (1 amostra é separada para teste e as n-1 restantes são treinadas) variando o número de vizinhos de 1 a 34. Neste caso encontrou-se uma acurácia máxima de 73.53%, sendo a métrica de distância de *Chebychev* e k=1. A distância de *Chebychev* é um cálculo de distância no qual considera-se o máximo valor da distância de uma dimensão.
O método SVM Multiclasse teve duas variações: 3 amostras para treino e 2 para teste, e 4 amostras para treino e 2 para teste. A acurácia máxima encontrada foi:

- 3 amostras de treino e 2 de teste: 42.86%
- 4 amostras de treino e 1 de teste: 85.71%

Diante desses resultados, pensou-se numa alternativa para aumentar a taxa de acerto. Analisando os 121 pontos da Fig. 5, selecionou-se 10 destes pontos (Fig. 7) e todas as etapas descritas até o momento foram refeitas. O critério de escolha dos pontos foi empírico, buscando os pontos que representassem bem as alterações nos elementos que compõem a expressão facial: testa, sobrancelha, olhos, boca e queixo.
Para esse teste, a acurácia máxima encontrada foi:
kNN:

- *Chebychev* e k=1: 73.53%

SVM Multiclasse:

- 3 amostras de treino e 2 de teste: 42.86%
- 4 amostras de treino e 1 de teste: 71.43%

Os resultados mostrados até aqui foram da etapa de Classificação I, que consideram somente a base de dados original, sem os dados sintéticos gerados via SMOTE.

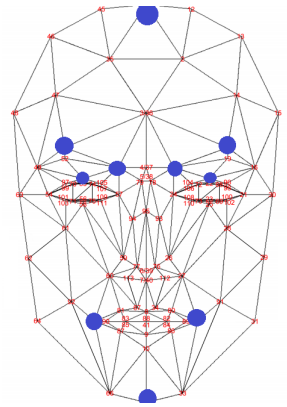


Figura 7. 10 pontos selecionados - círculo azul

Ainda em busca de melhores resultados, aplicou-se o método SMOTE nas duas variações do SVM e apenas nos dados de treinamento. Quando havia 3 dados de treinamento, o SMOTE gerou mais 6 dados (9 no total) e quando se tinha 4, o SMOTE gerou mais 16 (20 no total). Para 3 amostras de treinamento, os resultados obtidos estão na Tab. 2 e para 4 amostras de treinamento, os resultados obtidos estão na Tab. 3.

Tabela 2. Resultados do SVM + SMOTE (3 dados de treinamento)

Amostra de teste	**Acurácia (%)**	**Número de amostras classificadas erroneamente**
01 e 02	64,29	5
01 e 03	35,71	9
01 e 04	35,71	9
01 e 05	42,86	8
02 e 03	64,29	5
02 e 04	57,14	6
02 e 05	**85,71**	2
03 e 04	42,86	8
03 e 05	28,57	10
04 e 05	50	7

Tabela 3. Resultados do SVM + SMOTE (4 dados de treinamento)

Amostra de teste	**Acurácia (%)**	**Número de amostras classificadas erroneamente**
05	**85,71**	1
04	42,86	4
03	51,14	3
02	**85,71**	1
01	14,29	6

5. CONCLUSÃO

Em relação aos classificadores aplicados ao *dataset* original, obteve-se uma acurácia máxima de 85,71% com o SVM Multiclasse, sendo 4 amostras de treino e 1 para teste. Apesar do k-NN ser indicado para conjuntos de poucas amostras e este é o caso desse trabalho, acredita-se que a grande dimensionalidade (quantidade de características) tenha sido um fator determinante para a taxa de acerto inferior. Percebeu-se, também, que os 10 pontos escolhidos não são pontos que representam bem os quadros e é necessário uma análise mais minuciosa dos mesmo, ou seja, outras técnicas de seleção de características podem ser testadas.

Em relação ao método SMOTE que foi utilizado com o intuito de melhorar a acurácia de classificação, observou-se que os resultados obtidos não foram eficientes neste quesito. Esse resultado deve-se ao fato de que o método SMOTE gera o

mesmo número de amostras de dados sintéticas para cada exemplo inicial minoritário, sem considerar amostras vizinhas, o que aumenta a ocorrência de sobreposição entre as classes e esta sobreposição não gerou a diversidade desejada, comprometendo a acurácia do método de classificação.
Os resultados alcançados nesse trabalho foram satisfatórios, o que motiva incorporação da metodologia proposta em um sistema automático de reconhecimento de Libras.

AGRADECIMENTOS

Os autores agradecem ao PPGEE-UFMG pelo incentivo e direcionamento. O presente trabalho foi realizado com o apoio financeiro da CAPES - Brasil.

NOMENCLATURA

(x, y)	Coordenada de cada ponto do rosto	k	Número de vizinhos mais próximos
D	Descritor (matriz de coordenadas)	x_i	Posição da amostra referência no método SMOTE
$Vetor$	Vetor de características	x_k	Posição da amostra vizinha a x_i
X	Entrada do classificador	δ	Valor randômico entre 0 e 1
Y	Saída desejada do classificador	x_{novo}	Posição da nova amostra gerada pelo SMOTE

REFERÊNCIAS

Almeida, S.G.M., 2014. *Extração de Características em Reconhecimento de Parâmetros Fonológicos da Língua Brasileira de Sinais utilizando Sensores RGB-D*. Tese (Doutorado), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.

Capovilla, F.C., Raphael, W.D. e Maurício, A.C.L., 2012a. *Dicionário Enciclopédico Ilustrado Trilíngue da Língua Brasileira de Sinais (Libras) baseado em Linguística e Neurociências Cognitivas, Volume I: Sinais de A a H. [S.l.]*. Edusp, Brasil.

Capovilla, F.C., Raphael, W.D. e Maurício, A.C.L., 2012b. *Dicionário Enciclopédico Ilustrado Trilíngue da Língua Brasileira de Sinais (Libras) baseado em Linguística e Neurociências Cognitivas, Volume II: Sinais de I a Z. [S.l.]*. Edusp, Brasil.

Chang, C.C. e Lin, C.J., 2014. "Libsvm – a library for support vector machines". URL https://www.csie.ntu.edu.tw/~cjlin/libsvm/. Acesso em: 01/03/2016.

Chawla, N.V., Bowyer, K.W., Hall, L.O. e Kegelmeyer, P.W., 2002. "Smote: Synthetic minority over-sampling technique". *Journal of Artificial Intelligence Research 16*, pp. 321–357.

Kuo, C.C., Glover, F. e Dhir, K.S., 1993. "Analyzing and modeling the maximum diversity problem by zero-one programing". *Decision Sciences*, Vol. 24, No. 6, pp. 1171–1185.

Oliveira, E. e Jaques, P.A., 2013. "Classificação de emoções básicas através de imagens capturadas em vídeos de baixa resolução". *Revista Brasileira de Computação Aplicada*, Vol. 5, No. 2, pp. 40–54.

Patrick, E. e Fischer, F., 1970. "A generalized k-nearest neighbor rule". *Elsevier*, Vol. 16, No. 2, pp. 128–152.

Pedrini, H. e Schwartz, W.R., 2008. *Análise de Imagens Digitais: princípios, algoritmos e aplicações*. Thomson Learning, São Paulo, Brasil.

Pedroso, F.J.C. e Salles, E.O., 2012. "Reconhecimento de expressões faciais baseado em modelagem estatística". *XIX Congresso Brasileiro de Automática*, pp. 631–638.

NOTA DE RESPONSABILIDADE

Os autores são os únicos responsáveis pelo material reproduzido nesse artigo.

A.2 WFPA - Workshop on Face Processing Applications - 2016

An approach for Brazilian Sign Language (BSL) recognition based on facial expression and k-NN classifier.

An approach for Brazilian Sign Language (BSL) recognition based on facial expression and k-NN classifier

Tamires Martins Rezende, Cristiano Leite de Castro
The Electrical Engineering Graduate Program
Federal University of Minas Gerais
Belo Horizonte, Brazil
Email: {tamires,crislcastro}@ufmg.br

Sílvia Grasiella M. Almeida
Department of Industrial Automation
Federal Institute of Minas Gerais - Ouro Preto
Ouro Preto, Brazil
Email: silvia.almeida@ifmg.edu.br

***Abstract*—The automatic recognition of facial expressions is a complex problem that requires the application of Computational Intelligence techniques such as pattern recognition. As shown in this work, this technique may be used to detect changes in physiognomy, thus making it possible to differentiate between signs in BSL (Brazilian Sign Language or LIBRAS in Portuguese). The methodology for automatic recognition in this study involved evaluating the facial expressions for 10 signs (to calm down, to accuse, to annihilate, to love, to gain weight, happiness, slim, lucky, surprise, and angry). Each sign was captured 10 times by an RGB-D sensor. The proposed recognition model was achieved through four steps: (i) detection and clipping of the region of interest (face), (ii) summarization of the video using the concept of maximized diversity, (iii) creation of the feature vector and (iv) sign classification via k-NN (k-Nearest Neighbors). An average accuracy of over 80% was achieved, revealing the potential of the proposed model.**

***Keywords*-RGB-D sensor; Brazilian Sign Language; k-NN; Facial expression.**

I. Introduction

Facial expressions are an important non-verbal form of communication, characterized by the contractions of facial muscles and resulting facial deformations [1]. They demonstrate feelings, emotions and desires without the need for words, and are an essential element in the composition of signs.

