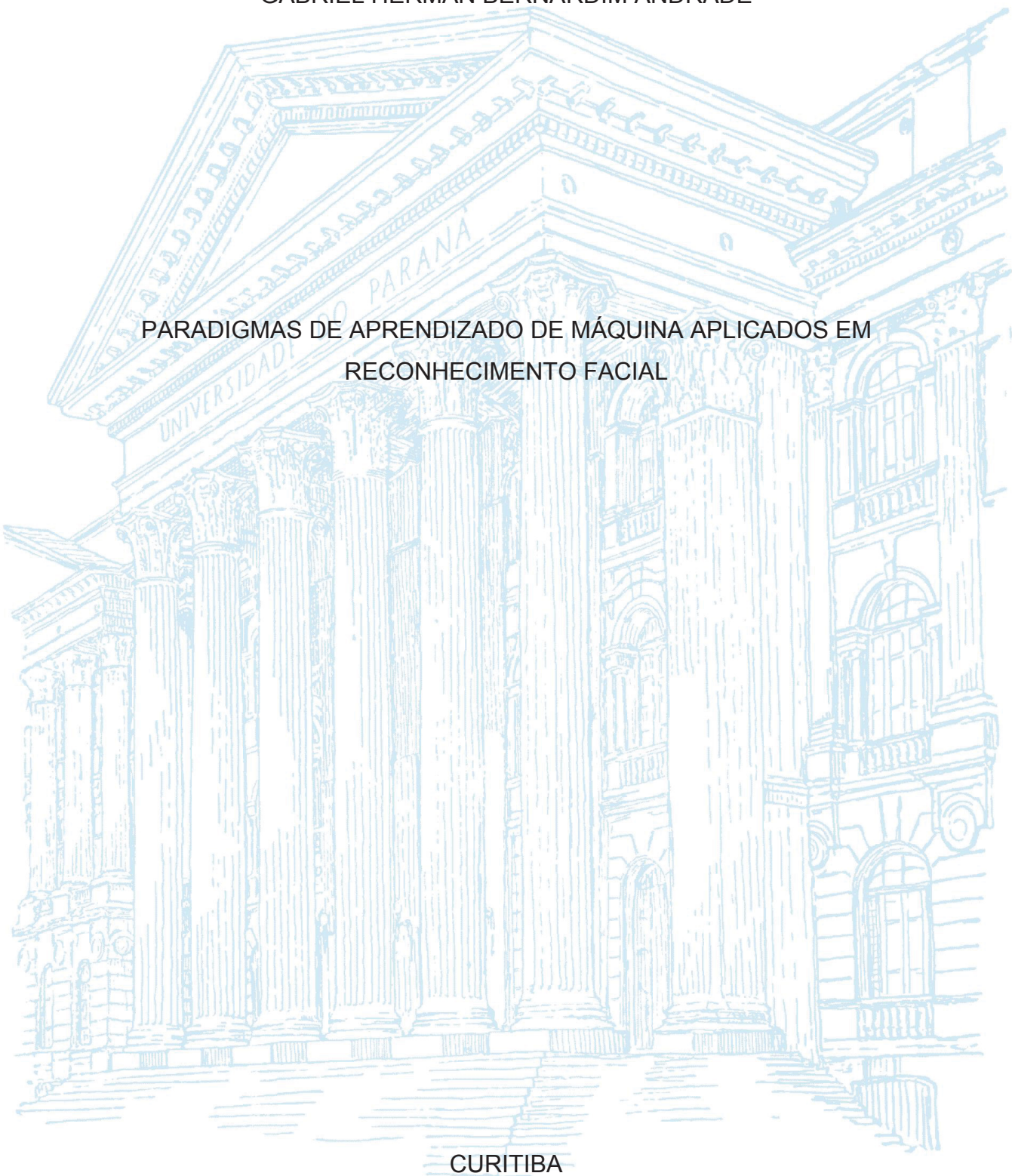
UNIVERSIDADE FEDERAL DO PARANÁ

GABRIEL HERMAN BERNARDIM ANDRADE



PARADIGMAS DE APRENDIZADO DE MÁQUINA APLICADOS EM RECONHECIMENTO FACIAL

CURITIBA

2019

GABRIEL HERMAN BERNARDIM ANDRADE

PARADIGMAS DE APRENDIZADO DE MÁQUINA APLICADOS EM RECONHECIMENTO FACIAL

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Área de concentração em Sistemas Eletrônicos, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como parte das exigências para obtenção do título de mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Leandro dos Santos Coelho.

CURITIBA

2019

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

A553p

Andrade, Gabriel Herman Bernardim
Paradigmas de aprendizado de máquina aplicados em reconhecimento facial [recurso eletrônico] / Gabriel Herman Bernardim Andrade. – Curitiba, 2019.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós- Graduação em Engenharia Elétrica, 2019.

Orientador: Leandro dos Santos Coelho.

1. Aprendizado do computador. 2. Kinect (Controlador programável). 3. Expressão facial. 4. Algorítmos computacionais. 5. Redes neurais (Computação). I. Universidade Federal do Paraná. II. Coelho, Leandro dos Santos. III. Título.

CDD: 006.42

Bibliotecária: Vanusa Maciel CRB- 9/1928

MINISTÉRIO DA EDUCAÇÃO
SETOR DE TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA ELÉTRICA - 40001016043P4

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **GABRIEL HERMAN BERNARDIM ANDRADE** intitulada: **PARADIGMAS DE APRENDIZADO DE MÁQUINA APLICADOS EM RECONHECIMENTO FACIAL**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 24 de Junho de 2019.

LEANDRO DOS SANTOS COELHO
Presidente da Banca Examinadora (UFPR)

JULIANA ALMANSA MALAGOLI
Avaliador Interno (UFPR)

JULIO CÉSAR NIEVOLA
Avaliador Externo (PUC/PR)

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, por me dar a capacidade e as condições de trabalhar e ser persistente para alcançar todos os objetivos pretendidos neste trabalho.

Agradeço aos meus pais Edilson e Irene, pelo apoio e motivação durante o desenvolvimento deste trabalho. Aos meus irmãos, Vinicius, Matheus e Sofia, pelos momentos de diversão e relaxamento que foram muito importantes para não sucumbir ao estresse e ao cansaço. Em especial à minha avó Irene, por tudo que faz por nós, mesmo com idade avançada.

Gostaria também de agradecer aos meus amigos, João, Jorge e Murilo, pelo apoio, sugestões, conselhos (e pelas noites jogando videogame) que foram de muita ajuda para o desenolvimento deste trabalho.

E por último, mas não menos importante, quero muito agradecer ao meu orientador, Leandro dos Santos Coelho, que me acompanha desde a minha graduação, pelo suporte, conselhos e, principalmente, pela paciência durante toda esta jornada.

“Quando a mente muda, a gente anda pra frente.
E quando a gente manda ninguém manda na gente.
Na mudança de atitude não há mal que não se mude
nem doença sem cura.
Na mudança de postura a gente fica mais seguro.
Na mudança do presente a gente molda o futuro!”