To determine the meaning of a sign – the smallest unit of the language – the location, orientation, configuration, and trajectory of both hands is essential. In addition to these characteristics, Non-Manual Expressions are features that can qualify a sign and add to its meaning, as well as being specific identifiers of a given sign [2].

The BSL recognition using computational methods is a challenge for a variety of reasons:

- There is currently no standardized database containing signs in a format that allows for the validation of computational classification systems;
- One sign is composed of various simultaneous elements;
- The language does not contain a consistent identifier for the start and end of a sign;
- Different people complete any given gesture differently.

To solve the first problem, a database was created with this study in mind. The following 10 signs - to calm down, to accuse, to annihilate, to love, to gain weight, happiness, slim, lucky, surprise, and angry - were chosen and captured 10 times, performed by the same speaker. An RGB-D sensor, the Kinect [3], operated through the nuiCaptureAnalyze [4] software was used.

According to [2], recent studies indicate the 5 main parameters of the BSL: point of articulation, hand configuration, movement, palm orientation and non-manual expressions. As the our focus is recognize facial expression, we chose 10 signs containing changes in facial expression during their execution. Then, this paper does an exploratory study of the peculiarities involved in non-manual sign language expression recognition.

Initially, a literature review of Computational Intelligence techniques applied to the sign recognition was accomplished. Promising results were reported recently in [2], in which feature extraction was done through the use a RGB-D sensor and a SVM (Support Vector Machine). However, the focus was in the motion of the hands. Another inspiring article is presented in [5]. While not addressing sign language, it applied Convolution Neural Networks method in the GAFFE dataset for facial expression recognition, achieving good results. Another important reference is [6], which proposed a system for recognizing expressions of anger, happiness, sadness, surprise, fear, disgust and neutrality, using the Viola-Jones algorithm to locate the face, extracting characteristics with the AM (Active Appearance Model) method, and classifying using k-NN and SVM.

Despite having distinct aims, these studies fit under the view of pattern recognition and served as the main references for the methodology proposed here.

It is important to note that the Maximum Diversity Problem was addressed in the Summarization. In the Classification step, cross-validation was used to identify the best value of k for k-Nearest Neighbor classifier and, through this, an average accuracy of 80% was reached.

The remainder of the paper is organized as follows: section II describes the database created for the validation of the method proposed. That is followed by an explanation of the methodology in section III. In section IV, the experiments and results are presented, with a conclusion in section V.

II. The Brazilian Sign Language database

For the creation of the database, the steps followed in [2] were used as reference. The first step was to choose the signs that contained changes in facial expression during execution (to calm down, to accuse, to annihilate, to love, to gain weight, happiness, slim, lucky, surprise and angry). After this, the signs were recorded. A scenario was created so that the speaker was in a fixed position (sitting in a chair), in approximately 1.2 meters from the RGB-D sensor, as shown in figure 1. This configuration was adopted since it allowed for focusing on the face. With the 10 signs each recorded 10 times with the same speaker, the balanced database had a total of 100 samples.

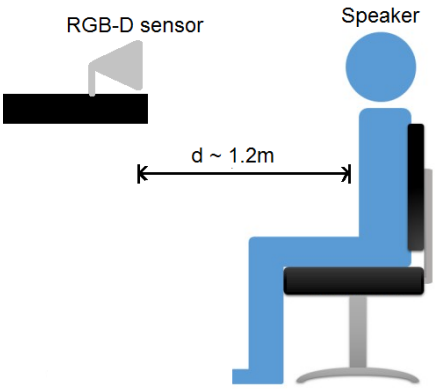


Fig. 1. Scenario created for recording the signs

Given the focus on facial expressions, nuiCaptureAnalyze was used to extract xy-coordinates of 121 points located across the face as in figure 2. These points served as the base descriptors for the face.

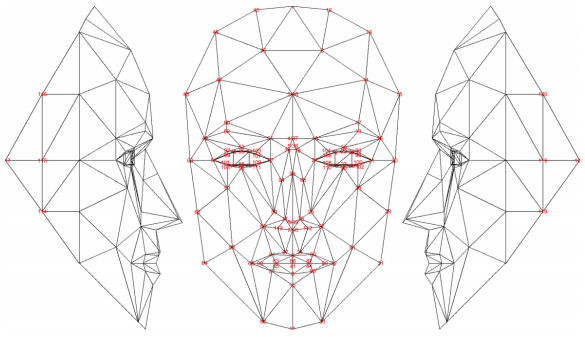


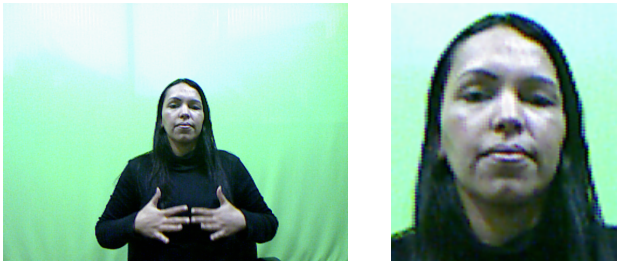
Fig. 2. Facial model used, with labeled points

III. Methodology

The following steps were defined so that the classification model was relevant to the available data extracted from the signs, with the objective of maximizing the model's accuracy. All the steps listed below were implemented in Matlab R2014a [7].

A. Detection of the region of interest

The original video contained view of the entire upper torso, so it was important to segregate the face specifically. This was done with the central pixel of the original frame as a reference, with the rectangular region cut out at a fixed coordinate. Figures 3a and 3b show a full frame and the separated area of interest respectively.



(a) Full frame (b) Selected face

Fig. 3. Facial selection

The face images are the inputs to the next step (Summarization) which will detect the most significant changes in facial expression.

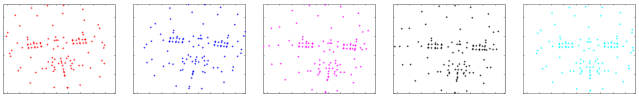
B. Summarization

This step was essential for the work, as it eliminates redundant frames allowing for a reduction of computational costs and more efficient feature extraction. Faced with the myriad of summarization techniques found in the literature, this study utilized the classic optimization problem, known as the Problem of Maximum Diversity [8], to extract the most relevant frames in the video.

After having selected the region of interest, the video recorded at approximately 30 fps was summarized through a process of selecting the n frames that contained the most diverse information. Based on the tests by [2], the specified value for n was five, such that each recording of each sign was summarized to a set of the five most significant frames as seen in figures 4 and 5. It is important to highlight that this summarization yielded feature vectors of equal size, regardless of the time that was taken to complete a given sign.

(a) 14 (b) 25 (c) 46 (d) 60 (e) 91

Fig. 4. The five most relevant frames extracted from a recording of the sign "to love"



(a) 14 (b) 25 (c) 46 (d) 60 (e) 91

Fig. 5. The 121 face points from the most relevant frames of the sign "to love"

C. Feature Vector

The objective of this step was to represent each sample in a manner that was robust and invariant to transformations. Each of the five frames returned from the summarization contained a descriptor formed from the concatenation of the xy-coordinates of the 121 facial points recorded by the nuiCaptureAnalyze software. Thus, the dimensions for a descriptor of a single frame are 1x242, with the following format:

$$D = \left[\begin{array}{cccccccc} x_1 & y_1 & x_2 & y_2 & \dots & x_{121} & y_{121} \end{array} \right]_{1x242}$$

The feature vector for a sample is formed through the concatenation of the descriptors of its five selected frames. Thus, the final representation of any sample is a vector of length 1210.

$$Vector = \left[\begin{array}{ccccc} D_1 & D_2 & D_3 & D_4 & D_5 \end{array} \right]_{1x1210}$$

D. Classification

The k-NN method [9] was used for the classification step, as it is the recommended classifier for a database with few samples.

To determine the class of a sample *m* not belonging to the training set, the k-NN classifier looks for the *k* elements of the training set that are closest to *m* and assigns its class based on which class represents the majority of these selected *k* elements.

Initially, the 10 recordings for each of the 10 signs were randomized in order to prevent that the same recordings be selected for the training or testing groups. 80% of the data was selected for training, and 20% for testing, such that each train group had 8 samples and test group had 2 samples for each sign.

With the selected training data, a cross-validation was used to find the value for *k* that provided the highest accuracy rate, the k_{best}. Thus, the training data were divided into 5-folds of the same size and 5 cross-validation iterations were performed. For each one, 1-fold was removed for testing and the remaining were used for training, as shown in figure 6.