(Gabriel, o Pensador, 2001)

RESUMO

As expressões faciais desempenham um papel significativo na interação interpessoal, sendo capazes de exprimir estado emocional, veracidade e adicionar contexto à comunicação verbal. O reconhecimento automático de expressões faciais ainda é um desafio para os computadores, visto que é complicado identificar e separar as características relevantes de cada expressão. Quando lidando com imagens 2D, problemas tais como diferenças de iluminação, posição e oclusão facial são empecilhos para o bom desempenho destes sistemas. Na tentativa de contornar este problema, vários métodos que utilizam modelos 3D da face foram propostos. Entretanto, sensores 3D de alta resolução continuam tendo custo elevado e alto tempo de captura para aquisição de imagens, o que inviabiliza a aplicação desta tecnologia em sistemas de tempo real. O sensor Kinect da Microsoft® se apresenta como uma alternativa barata e rápida para aquisições de imagens de profundidade, porém as imagens por esta capturadas possuem menor resolução e maior nível de ruído, o que pode resultar em falha na captura de características e informações relevantes para o modelamento das emoções faciais. A transferência do conhecimento adquirido por meio do treinamento de um algoritmo sobre dados em alta resolução para a aplicação em imagens adquiridas com o sensor Kinect pode contribuir para a minimização deste tipo de problema. Neste sentido, o objetivo desta dissertação é desenvolver um sistema capaz de reconhecer seis expressões faciais básicas, por meio de imagens em 3D adquiridas por um sensor Kinect, classificadas por modelos de Aprendizado de Máquina treinados sobre a base de dados Bosphorus DB, obtida por um *scanner* 3D de alta resolução. A geração dos modelos de aprendizado sobre a base de dados foi realizada por meio da avaliação de desempenho de três abordagens de extração de características, por meio de geometria (Pontos Fiduciais Faciais), textura (Padrões Binários Locais) e camadas de convolução. Os vetores de características extraídos da base de dados foram empregados para treinar e comparar o desempenho de quatro algoritmos de Aprendizado de Máquina nesta tarefa, Máquina de Vetores de Suporte, K-Vizinhos mais Próximos, Redes Neurais Artificiais e Comitês de Máquinas. A abordagem utilizando uma adaptação da Rede Neural Convolucional AlexNet para trabalhar com imagens RGB-D obteve a melhor desempenho de classificação sobre a base de dados, atingindo 86.67% de precisão. Este modelo foi então adaptado para a classificação das imagens capturadas pelo Kinect, sendo capaz de identificar 72,62% das expressões faciais corretamente.

Palavras-chave: Aprendizado de Máquina. Kinect. Computação Afetiva. Reconhecimento de expressões faciais. RGB-D.

ABSTRACT

Facial expressions play a significant role in interpersonal interaction, being able to express emotional state, veracity and add context to the verbal communication. Automatic facial expression recognition is still a challenge for computers, since it is complicated to identify and isolate relevant characteristics for each expression. When dealing with 2D images, problems such as dynamic lighting, position and facial occlusion are burdens in order for this kind of system to achieve good performance. As an attempt to circumvent this problem, several methods that use 3D face models were proposed. However, high-resolution 3D sensors continue to have high cost and long capture time, which makes it unfeasible to be applied to real-time systems. The Microsoft® Kinect is a fast and inexpensive alternative for depth image acquisition, but the images it captures have poor resolution and higher noise levels, which can result in failure to capture relevant features and information required to model facial emotions. The transfer of the acquired knowledge through the training of an algorithm on high resolution data to be applied on a set of images captured by the Kinect can contribute to the mitigation of this kind of problem. In this sense, the objective of this dissertation is to develop a system capable of recognizing six basic facial expressions through 3D images acquired by a Kinect device, classified by Machine Learning models trained over high resolution 3D scanner data, provided by the Bosphorus database. The generation of the learning models over the database images was performed through the evaluation of three kind of facial features extraction, through geometry (Facial Fiducial Points), texture (Local Binary Patterns) and convolutional layers. Feature vectors extracted from the database were used to train and compare the performance of four Machine Learning algorithms for this task, Support Vector Machines, K-Nearest Neighbors, Artificial Neural Networks and Ensembles. The approach using an adapted AlexNet Convolutional Neural Network, able to process RGB-D images, obtained the best classification performance applied to the database, reaching an accuracy of 86.67%. This model was then adapted to the classification of the images captured by the Kinect, being able to identify 72.62% of the facial expressions correctly.

Keywords: Machine Learning. Kinect. Affective Computing. Facial Expression Recognition. RGB-D.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

2D	Duas dimensões ou Bidimensional
3D	Três dimensões ou Tridimensional
AM	Aprendizado de Máquina
CNN	Rede Neural Convolucional (do inglês, *Convolutional Neural Network*)
CPU	Unidade Central de Processamento (do inglês, *Central Processing Unit*)
DRHP	Padrões de Histograma de Classificação Direcional (do inglês, *Directional Rank Histogram Pattern*)
E3	*Electronic Entertainment Expo*
FACS	Sistema de Codificação de Ação Facial (do inglês, *Facial Action Coding System*)
FPS	Quadros por Segundo (do inglês, *Frames per Seconds*)
GPU	Unidade de Processamento Gráfico (do inglês, *Graphics Processing Unit*)
HAOG	Histogramas de Gradientes Médios Orientados (do inglês, *Histogram of Average Oriented Gradients*)
HMM	Modelo Oculto de Markov (do inglês, *Hidden Markov Model*)
IA	Inteligência Artificial
K-NN	K-Vizinhos mais próximos (do inglês, *K-Nearest Neighbours*)
LBP	Padrão Local Binário (do inglês, *Local Binary Pattern*)
LDA	Análise Discriminante Linear (do inglês, *Linear Discriminant Analysis*)
LDSP	Padrões de Intensidade Direcional Local (do inglês, *Local Directional Strength Pattern*)
LFDA	Análise Discriminante Local de Fisher (do inglês, *Local Fisher Discriminant Analysis*)
LLE	Incorporação Linear Local (do inglês, *Locally Linear Embedding*)
LNBP	Padrão Local Normal Binário (do inglês, *Local Normal Binary Pattern*)

LPP	Projeção de Preservação de Localidade (do inglês, *Locality Preserving Projections*)
LRN	Normalização de Resposta Local (do inglês, *Local Response Normalization*)
LSTM	Memória de Longo Prazo (do inglês, *Long Short-Term Memory*)
LTSA	Alinhamento de espaço tangente local (do inglês, *Local tangent space alignment*)
MLP	Perceptron Multi Camada (do inglês, *Multi Layer Perceptron*)
PCA	Análise de Componentes Principais (do inglês, *Principal Component Anayisis*)
RELU	Unidade Linear Retificada (do inglês, *Rectified Linear Unit*)
RGB	Vermelho, Verde, Azul (do inglês, *Red, Green, Blue*)
RGB-D	Vermelho, Verde, Azul e Profundidade (do inglês, *Red, Green, Blue and Depth*)
RNA	Rede Neural Artificial
SDK	Kit de desenvolvimento de *software* (do inglês, *Software Development Kit*)
SOMs	Mapas auto-organizados (do inglês, *Self Organizing Maps*)
SRC	Classificação baseada em representação esparsa (do inglês, *Sparse Representation-based Classification*)
SVD	Decomposição de valor singular (do inglês, *Singular-Value Decomposition*)
SVM	Máquinas de vetores de suporte (do inglês, *Support Vector Machines*)
SVR	Regressão de vetores de suporte (do inglês, *Support Vector Regression*)
UA	Unidades de Ação
UX	Experiência de Usuário (do inglês, *User Experience*)
WLD	Descritor Local de Weber (do inglês, *Weber Local Descriptor*)

SUMÁRIO

1 INTRODUÇÃO

A ideia da construção de máquinas inteligentes acompanha o ser humano há diversos anos e, com o advento do computador na década de 1940, surgiram os meios necessários para atingir este objetivo de maneira eficiente. Coloquialmente, o termo “Inteligência Artificial” (IA) é aplicado a uma máquina que apresenta capacidades que imitam as funções cognitivas que os seres humanos associam à “aprendizado” ou “resolução de problemas”, o que exige a capacidade humana de raciocinar (NASCIMENTO JR; YONEYAMA, 2004).