The equation 1 shows the range of *k*. Given there were 100 samples in total, the tested values for *k* were 1 to 10.

$$1 \leq k \leq \sqrt{number \ \ of \ \ samples} \tag{1}$$

For each value of *k*, 5 iterations of cross-validation were performed and the average accuracy was obtained. The value for *k* that provided the best result was used for the group test.

The average accuracy (acc_{avg}) and standard deviation (σ) for the testing set were obtained for the 10 iterations of the classification algorithm as shown in *Algorithm 1*. The metric distance used in k-NN method was the Euclidean distance.

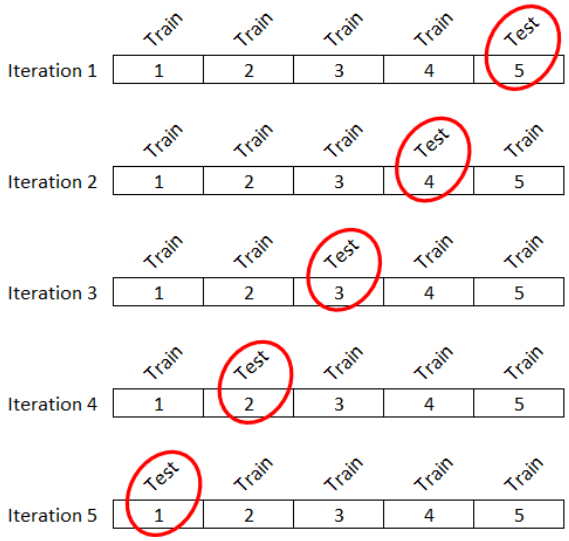


Fig. 6. Process for cross-validation using 5-folds

Algorithm 1: K-NN CLASSIFICATION

Input: Sign samples
Output: acc_{avg} and σ of the 10 iterations

Start
  for *w = 1 to 10* **do**
    Randomizes the samples of each sign
    train $\leftarrow$ 80% of the data
    test $\leftarrow$ 20% of the data
    for *k = 1 to 10* **do**
      $testV \leftarrow$ CROSS-VALIDATION(5-fold)
      $acc(k) \leftarrow$ K-NN$(testV, k)$
    end
    $[acc \ ind] \leftarrow max(acc)$
    $k_{best} \leftarrow ind$
    $acc_{test}(w) \leftarrow$ K-NN$(test, k_{best})$
  end
  $acc_{avg} \leftarrow mean(acc_{test}(w))$
  $\sigma \leftarrow std(acc_{test}(w))$
end
retorn acc_{avg}, σ

IV. Experiments and Results

It is known in literature that during the sign acquisition, distortions (offset, warping, etc.) can arise in different recordings of a same sign. This mainly occurs due to the natural displacement of the speaker's face during the sign recording. In order to overcome this problem and obtain samples invariant to distortions, some transformation procedures were applied on raw data. The experimental datasets considered in this study are described as follows.

First Experiment (EX.1): The first experiment consisted of the implementation of the methodology described throughout the paper, without any modification of the sign descriptors. In other words, the classification was performed with the raw data.

Second Experiment (EX.2): In the second experiment there was a processing of the database. For each recording, Z-Score Normalization was applied to all 5 frames of each point, based on equations 2 and 3.

$$x_{new} = \frac{x - \bar{x}}{\sigma(x)} \tag{2}$$

$$y_{new} = \frac{y - \bar{y}}{\sigma(y)} \tag{3}$$

Third Experiment (EX.3): In experiment 3, the data from each frame were updated according to equations 4 and 5. Using centroid normalization, each point was represented with reference to the mean point for that frame.

$$x_{newpointP} = x_{pointP} - \bar{x}_{frame} \tag{4}$$

$$y_{newpointP} = y_{pointP} - \bar{y}_{frame} \tag{5}$$

Table I contains the summary of the results from each of the experiments, as well as the values for k_{best} obtained from the 10 iterations of the algorithm. In figure 7, it is possible to compare the distribution of the percent accuracies for each of the three experiments.

TABLE I
RESULTS AFTER 10 EXECUTIONS OF THE CLASSIFICATION ALGORITHM.

Data	**Average Accuracy**	σ	k_{best}
EX.1	84%	8,76	1
EX.2	79%	6.99	1, 2 e 4
EX.3	83%	10.33	1

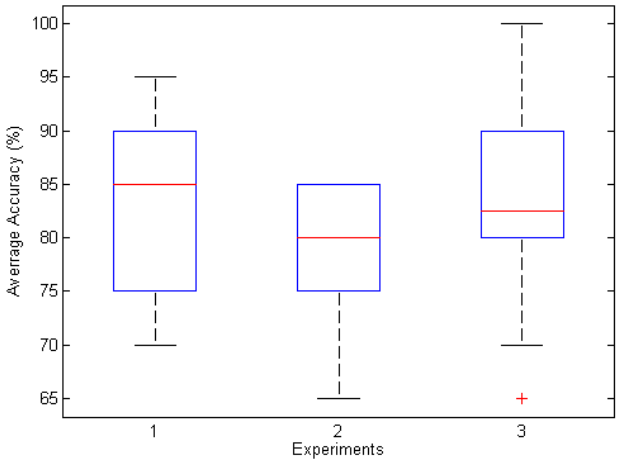


Fig. 7. Box plot of the percent accuracies for the three conducted classification rounds

V. CONCLUSIONS AND FUTURE WORK

The BSL recognition is a challenge problem. It is an area still in development that currently has no robust system for the classification of signs, such as phonemes, phrases or even a conversation, due to limitations resulting from the lack of a database with signs in BSL well structured. Despite these difficulties, working toward the development of a robust system is highly motivating, given the social impact that a system of this complexity can achieve.

In this paper, attention was taken to standardize the recording of all signs, randomize the sample to avoid bias in the data, in addition to following the statistical guidelines for choosing the best k.

With the aim of correctly classifying 10 signs, it was found that the methodology adopted had a considerable performance, achieving a maximum average accuracy of 84%. In addition, the system was shown to be robust when dealing with possible shifting of the face between different samples.

For future work, one of the intentions it to apply an SVM Multi-class classifier, adjusting the cost parameter C, that determines a balance between maximizing the margin and minimizing the misclassification [10], and the parameter γ, gamma of the kernel function. Another objectives are to verify the importance of the information about depth and to perform a selection of features reducing the dimensionality of the data.

ACKNOWLEDGMENT

The authors of this article would like to thanks PPGEE-UFMG for the incentive and guidance. The present work was completed with the financial support of CAPES - Brazil.

REFERENCES

[1] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Patter Recognition 36*, pp. 259–275, 2002.
[2] S. G. M. Almeida, "Extração de características em reconhecimento de parâmetros fonológicos da língua brasileira de sinais utilizando sensores rgb-d," Ph.D. dissertation, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil, 2014.
[3] Microsoft Windows, "Kinect." [Online]. Available: https://developer.microsoft.com/en-us/windows/kinect
[4] ——, "nuicaptureanalyse." [Online]. Available: http://nuicapture.com/download-trial/
[5] D. Hamester, P. Barros, and S. Wermter, "Face expression recognition with a 2-channel convolutional neural network," *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1 – 8, 2015.
[6] F. J. C. Pedroso and E. O. Salles, "Reconhecimento de expressões faciais baseado em modelagem estatística," *XIX Congresso Brasileiro de Automática*, pp. 631–638, 2012.
[7] MathWorks, "Matlab r2014a."
[8] C. C. Kuo, F. Glover, and K. S. Dhir, "Analyzing and modeling the maximum diversity problem by zero-one programing," *Decision Sciences*, vol. 24, no. 6, pp. 1171–1185, 1993.
[9] E. Patrick and F. Fischer, "A generalized k-nearest neighbor rule," *Elsevier*, vol. 16, no. 2, pp. 128–152, 1970.
[10] M. H. Granzotto and L. C. Oliveira-Lopes, "Desenvolvimento de sistema de detecção de falhas baseado em aprendizado estatístico de máquinas de vetores de suporte," *XX Congresso Brasileiro de Engenharia Química*, pp. 1–10, 2014.

Apêndice B

Execução dos sinais

Tabela B.1: Descrição para execução de cada sinal.

Sinal	**Como executar**
Acalmar	Mãos verticais abertas, palmas para frente, inclinadas para baixo, mão direita atrás da esquerda. Afastá-las para os lados opostos, movendo-as ligeiramente para baixo.
Acusar	Mão em '4', palma para a esquerda, ponta do indicador tocando a ponta do nariz. Movê-la para sempre. Expressão facial opcional.
Aniquilar	Mão esquerda aberta, palma para cima; mão direta aberta, palma para baixo, tocando a palma esquerda. Girar a mão direita pelo pulso, com força, para frente e para a direita, e então movê-la para a direita, com expressão facial contraída.
Apaixonado	Mão vertical aberta, palma pra trás, dedos separados e curvados, diante da boca aberta e com a língua para fora. Mover a mão em círculos verticais para a esquerda (sentido anti-horário), com os olhos semiabertos.
Engordar	Com as mãos fechadas palma a palma, próximas uma a outra, movê-las para os lados opostos com a bochechas infladas.
Felicidade	Mão horizontal aberta, palma pra trás, tocando o peito. Movê-la num circuito vertical para a esquerda (sentido anti-horário).
Magro	Mão fechada com o dedo mínimo estendido, palma para trás, ao lado direito da face enquanto as bochechas são sugadas
Sortudo	Mãos em L horizontal, palmas das mãos para trás. Balançá-las para cima e para baixo.
Surpresa	A mão se move num arco para trás e para cima, toca o peito e continua o arco para frente, pairando defronte o rosto sorridente.
Zangado	Mão horizontal aberta, palma para trás, dedos curvados tocando o peito. Movê-las para cima e para baixo, com a testa franzida.

Fonte: Capovilla et al. (2012a) e Capovilla et al. (2012b)

Apêndice C

Sumarização

C.1 Sinal Acalmar

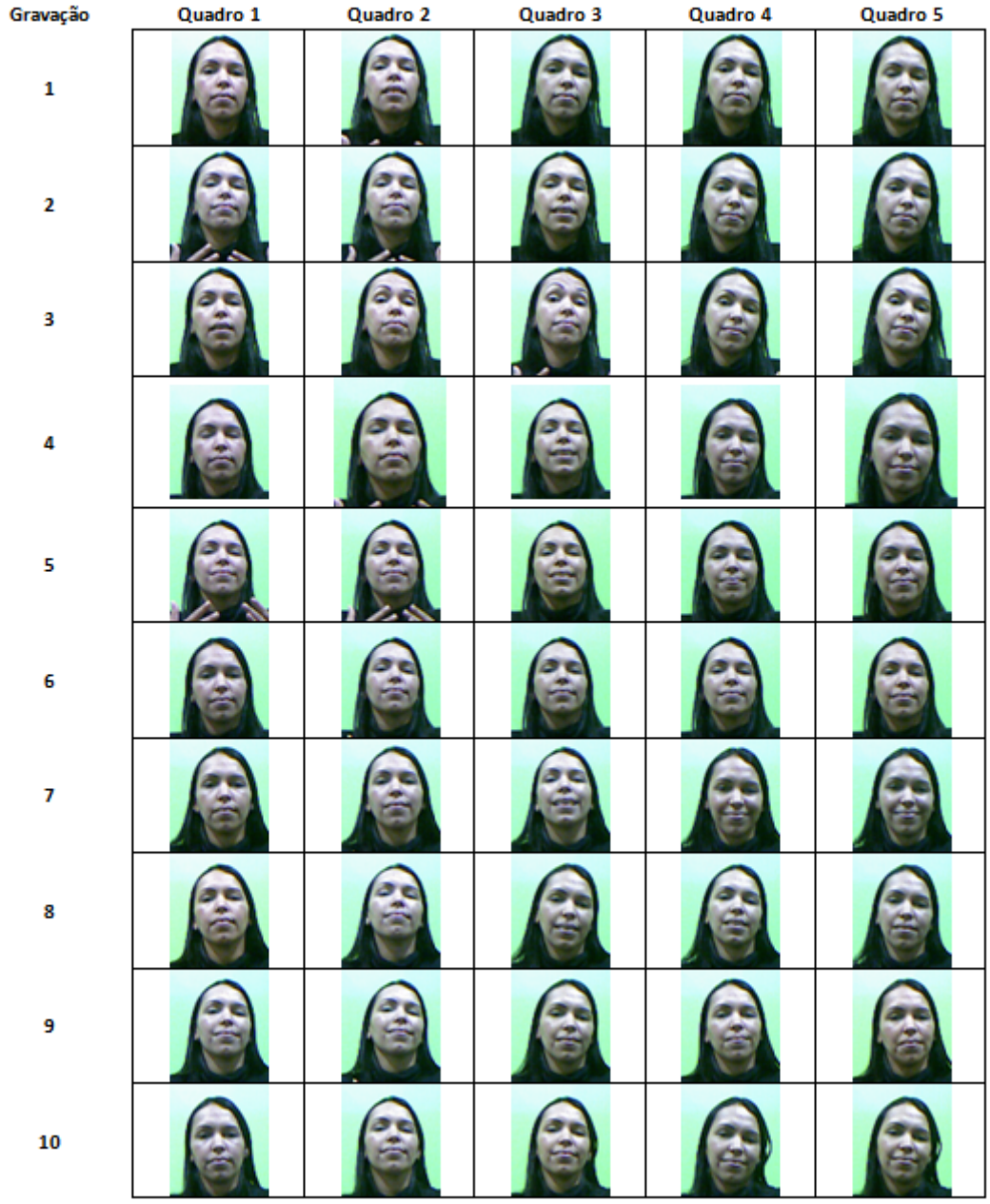


Figura C.1: 5 quadros significativos de cada gravação do sinal Acalmar.

C.2 Sinal Acusar

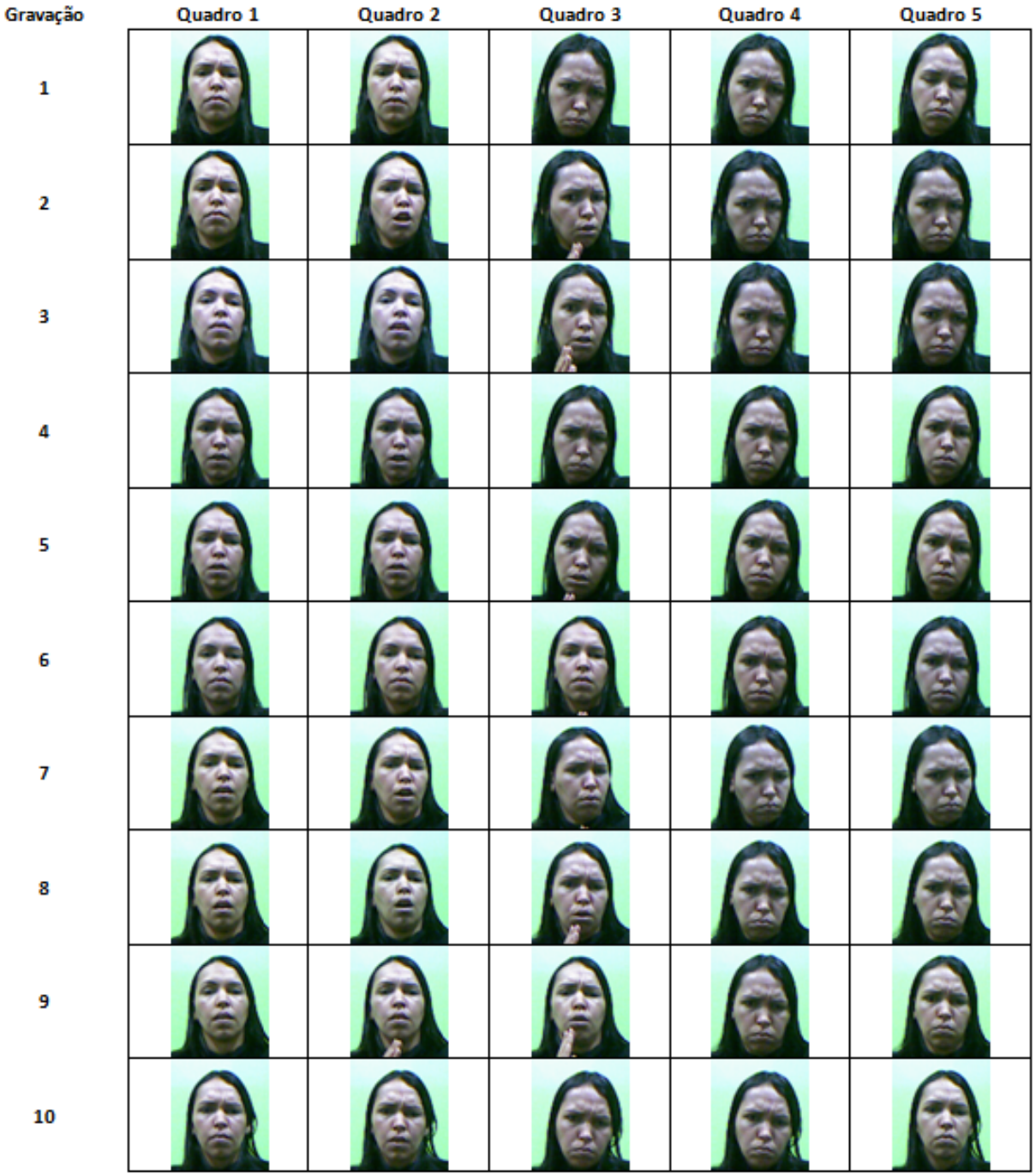


Figura C.2: 5 quadros significativos de cada gravação do sinal Acusar.