O ato de aprender é um fenômeno complexo e, como a inteligência, abrange uma ampla gama de processos que são difíceis de definir por si mesmos. Pode ser definido por frases simples, como "ganhar conhecimento" ou "entender", mas algumas etapas são necessárias para fazê-lo, tais como adquirir um novo conhecimento declarativo, desenvolver as habilidades cognitivas por meio da instrução ou prática, de forma a moldar a nova informação de forma genérica e ampla, descobrindo novos conceitos por meio da observação e experimentação.

A essência do Aprendizado de Máquina (do inglês, *Machine Learning,* AM) é fazer com que os computadores possam identificar dependências funcionais dentro dos dados observados, de forma a fazer inferências úteis, utilizando o conhecimento que foi adquirido por meio destes. Em outras palavras, o AM permite construir modelos preditivos a partir de dados observados e aplicar esses modelos para gerar previsões baseadas em novas informações não conhecidas (ZOPPIS et al., 2018).

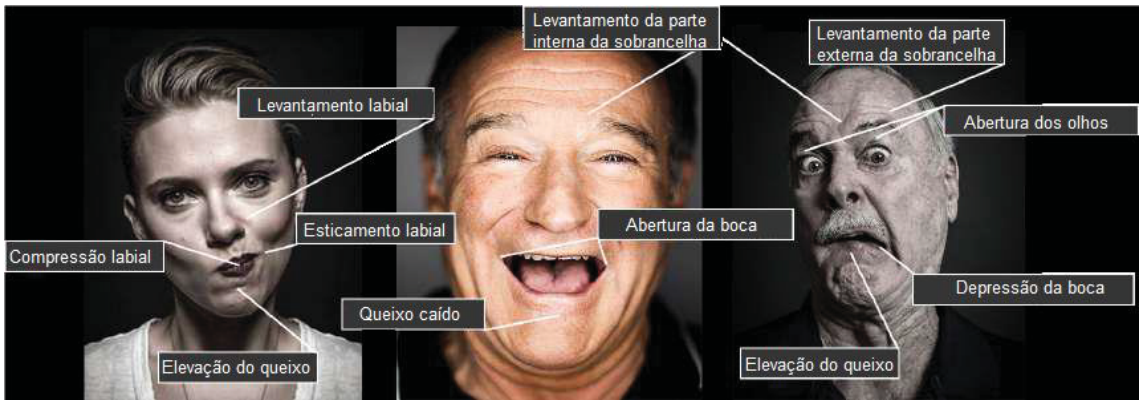
Uma das principais facetas da inteligência humana é a capacidade de analisar e compreender informações sensoriais. O olho humano é um complexo órgão responsável por capturar e focalizar a luz, transformando-a em sinais elétricos que são transmitidos ao cérebro e finalmente traduzidos em imagens (LAMB, 2011). A capacidade visual humana é uma importante faculdade sensorial que fornece respostas rápidas e detalhadas sobre o ambiente, propiciando subsídios para a tomada de decisão em um prazo curto de tempo.

O estudo da visão computacional aborda o desenvolvimento de sistemas que permitam que os computadores processem informações visuais de forma “similar” a um ser humano, ou seja, compreendam conceitos de alto nível a partir

de imagens ou vídeos digitais, da mesma forma que o sistema visual humano pode fazer. Suas tarefas incluem métodos para aquisição, processamento e análise de imagens digitais para fazer a extração de dados e interpretá-los.

Na comunicação humana, ser capaz de interpretar sinais do tipo áudio visuais é parte fundamental do processo. Segundo Mehrabian (1968), a comunicação humana é composta 7% de forma verbal, 38% vocal e 55% por meio de expressões faciais. Quando de um ponto de vista computacional, a máquina ainda fica aquém em tal tipo de análise, porém, com o atual avanço da tecnologia, a melhoria das técnicas de interação homem-máquina se tornou uma demanda crescente. Uma das formas de tornar esse processo de comunicação mais simples e transparente para o usuário se dá por meio da análise e compreensão de gestos, comportamentos e emoções (PORIA et al., 2017).

FIGURA 1.1 – Exemplos de expressões faciais e suas características marcantes.



FONTE: Adaptado de (BREUER; KIMMEL, 2017)

As áreas de pesquisa de Interação Humano-Computador e Computação Afetiva têm explorado as expressões faciais como um recurso para o aprimoramento da comunicação entre o homem e a máquina, além de compreender o estado emocional do usuário, as expressões faciais contêm informações relevantes sobre comportamento humano e desempenham um papel crucial na comunicação interpessoal (PERVEEN et al., 2012). As expressões faciais são a forma de comunicação não verbal mais eficiente para a transmissão de emoções humanas e trazem informações importantes sobre o estado mental, emocional e físico dos indivíduos envolvidos. Estas informações, entretanto, estão codificadas sob diversas características faciais complexas, tais como as da FIGURA 1.1 (ZHANG et al., 2016).

1.1 IDENTIFICAÇÃO DO PROBLEMA

A tarefa de reconhecimento de expressões faciais pode ser uma tarefa trivial para um ser humano, devido às diversas analogias e conexões criadas pelo cérebro. No entanto, para uma máquina, algo que fuja de expressões simples e distintas (como os sentimentos de alegria e raiva) torna-se uma tarefa complexa.

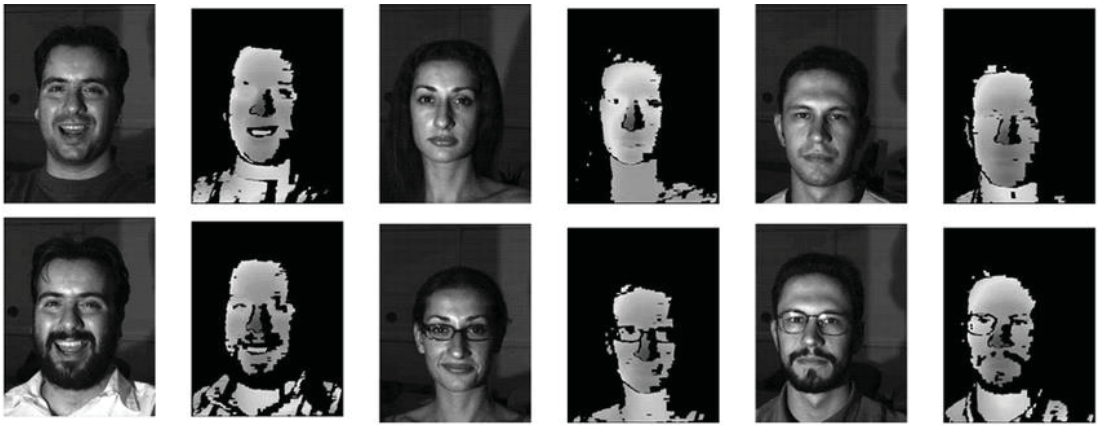
Empregar a face para reconhecer outra pessoa é uma das formas mais naturais adotadas na identificação de pessoas. Entretanto, mesmo com diversas pesquisas realizadas nas últimas duas décadas (COHEN et al., 2003; SHAN et al., 2009; COSSETIN, 2015; CHANTHAPHAN et al., 2016; TARNOWSKI et al., 2017), relacionadas à área de reconhecimento expressões faciais, não existem muitas aplicações em larga escala que se utilizem exclusivamente deste traço biométrico. Esta tarefa ainda é um desafio para os computadores, visto que é complicado identificar e separar as características de cada expressão. A face de um indivíduo em duas expressões diferentes pode ser similar, enquanto características faciais de dois indivíduos com a mesma expressão podem ser distantes uma da outra. Além disso, um sistema deste tipo deve ser robusto de forma a poder avaliar a emoção em rostos com características faciais totalmente distintas, bem como ser capaz de lidar com imagens capturadas em um ambiente descontrolado, que podem sofrer com variações de iluminação, ruído e ângulo da face, por exemplo. A FIGURA 1.2 apresenta três indivíduos com uma expressão feliz. Como pode ser visto, as características faciais variam uma em relação à outra, não apenas na maneira como os indivíduos demonstram a expressão, mas também na iluminação, contraste, posição e fundo (CARDIA NETO, 2014; COSSETIN, 2015; WEI et al., 2016; ZHANG et al., 2016).

FIGURA 1.2 – Três indivíduos distintos apresentando expressão de felicidade.

FONTE: Adaptado de (LOPES, 2016)

Uma forma de aumentar a taxa de acerto de algoritmos de reconhecimento facial em cenários reais descontrolados é por meio do uso de informações da face retratadas em 3D (FIGURA 1.3), por meio da utilização de dados que representem a profundidade da imagem capturada (CARDIA NETO, 2014). Atualmente, a vasta maioria dos sistemas que dispõe da análise e reconhecimento de expressões faciais apoia-se basicamente em bases de dados de imagens estáticas ou de vídeos em duas dimensões (LYONS, 1999; KANADE et al., 2000; HAPPY et al., 2017; THE TFEID PROJECT, 2018). Mesmo que alguns desses sistemas tenham sucesso na realização de suas tarefas, a degradação do desempenho existe quando são apresentados dados de expressões que sofram com rotação da cabeça, movimento, mesmo que sutil, da pele e diferente iluminação (YIN et al., 2006).

FIGURA 1.3 – Exemplos de imagens em cor e seus respectivos mapas de profundidade retratando variações de expressões faciais, penteado, maquiagem, barba e uso de óculos.



FONTE: Adaptado de (TSALAKANIDOU et al., 2005).

Pelo fato dos rostos humanos serem objetos inerentemente tridimensionais, o processo de representação de faces em imagens duas dimensões gera deficiências e perda de informações de características geométricas importantes (MAO, 2015). As imagens em 3D são invariantes a iluminação e apresentam uma robustez maior em relação ao posicionamento da face (LI et al., 2013). É possível, por exemplo, por meio dos dados em 3D, rotacionar ou projetar iluminação em um modelo tridimensional de forma a ajustar e eliminar as mudanças geradas pelo ambiente, o que é, em muitos casos, inviável apenas com o uso de dados em duas dimensões (CARDIA NETO, 2014; ZOHRA; GAVRILOVA, 2017a).

Outro desafio no reconhecimento automático de expressões faciais se mostra na execução desta tarefa em tempo real. As abordagens de reconhecimento baseadas em câmeras RGB (do inglês, *Red, Green, Blue*) são em geral custosas em termos computacionais (LU et al., 2017). Embora a identificação em tempo real possa ser realizada por meio da realização de cálculos complexos na fase inicial de pré-processamento, em vez da fase de reconhecimento em tempo real, o problema fundamental ainda existe (MA et al., 2013). O uso de um conjunto de dados de profundidade pode auxiliar na redução a complexidade computacional necessária para execução da tarefa.

1.2 MOTIVAÇÃO

A análise automática de expressões faciais em tempo real pode facilitar a interação humano-máquina, tornando a troca de informações entre estas partes mais natural e eficiente. Um sistema amplamente acessível com capacidade de identificar expressões faciais pode vir a ser aplicado como uma ferramenta para a pesquisa em ciência comportamental, melhor experiência em jogos (PEREIRA et al., 2019), detecção de sonolência em motoristas (KANG, 2013; JIE et al., 2018) e detecção de dor em pacientes (HOSSAIN, 2016; RODRIGUEZ et al., 2017). Na área da saúde, por exemplo, com a análise feita de forma adequada, é possível descobrir irregularidades na saúde de uma pessoa, tanto físicas como sentimentais. Além disso, existe a possibilidade da aplicação de um sistema deste tipo para adaptar a interface de um sistema para as necessidades do usuário quando o sistema detectar, por exemplo, uma expressão de dificuldade de um usuário idoso, permitindo o aumento da fonte de forma automática (CHANTHAPHAN et al., 2016).

O reconhecimento de expressões faciais tem sido uma área de pesquisa ativa por, pelo menos, os últimos dez anos (ZENG et al., 2009; AIRES et al., 2014; LIU et al., 2014; COSSETIN, 2015; LOPES, 2016; ZHAO et al., 2016; SAVRAN; SANKUR, 2017; HAAMER et al., 2018). As áreas de estudo em que a visão computacional é utilizada vêm se desenvolvendo acentuadamente devido ao aumento do poder computacional das máquinas, em paralelo às melhorias nas técnicas de processamento e aquisição de imagens. Porém, o

reconhecimento de uma expressão facial ainda não é um problema fácil para métodos de aprendizado de máquina.

Os dados faciais 3D podem fornecer uma alternativa promissora de compreender as características do rosto humano no domínio tridimensional e têm potencial para melhorar o desempenho do sistema de reconhecimento. Cada vez mais pesquisadores se concentram no reconhecimento de rosto 3D, principalmente devido aos avanços nas tecnologias de escaneamento tridimensional (LI et al., 2013; AIRES et al., 2014; CARDIA NETO, 2014; ZHANG et al., 2016; SILVA, 2017).

Parte das pesquisas recentes que fazem uso de imagens 3D para reconhecimento facial faz uso de sensores relativamente lentos (PHILLIPS et al., 2005; ZOHRA; GAVRILOVA, 2017b). Dispositivos como o *scanner* de cabeça à laser Cyberware (CYBERWARE INC., 2017) e as câmeras de luz estruturada Minolta (MINOLTA CO. LTD., 2017), têm capacidades industriais e permitem a aquisição de imagens com alta resolução e alto nível de precisão. Porém, os dispositivos mencionados possuem grande porte, elevado custo e necessitam de um passo de montagem antes de sua utilização. Além disso, a captura de um quadro de imagem leva vários segundos. Tais características dificultam a aplicação deste tipo de técnica em cenários de aplicações em tempo real (ZOHRA; GAVRILOVA, 2017a).

FIGURA 1.4 – Vista externa do sensor Microsoft Kinect® v2.

FONTE: https://news.microsoft.com/presskits/xbox/

O lançamento do sensor Microsoft Kinect® (MICROSOFT CORPORATION, 2017) em novembro de 2010 enriqueceu as opções de câmeras de profundidade disponíveis, se diferenciando pelo baixo preço, tamanho compacto e a capacidade de capturar dados de profundidade e imagens RGB à altas taxas em comparação aos demais dispositivos. O Kinect

permite realizar o rastreamento de uma pessoa por meio de diversos pontos corporais, como a movimentação de seu corpo, braços, pernas, mãos e cabeça, incluindo o mapeamento e a identificação de pontos na face. Em 2013, a Microsoft lançou o Kinect v2 (FIGURA 1.4), que introduziu, principalmente, o aumento da precisão da medição de profundidade das imagens capturadas. Como resultado, os dados capturados pela nova versão apresentam resolução superior e menor nível de ruído. Possui, também, um campo de visão de 60% maior que a versão anterior e melhorias nos Kits de desenvolvimento de *software* (do inglês, *Software Development Kit*, SDK) disponíveis, possibilitando a detecção de expressão facial, posição e a orientação de 25 pontos corporais, o peso estimado aplicado sobre cada membro e a velocidade dos movimentos da pessoa capturada. (O'BRIEN, 2013; YANG et al., 2015).