C.3 Sinal Aniquilar

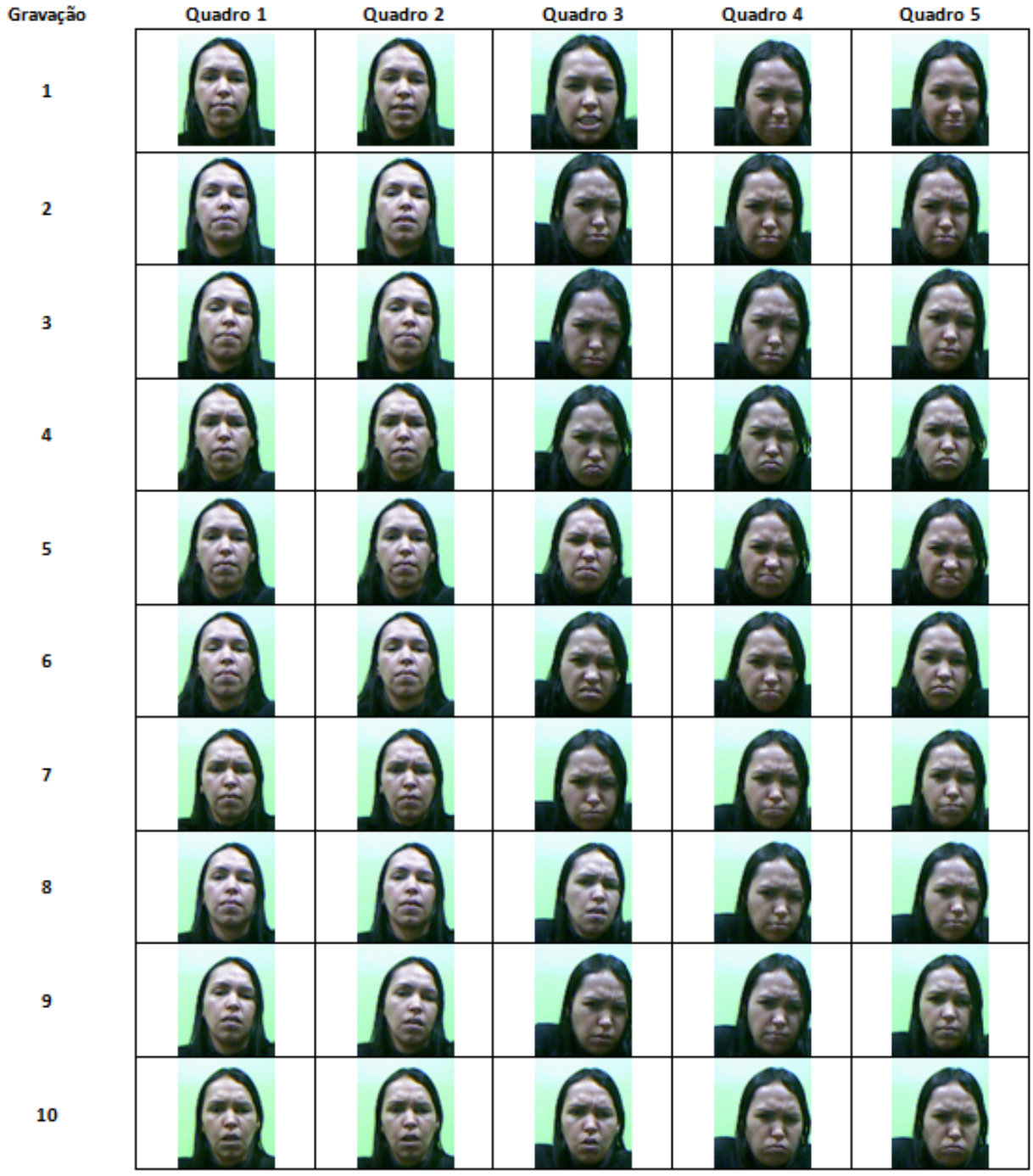


Figura C.3: 5 quadros significativos de cada gravação do sinal Aniquilar.

C.4 Sinal Apaixonado

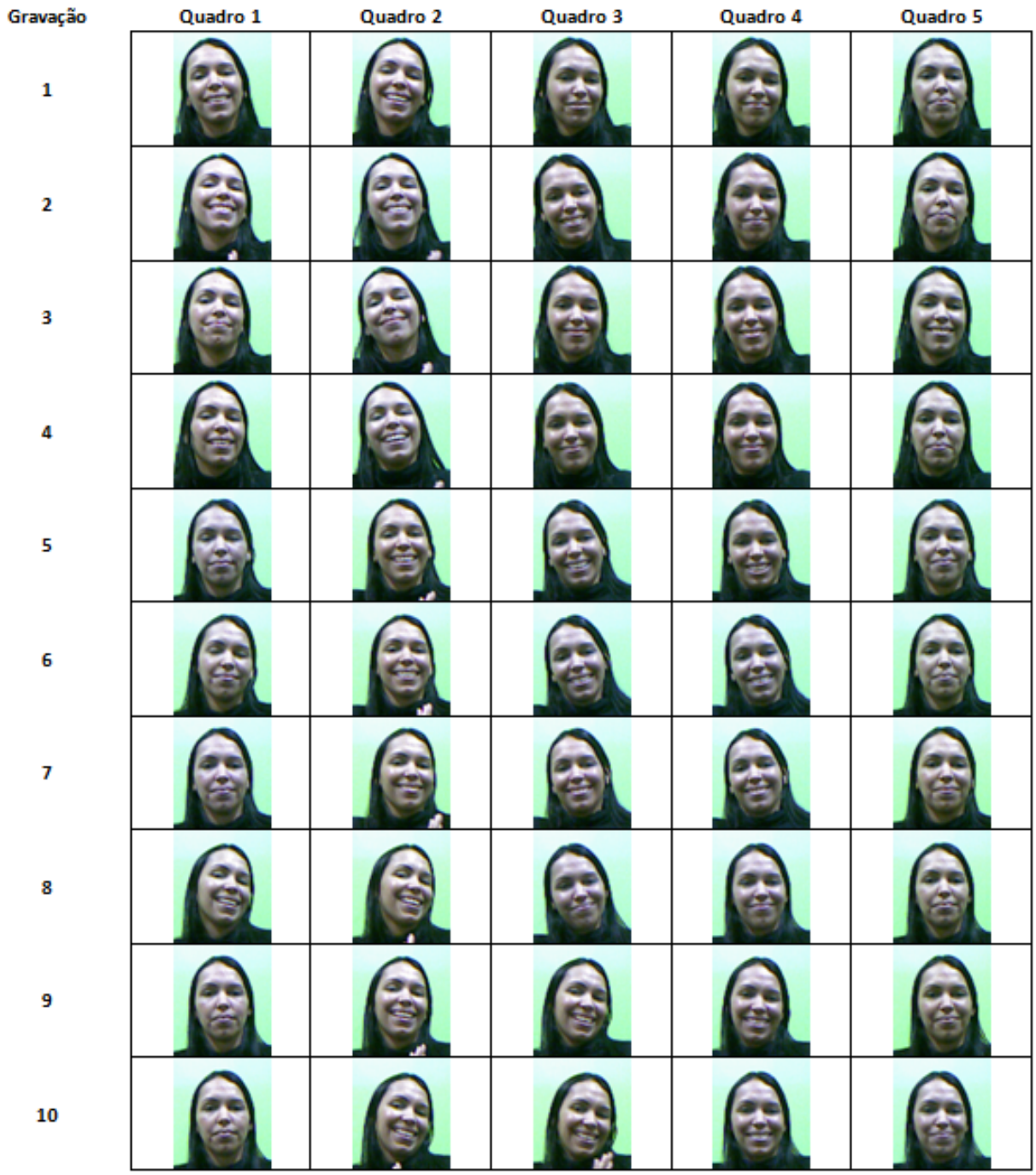


Figura C.4: 5 quadros significativos de cada gravação do sinal Apaixonado.