Devido ao seu baixo custo, tamanho reduzido e alta velocidade de aquisição, o sensor Kinect é uma ótima opção para aquisição de imagens RGB-D para aplicações de pesquisa (YANG et al., 2015), permitindo a modelagem em tempo real de forma não invasiva de um rosto humano. Entretanto, mesmo com as melhorias do Kinect v2, quando em comparação a sensores industriais, este apresenta uma precisão da varredura relativamente baixa, como apresentado na TABELA 1.1.

TABELA 1.1 – Comparação entre parâmetros de diferentes scanners 3D.

Sensor	**Velocidade de captura (segundos)**	**Tempo de Carga (segundos)**	**Tamanho (cm^3)**	**Preço (USD)**	**Precisão (mm)**
3dMD	0,002	10	N/A	>$50k	< 0,2
Minolta	2,5	N/A	23000	>$50k	~0,1
Artec Eva	0,063	N/A	2635	>$20k	~0,5
3D3 HDI R1	1,3	N/A	N/A	>$10k	~0,3
SwissRanger	0,02	N/A	283	>$5k	~10
DAVID SLS	2,4	N/A	N/A	>$2k	~0,5
Kinect	0,033	N/A	680	<$200	~1,5 - 50

FONTE: Adaptado de (LI et al., 2013).

Além disso, seu sensor de profundidade gera ruído nos dados, quando em comparação com outros sensores 3D disponíveis (FIGURA 1.5) e, em certos casos, é necessário aplicar um tratamento para suavizar as imagens e corrigir falhas de aquisição (LI et al., 2013; CARDIA NETO, 2014). Por estes motivos, os

dados produzidos por este, necessitam de algoritmos mais robustos e complexos para superar esta limitação. (YANG et al., 2015).

FIGURA 1.5 – Comparação entre os dados de profundidade adquiridos pelos sensores Minolta e Kinect.

FONTE: Adaptado de (LI et al., 2013).

A junção dos pontos fortes de ambos os tipos de sensores se mostra com potencial para o desenvolvimento de sistemas de reconhecimento facial em tempo real. Com a acentuada quantidade de dados faciais de alta qualidade disponíveis para a comunidade científica atualmente, por meio de estudos e bases de dados produzidas com diversos *scanners* industriais, é possível gerar um modelo complexo e robusto, capaz de ser adaptado para aplicação sobre os dados de baixa qualidade obtidos em tempo real com o sensor Kinect.

1.3 OBJETIVOS

Nesta seção são apresentados o objetivo geral e os objetivos específicos.

1.3.1 Objetivo Geral

O objetivo desta dissertação é desenvolver um sistema de identificação de emoções por meio do reconhecimento de expressões faciais para identificação de seis expressões básicas (raiva, desgosto, medo, felicidade, tristeza e surpresa), a partir de um modelo de representação tridimensional, obtido com um sensor Kinect, baseado em diferentes técnicas de classificação de dados, treinadas sobre bases de sensores tridimensionais industriais, fundamentadas em abordagens de Aprendizado de Máquina, Visão Computacional e Processamento de Imagens.

1.3.2 Objetivos Específicos

Os objetivos específicos desta proposta são os seguintes:

- Extrair informações representativas de expressões faciais por meio de características geométricas e de textura, obtidas a partir dos dados da base de imagens faciais RGB-D Bosphorus DB;
- Treinar modelos de AM, tais como Máquinas de vetores de suporte (do inglês *Support Vector Machines*, SVM), K-Vizinhos Mais Próximos (do inglês, *K-Nearest Neighbours*, KNN), Redes Neurais Convolucionais (do inglês *Convolutional Neural Networks*, CNNs) e Comitês de Máquinas para classificação de faces utilizando os vetores de características faciais extraídas;
- Comparar o desempenho dos algoritmos de AM sobre a base de dados na tarefa de identificação de expressões faciais, por meio da matriz de confusão e do tempo de processamento;
- Desenvolver funções básicas de comunicação entre o sensor Kinect e o ambiente computacional MATLAB, da MathWorks;
- Aplicar o melhor modelo obtido para desenvolvimento de um sistema capaz de extrair características e classificar as emoções apresentadas pelas expressões faciais extraídas por meio do fluxo de vídeo gerado pelo sensor Kinect.

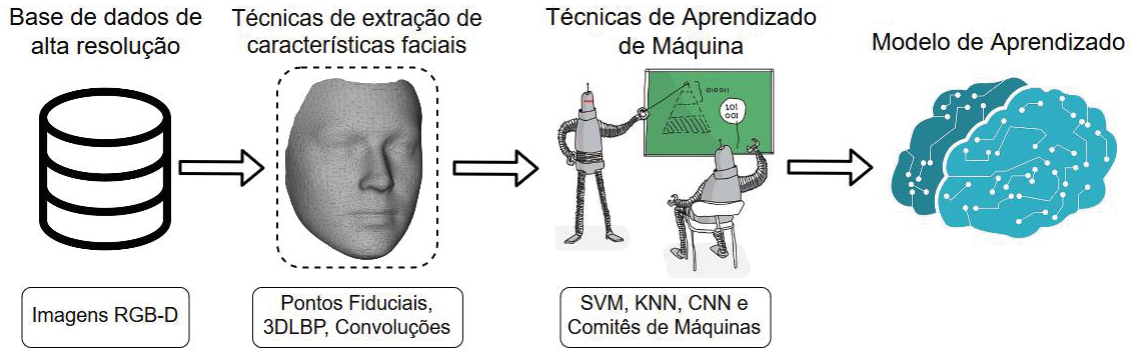
1.4 CONTRIBUIÇÕES

Esta dissertação apresenta uma pesquisa sobre análise e classificação de expressões faciais em imagens tridimensionais num ambiente computacional. Nela são abordados os passos de aquisição de uma imagem digital tridimensional, a extração de características faciais e classificação da expressão facial detectada em uma de seis emoções básicas (raiva, desgosto, medo, felicidade, tristeza e surpresa).

A metodologia proposta consiste em dois processos de momentos distintos: o treinamento de um modelo de aprendizado para classificação de expressões faciais e a aplicação deste modelo em novos dados obtidos por meio de uma câmera RGB-D. Como apresentado na FIGURA 1.6, a metodologia para geração do modelo de aprendizado é baseada em uma base de dados faciais

tridimensionais de alta resolução, sobre os quais as técnicas de processamento de imagens e de AM são aplicadas de forma a isolar características que permitam separar as diferentes expressões faciais entre si. Neste passo, são comparadas algumas abordagens distintas tanto de extração de características quanto de AM, permitindo a busca de um modelo mais apropriado para a tarefa.

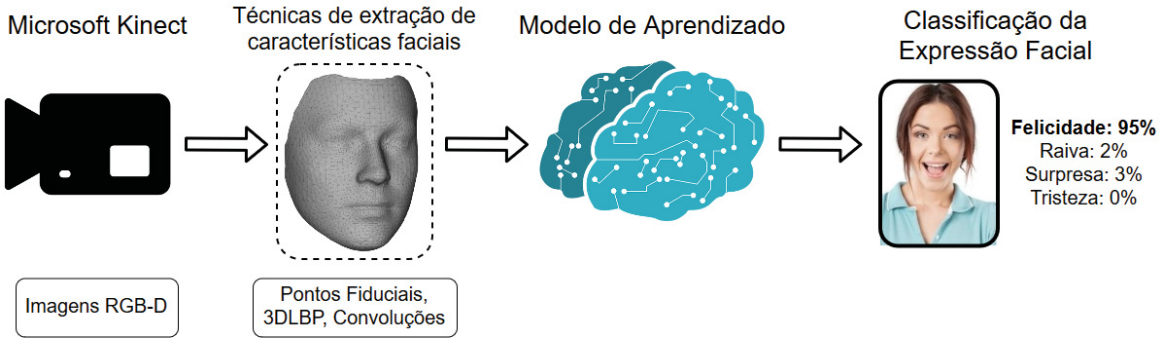
FIGURA 1.6 – Metodologia proposta para treinamento do modelo de aprendizado.



FONTE: O autor (2018).

Em um segundo momento, o modelo de aprendizado gerado é aplicado sobre dados de menor resolução e maior ruído obtidos pelo sensor Microsoft Kinect, como mostrado na FIGURA 1.7. A partir destas imagens, espera-se que o modelo gerado seja capaz de identificar corretamente a expressão facial apresentada.

FIGURA 1.7 – Metodologia proposta para classificação das expressões faciais.



FONTE: O autor (2018).

O método proposto se difere da abordagem tradicional empregada na literatura pelo fato dos dados apresentados para os algoritmos de AM não serem provenientes de uma base de dados gerada por dados do próprio Kinect, mas sim por transferência de aprendizado de uma base de dados gerada por sensores industriais de alta resolução, adaptada para aplicação com os dados

gerados pelo Kinect. O uso de tal técnica permite com que seja possível gerar modelos de classificação de características faciais robustos, visto que os sensores industriais são capazes de capturar uma extensa gama de detalhes que a baixa resolução do Kinect não é capaz de armazenar, aumentando a possível precisão de reconhecimento. Ao mesmo tempo, a aplicação do modelo previamente gerado à alta taxa de captura do Kinect permite que, pelo uso de classificadores com baixo custo computacional, o sistema seja aplicado para classificação em tempo real.

1.5 ESTRUTURA DO TRABALHO

Neste capítulo foram apresentados os desafios e a motivação para este trabalho, bem como os objetivos gerais e específicos, e as contribuições desta dissertação.

O restante está organizado da seguinte forma. No capítulo 2 é apresentado um apanhado da literatura sobre estudos recentes relacionados ao reconhecimento de expressões faciais, sendo detalhada a abordagem implementada por cada um, assim como os resultados obtidos.

As técnicas de AM empregadas para classificação dos dados e identificação da expressão correspondente são relatadas no capítulo 3. O capítulo 4 aborda as técnicas utilizadas para análise, identificação e representação de expressões faciais em imagens digitais.

O capítulo 5 detalha o método proposto para reconhecer expressões a partir dos dados tridimensionais, bem como os procedimentos seguidos para a realização dos experimentos. Os resultados obtidos e a discussão frente aos diversos trabalhos encontrados na literatura são apresentados no capítulo 6.

Por fim, no capítulo 7 são apresentadas as conclusões parciais obtidas por esta proposta de dissertação e o cronograma para a escrita e defesa de dissertação.

2 APANHADO DA LITERATURA

Diversos setores da indústria vêm obtendo resultados relevantes e bem-sucedidos na aplicação de sistemas de visão computacional na realização de diversas tarefas. Os sistemas de classificação ou identificação de objetos em imagens digitais são pesquisados em diversas áreas do conhecimento e estão cada vez mais presentes no ramo industrial (ROCHA, 2015). As ferramentas computacionais evoluíram no que tange a sua capacidade de processamento e tornaram-se acessíveis em termos de custo, favorecendo o desenvolvimento e aplicação desse tipo de sistema.

2.1 RECONHECIMENTO DE EXPRESSÕES FACIAIS

O reconhecimento de expressões faciais é um campo de pesquisa da visão computacional em ativo crescimento nos últimos dez anos, visto que, em comparação a outros canais de expressão de emoções, tais como as ações corporais e a fala, as expressões faciais apresentam maior força expressiva e facilidade de identificação, produzindo maior área de aplicação (MAO, 2015).

As pesquisas apresentadas recentemente na literatura voltam-se principalmente para o reconhecimento de emoções complexas e espontâneas (VINCIARELLI et al., 2012; YAN et al., 2016; HAPPY et al., 2017; YANG et al., 2018). Estas exigem o uso de algoritmos de detecção de características e classificação sofisticados com a presença de restrições temporais.

A maior parte das pesquisas relacionadas (detalhes podem ser obtidos em KUMARI et al. (2015) e (KUMAR; SHARMA, 2018) para duas revisões importantes) à análise de expressões faciais foram inspiradas a partir do trabalho de Ekman (1982), no qual foi proposto o sistema de codificação de ação facial (*Facial Action Codification System*, FACS), a primeira técnica abrangente para a identificação de todos os movimentos básicos dos músculos faciais visualmente distintos e observáveis, denominando-os Unidades de Ação (do inglês *Action Units* ou UAs). Cada UA possui alguma base muscular relacionada e uma determinada expressão facial pode ser descrita por uma combinação de UAs.

Mase (1991) propôs uma das primeiras a usar técnicas de processamento de imagem para reconhecer as expressões faciais. Oliver et al. (2000) usaram o rastreamento da face para extrair características de formato da

boca e usou-as para reconhecimento de expressão facial baseado em HMM (capaz de reconhecer as expressões neutro, feliz, triste e uma boca aberta).

Mais recentemente, Zhang et al. (2012) empregaram características de representação facial extraídas por meio de padrões binários locais (do inglês, *Local Binary* Pattern, LBP). Para resolver o problema de alta dimensionalidade dos dados gerados pelos métodos baseados em textura, o algoritmo LFDA (*Local Fisher Discriminant Analysis*) foi usado para produzir representações de baixa dimensionalidade dos dados obtidos pelos LBP. Estes dados alimentam uma SVM treinada para classificação de expressões faciais.

Cossetin (2015) abordou o problema de forma similar, por meio do uso de métodos extratores de características baseados em textura da imagem, LBP e WLD (*Weber Local Descriptor*). Este propôs uma técnica de redução de dimensionalidade que seleciona atributos utilizando exemplos compostos por duas expressões faciais, alimentando um conjunto de 21 classificadores do tipo SVM para diferenciação *um-contra-um* entre sete expressões faciais básicas. O resultado da classificação é atribuído à expressão facial com mais votos no conjunto de classificadores binários especializados.

A popularização de técnicas de aprendizado profundo também se proliferou para a área de reconhecimento de expressões faciais, com a presença de diferentes implementações de CNNs. Dentre estes trabalhos, Mayya et al. (2016) apresentaram um modelo de classificação por meio de uma rede neural convolucional profunda, que é responsável por realizar tanto a extração quanto o processamento de características, usando a arquitetura Caffe (*Convolutional Architecture for Fast Feature Embedding*). O método de Mayya requer tempo significativamente menor para processamento dos dados em comparação com técnicas mais antigas de AM.

Mesmo em 2019, a comunidade científica continua ativa nesta área, sendo que, cada vez mais, tendendo para o uso de técnicas baseadas em princípio de aprendizado profundo. (DENG et al., 2019) aplicaram uma rede adversarial generativa condicional (do inglês, *Conditional Generative Adversarial Network*) de forma a modelar as características mais relevantes de cada expressão em um modelo que minimiza a variação intra-classe. Já Li et al. (2019) trabalharam em uma CNN capaz de identificar áreas com oclusão facial na imagem e focar em outras regiões de interesse que fornecem informações

relevantes para a classificação da emoção apresentada. Neste mesmo ano, Kim (2019) tenta explicar o resultado de classificação de expressões gerado por uma CNN por meio da análise das UAs representadas por cada uma das características extraídas pelas camadas convolucionais da rede.