C.5 Sinal Engordar

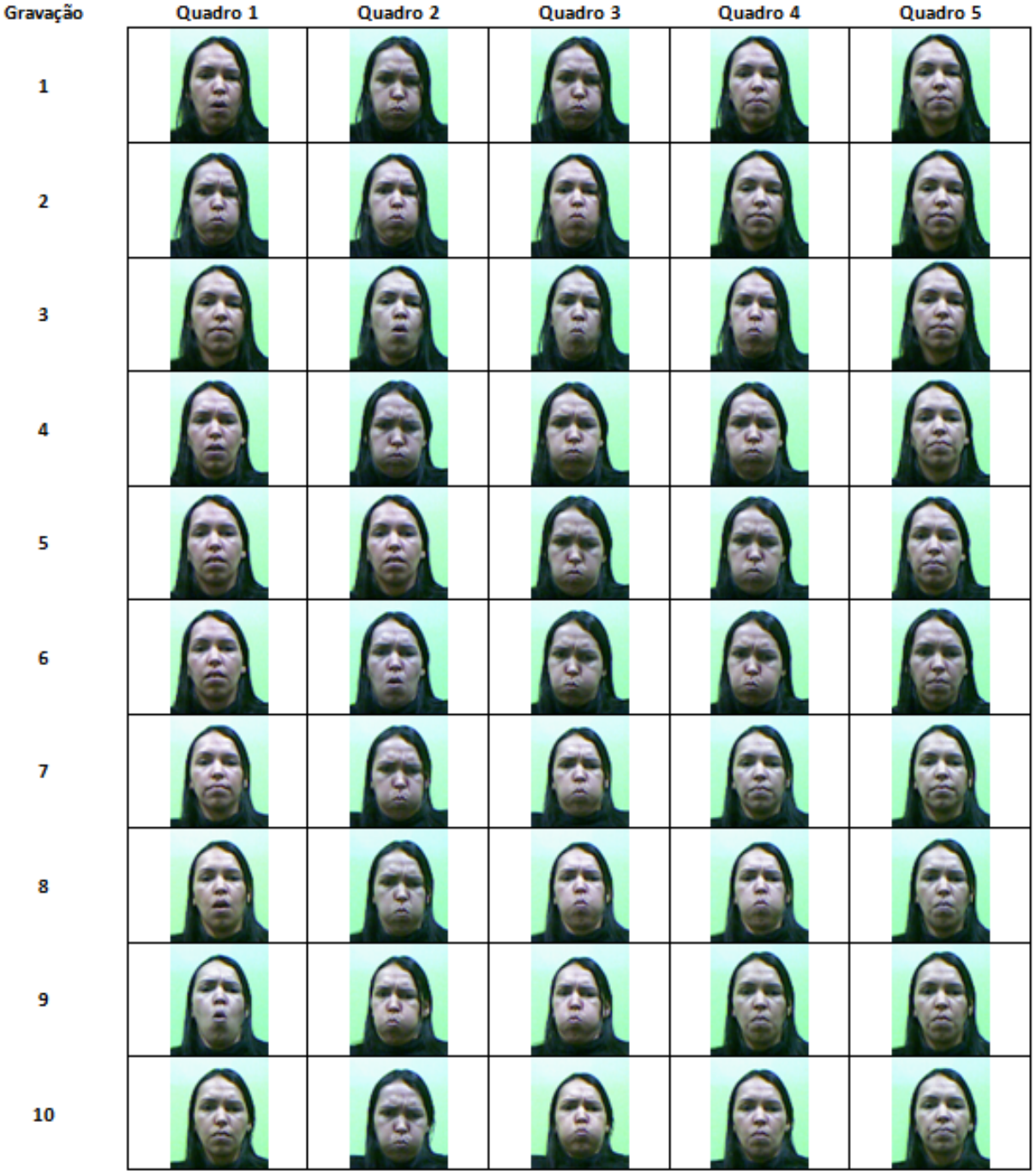


Figura C.5: 5 quadros significativos de cada gravação do sinal Engordar.

C.6 Sinal Felicidade

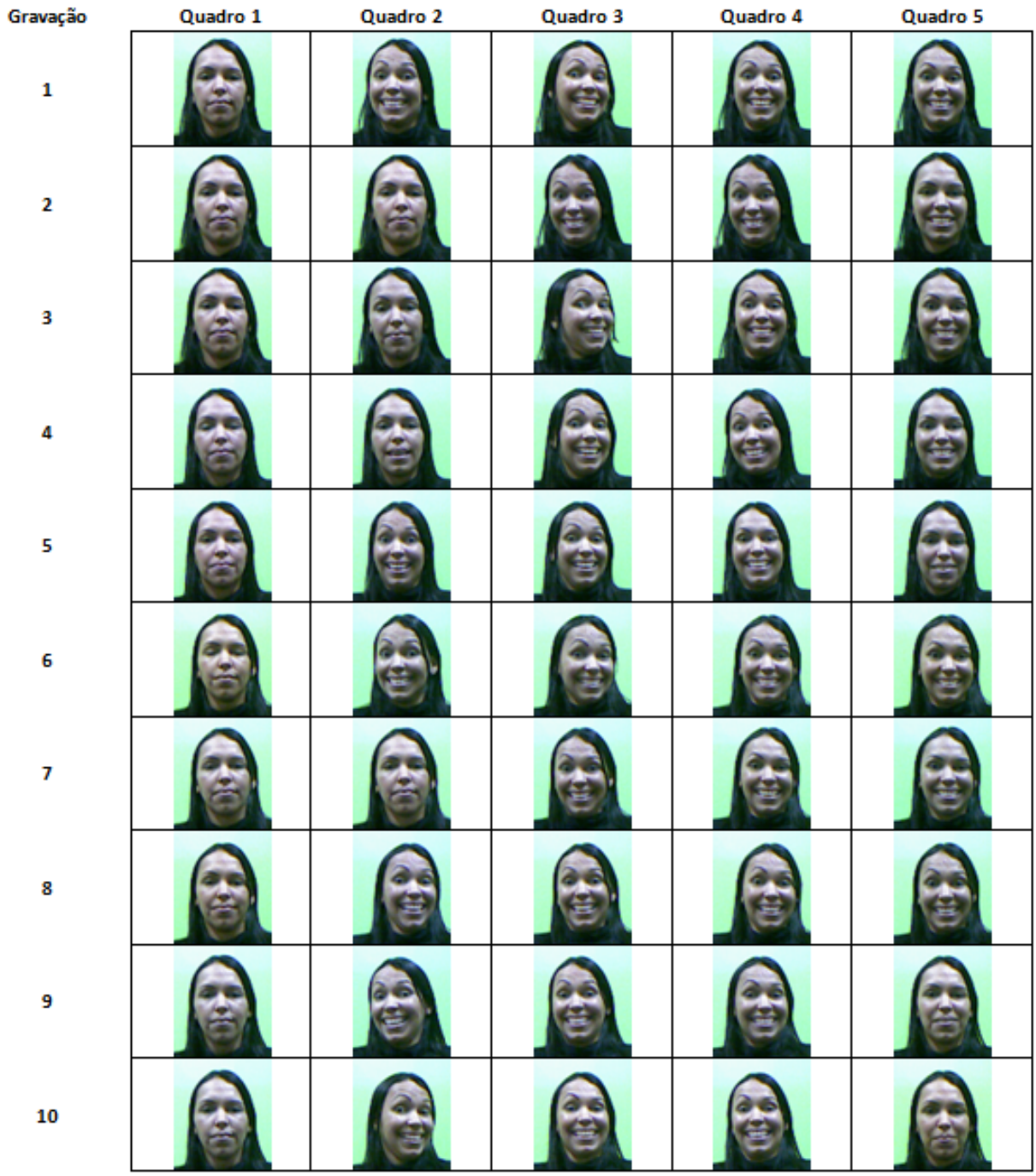


Figura C.6: 5 quadros significativos de cada gravação do sinal Felicidade.