2.2 RECONHECIMENTO DE EXPRESSÕES FACIAIS EM DADOS TRIDIMENSIONAIS

O uso dos dados 3D para análise facial não é tão amplo quanto o uso de imagens bidimensionais. Tecnologias de imagem 3D trazem soluções para a sensibilidade a mudanças nas condições do ambiente e do objeto, tais como iluminação e pose, fornecendo os meios para medir e reconstruir modelos tridimensionais que sofrem menor degradação devido a condições de ambiente. É interessante notar que a maioria dos trabalhos de reconhecimentos em 3D ainda são dependentes da detecção de pontos de referência em duas dimensões.

Com base no FACS, diversos bancos de dados de imagens de faces em 3D contendo anotações de expressões faciais foram desenvolvidos. Haamer et al. (2018) apresenta uma revisão das bases disponíveis na literatura. Dentre estes, a CASIA 3D (XU et al., 2004), BU-3DFE (YIN et al., 2006), Bosphorus DB (SAVRAN et al., 2008), e KinectFaceDB (MIN et al., 2014), foram gerados usando equipamentos de captura 3D.

Sendo a base BosphorusDB a utilizada neste trabalho, buscou-se por estudos que experimentaram suas técnicas sobre os dados desta, permitindo uma avalição comparativa ao final desta dissertação. Dentre estes, (SANDBACH et al., 2012) realizam um comparativo entre o desempenho de vetores de características extraídos por meio de uma extensão da técnica LBP para três dimensões, intitulada 3DLBP e a técnica de Padrões Locais Normais Binários (do inglês, *Local Normal Binary Pattern,* LNBP), aplicadas a múltiplas SVMs, na identificação de 25 unidades de ação facial. Também para identificação de UAs (SAVRAN; SANKUR, 2017) fazem uso somente da geometria facial 3D como representação da face.

TABELA 2.1 – Visão geral dos estudos de reconhecimento de emoções utilizando a base BosphorusDB.

Artigo	Dados	Representação de características faciais	Técnica IA	Extração automática?	Taxa de Acerto
(VRETOS et al., 2011)	3D	Momentos *Zernike* em imagens de profundidade	SVM	Sim	60,5%
(SANDBACH et al., 2012)	3D	LBP + LNBP	SVM	Sim	96,35% - 25 Unidades de ação
(ZHAO et al., 2013a)	2D+3D	Textura 2D (LBP) + Pontos Fiduciais Faciais	*Bayesian Belief Network*	Não	85,6%
(MOHAMMADI et al., 2014)	2D	Dicionário esparso de componentes principais da diferença de imagens	*Sparse Representation-based Classification*	Sim	72,41% - 6 classes
(MANISHA, DR JAGJIT SINGH, 2015)	3D	Características Geométricas (*Directional Rank Histogram Pattern* e *Local Directional Strength Pattern*)	*Kernel* PCA/GDA + CNN	Sim	96,25%
(VIERIU et al., 2015)	3D	LBP	*Random Forest*	Sim	66,50%
(ZHANG et al., 2015a)	2D+3D	Pontos Fiduciais Faciais	RNA/SVR/Comitês de Máquinas	Sim	92,2% - 6 classes
(AHMED. et al., 2016)	2D	Pontos Fiduciais Faciais	SVM	Sim	78,37% - 7 classes 89,53% - 5 classes
(DING, 2016)	2D+3D	2D: Características de textura (*Histogram of Second Order Gradients*) 3D: Características geométricas (*Histogram of mesh Gradients* e *Histogram of mesh Shape index*)	Seleção de atributos *Adaboost* + SVM	Sim	84,72%
(ZHAO et al., 2016)	2D+3D	Regiões de textura em torno de pontos fiduciais (*Deformable Partial Facial Model*)	SVM	Não	90,0%
(SAVRAN; SANKUR, 2017)	3D	Geometria facial (*Triangular wireframe mesh*)	*Naive Bayes*/SVM + *Adaboost*	Sim	96,8% - 25 Unidades de ação
(DERKACH; SUKNO, 2018a)	3D	Representação espectral de componentes de textura facial (*Graph Laplacian Features*)	SVM	Sim	77,33%

FONTE: O autor (2018).

Além disso, trabalhos como os de Zhao et al. (2013), Ahmed et al. (2016) e Zhao et al. (2016) empregaram diferentes técnicas de IA aplicados sobre abordagens de características geométricas codificadas sob pontos fiduciais faciais. Zhang et al. (2015) utilizou RNAs para identificar e quantificar a intensidade de UAs apresentadas em imagens RGB-D capturadas por meio do Kinect. Os demais trabalhos são listados na TABELA 2.1.

2.3 ESTUDOS UTILIZANDO O MICROSOFT® KINECT

Cardia Neto (2014) utilizou o Kinect para realizar o reconhecimento de faces voltado para a tarefa de identificação biométrica. Este desenvolveu um método que combina os descritores de extração de características faciais 3DLBP e histogramas de gradientes médios orientados (do inglês, *Histogram of Average Oriented Gradients,* HAOG) sobre a base EURECOM, mostrando que a junção de ambas as técnicas garante um melhor desempenho, mesmo com obstrução parcial da face, por meio de preenchimento simétrico. Li et al. (2013) aplicaram as capacidades 3D do Kinect para reconhecer emoções sob diferentes condições. Seu método foi baseado em uma base de imagens faciais RGB-D capturadas pelo sensor Kinect em baixa resolução com diferentes poses, expressões, iluminação e oclusão. Seus resultados demonstraram que o uso de informações RGB-D pode melhorar o desempenho do reconhecimento de emoção facial em comparação com os métodos usando somente informações 2D ou 3D.

Com o lançamento da versão 2.0 do Kinect, no ano de 2014, diversos pesquisadores demonstraram interesse em seu uso. Dentre estes, Yang et al. (2015) avaliaram a precisão do sensor de profundidade do Kinect v2, gerando um modelo de cone de forma a ilustrar sua distribuição de precisão. Estes também analisaram a variância dos valores de profundidade capturados identificando o comportamento ruidoso gerado pelo equipamento, além da distribuição de precisão, resolução dos dados de profundidade capturados, entropia de profundidade, ruído de borda e ruído estrutural.

Tarnowski et al. (2017) estudaram o reconhecimento de sete estados emocionais faciais básicos. Os coeficientes que descrevem os elementos de expressões faciais, extraídos por meio da SDK do Kinect, foram aplicados como características para os algoritmos de AM sobre o modelo facial tridimensional. A