C.7 Sinal Magro

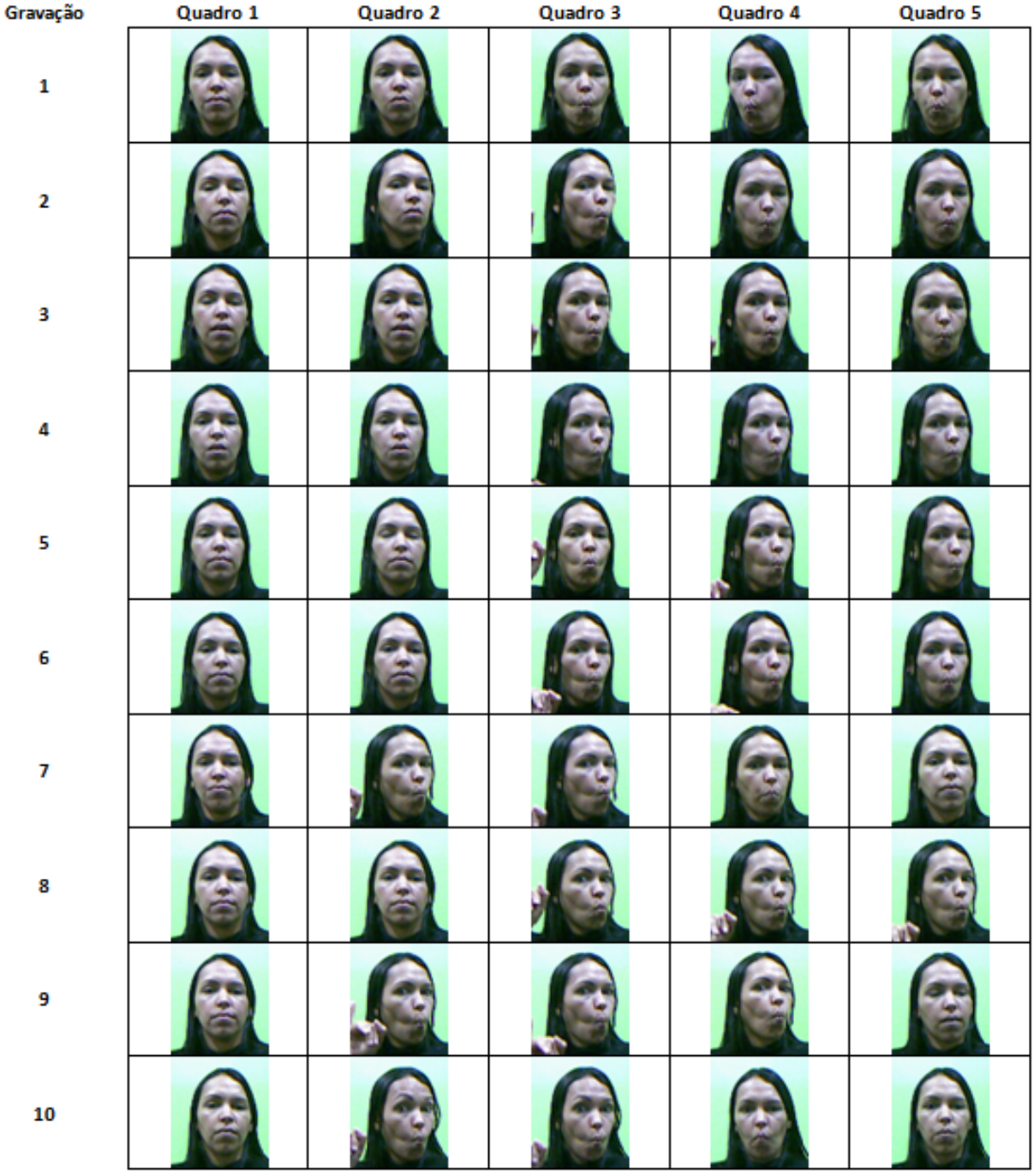


Figura C.7: 5 quadros significativos de cada gravação do sinal Magro.

C.8 Sinal Sortudo

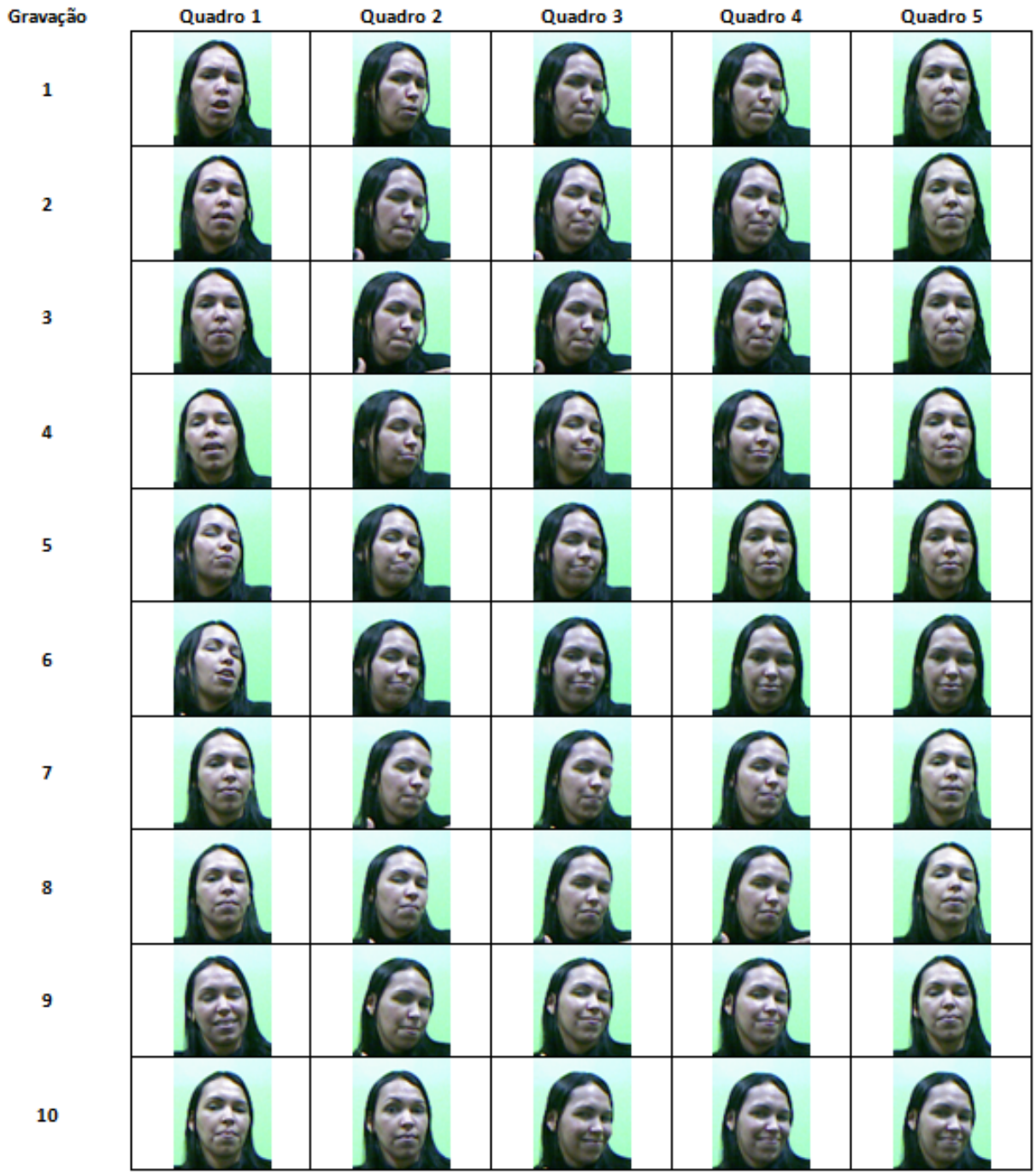


Figura C.8: 5 quadros significativos de cada gravação do sinal Sortudo.

C.9 Sinal Surpresa

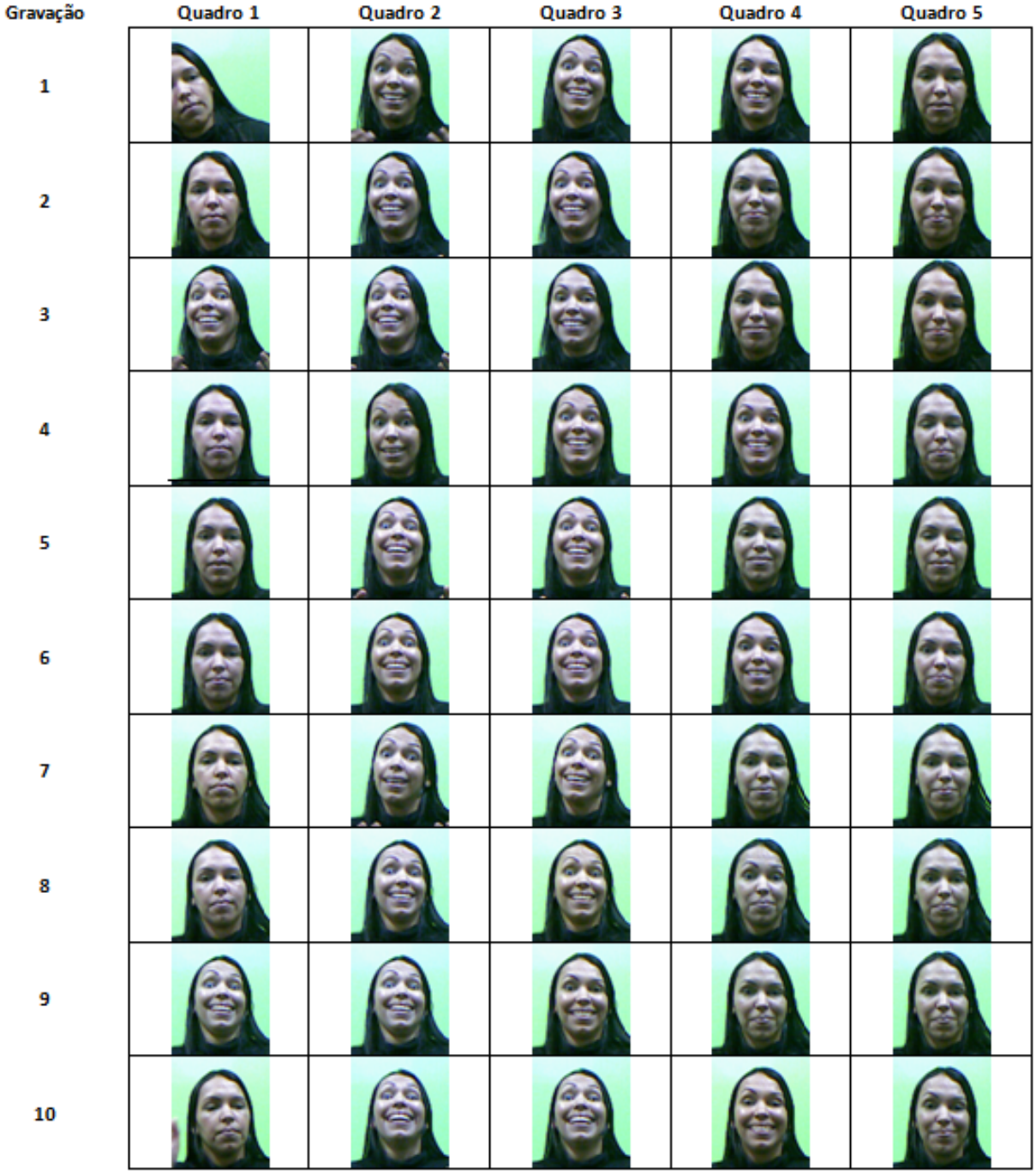


Figura C.9: 5 quadros significativos de cada gravação do sinal Surpresa.

C.10 Sinal Zangado

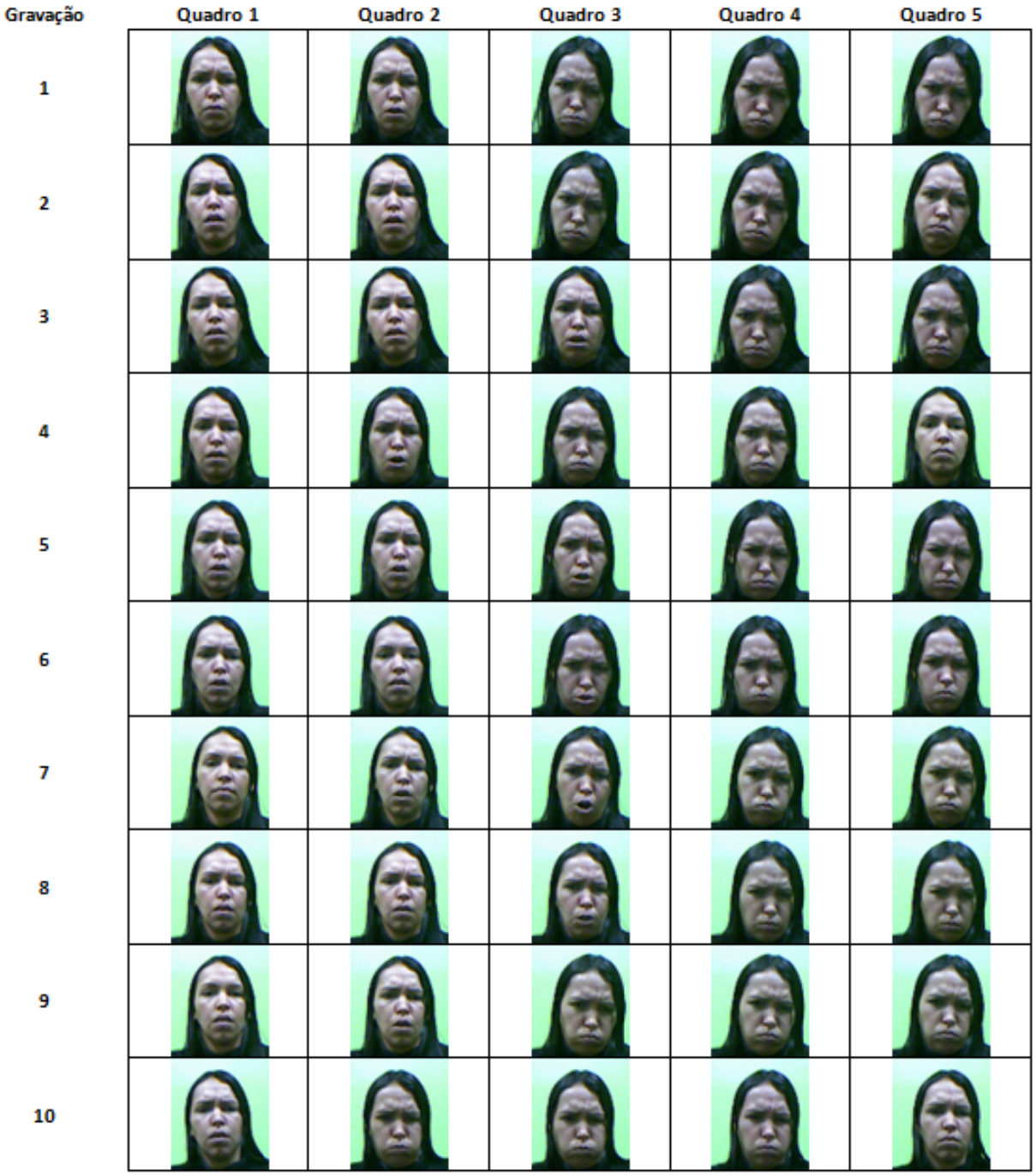


Figura C.10: 5 quadros significativos de cada gravação do sinal Zangado.

Apêndice D

Parâmetros

Tabela D.1: Resultados e parâmetros das 30 iterações nas implementações que apresentaram a maior taxa média de acerto para cada uma das configurações: vetor de características composto pelas coordenadas cartesianas e vetor composto pela aplicação do operador LBP, ambas classificadas pelo SVM.

	Pontos + SVM			**LBP + SVM**		
Iteração	**Taxa de acerto**	**C**	γ	**Taxa de acerto**	**C**	γ
1	80	32	$2{,}44 * 10^{-4}$	100	8	$2{,}44 * 10^{-4}$
2	70	16	$1{,}22 * 10^{-4}$	100	8	$2{,}44 * 10^{-4}$
3	85	8	$4{,}88 * 10^{-4}$	100	4	$4{,}88 * 10^{-4}$
4	80	32	$1{,}22 * 10^{-4}$	90	4	$2{,}44 * 10^{-4}$
5	80	128	$6{,}10 * 10^{-5}$	95	4	$4{,}88 * 10^{-4}$
6	75	4	$9{,}76 * 10^{-4}$	95	4	$4{,}88 * 10^{-4}$
7	75	64	$6{,}10 * 10^{-5}$	95	16	$6{,}10 * 10^{-5}$
8	85	2	$1{,}95 * 10^{-3}$	95	8	$2{,}44 * 10^{-4}$
9	85	4	$9{,}76 * 10^{-4}$	90	4	$4{,}88 * 10^{-4}$
10	90	2	$1{,}95 * 10^{-3}$	95	16	$1{,}22 * 10^{-4}$
11	80	128	$6{,}10 * 10^{-5}$	100	16	$1{,}22 * 10^{-4}$
12	75	16	$4{,}88 * 10^{-4}$	95	8	$1{,}22 * 10^{-4}$
13	90	2	$1{,}95 * 10^{-3}$	100	16	$1{,}22 * 10^{-4}$
14	90	32	$1{,}22 * 10^{-4}$	95	8	$1{,}22 * 10^{-4}$
15	90	8	$9{,}76 * 10^{-4}$	100	8	$2{,}44 * 10^{-4}$
16	90	2	$9{,}76 * 10^{-4}$	85	8	$2{,}44 * 10^{-4}$
17	90	128	$3{,}05 * 10^{-5}$	100	8	$2{,}44 * 10^{-4}$
18	90	128	$1{,}22 * 10^{-4}$	100	4	$4{,}88 * 10^{-4}$
19	90	4	$9{,}76 * 10^{-4}$	100	8	$2{,}44 * 10^{-4}$
20	80	128	$1{,}22 * 10^{-4}$	95	8	$1{,}22 * 10^{-4}$
21	85	16	$1{,}22 * 10^{-4}$	100	16	$1{,}22 * 10^{-4}$
22	80	2	$1{,}95 * 10^{-3}$	85	4	$4{,}88 * 10^{-4}$
23	70	2	$1{,}95 * 10^{-3}$	90	8	$1{,}22 * 10^{-4}$
24	85	8	$9{,}76 * 10^{-4}$	100	16	$1{,}22 * 10^{-4}$
25	75	32	$2{,}44 * 10^{-4}$	95	4	$4{,}88 * 10^{-4}$
26	90	8	$4{,}88 * 10^{-4}$	95	8	$2{,}44 * 10^{-4}$
27	80	64	$1{,}22 * 10^{-4}$	90	4	$2{,}44 * 10^{-4}$
28	70	2	$9{,}76 * 10^{-4}$	85	8	$1{,}22 * 10^{-4}$
29	80	512	$3{,}05 * 10^{-5}$	95	4	$2{,}44 * 10^{-4}$
30	90	128	$3{,}05 * 10^{-5}$	100	8	$2{,}44 * 10^{-4}$