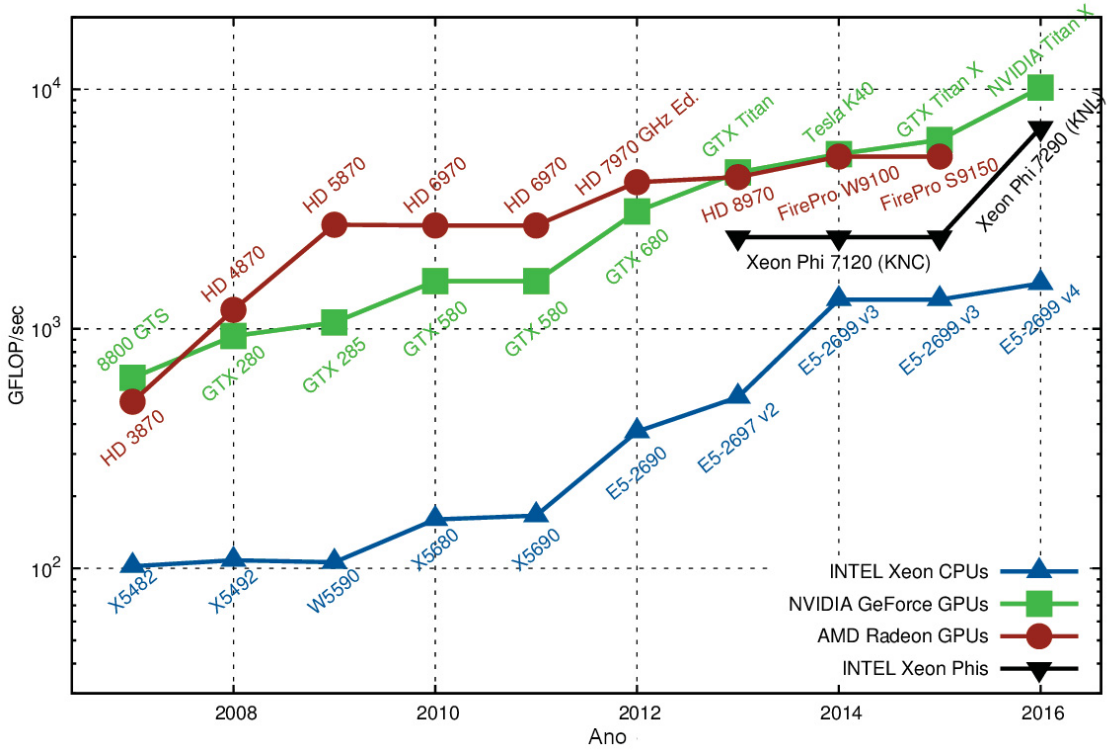
classificação das características foi realizada por meio de duas abordagens distintas, KNN e uma rede neural Perceptron Multicamada (do inglês, *Multi Layer Perceptron*, MLP).

Chanthaphan et al. (2016) analisaram o uso do Kinect v2 em aplicações de tempo real, empregando PFFs fornecidos pela SDK de rastreamento facial do equipamento, aplicados tanto para o treinamento de um algoritmo de KNN quanto para SVM. Amara et al. (2019) verificaram, pelos pontos de geometria facial, que a dimensão adicional de profundidade fornecida pelo Kinect permite aumento na acurácia de classificadores do tipo KNN.

3 APRENDIZADO DE MÁQUINA

Durante as últimas duas décadas, houve um acentuado aumento na quantidade de informações geradas, armazenadas e, principalmente, processadas, dado a popularização da tecnologia e ao aumento do poder computacional ao longo dos anos, conforme apresentado na FIGURA 3.1. Todo esse material pode ser considerado como uma “mina de ouro”, abrindo possibilidades de extração de informações relevantes escondidas em petabytes de dados.

FIGURA 3.1 – Comparativo do poder computacional entre CPUs e GPUs.



FONTE: Adaptado de (RUPP, 2013).

Esta informação minerada mostra-se útil, ao passo que grandes companhias, tais como Google, Facebook e Amazon, começaram a perceber a importância de encontrar padrões nesses dados, para poder prever os comportamentos de seus clientes, as redes sociais podem analisar os gostos e comportamentos das pessoas para exibir propaganda mais adequada, por exemplo. Atualmente, a análise de dados já não pode ser realizada manualmente devido à imensa quantidade de dados. Isso leva ao uso de ferramentas

computacionais capazes de analisar grandes quantidades de dados (*big data*) e aprender a extrair informações automaticamente (SAHLA, 2018).

O AM vem ao encontro dessa necessidade. Esta é uma subárea da Inteligência Artificial (IA) que se concentra no estudo e na construção de sistemas que possuem a capacidade de “aprender” por conta própria, a partir de um conjunto de dados, minimizando a dependência humana, permitindo a realização de inferências de novas informações a partir do que foi aprendido. Segundo Samuel (1959), o estudo de AM surgiu como uma forma de dar aos "computadores a capacidade de aprender sem serem explicitamente programados". Em outras palavras, AM é um processo usado para gerar modelos que são capazes de identificar padrões e extrair informações de dados para resolver um determinado problema e, consequentemente, melhorar automaticamente seu desempenho (STENROOS, 2017).

O AM tem sido uma pesquisa ativa desde a sua criação. O interesse nesta área é motivado pelas diversas aplicações dos "sistemas inteligentes" na vida cotidiana. Atualmente, vários dos *softwares* usados diariamente, tais como mecanismos de busca, redes sociais, assistentes pessoais, editores de texto, entre outros, incorporam, em alguns aspectos, algoritmos de aprendizado de máquina de forma a demonstrar um comportamento inteligente ou capaz de aprender com o uso.

A maior parte dos estudos de AM encontrados na literatura empregam técnicas de aprendizado supervisionado. Este tipo de abordagem requer o uso de um conjunto de dados de treinamento etiquetados. Cada instância deste conjunto deve ter um rótulo atribuído, no formato $(x, y) \in X \times Y$, sendo que x representa um conjunto de dados de entrada e y a predição verdadeira para esta amostra específica. Esses rótulos servem como “exemplos” para “ensinar” o algoritmo de aprendizado a prever o resultado esperado dado uma entrada de dados deste conjunto. Este treinamento por meio de pares de entrada-saída é utilizado para que o algoritmo posso encontrar uma função determinística que mapeie qualquer conjunto de dados de entrada para uma resposta correta, capaz de prever futuras instâncias de dados de entrada não previamente vistos também para uma resposta correta.

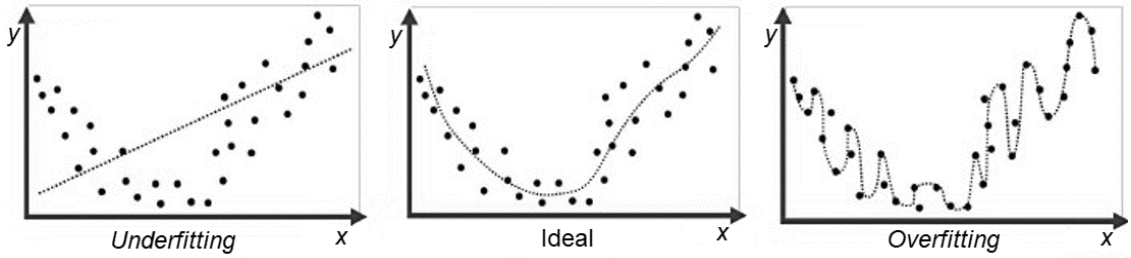
O aprendizado supervisionado faz uso do princípio da indução (inferência lógica). Para a indução derivar novo conhecimento que seja

representativo, os exemplos de cada classe devem possuir uma quantidade suficiente de dados, possibilitando, assim, a obtenção de hipóteses úteis e genéricas para um determinado tipo de problema. (MONARD; BARANAUSKAS, 2003).

A partir de um processo iterativo, os dados de treinamento são repassados ao algoritmo de aprendizado para gerar um modelo capaz de produzir uma previsão de saída. Essa previsão é comparada com o rótulo atribuído aos dados de treinamento para estimar o erro do modelo. Com base neste erro, o algoritmo ajusta os parâmetros do modelo para reduzi-lo, até um limiar de desempenho determinado (geralmente utiliza-se a precisão do modelo quando aplicado nos dados de validação) seja atingido. Em diversas situações, faz-se necessário retornar aos passos anteriores do processo, como o pré-processamento dos dados ou a escolha do algoritmo de aprendizado, quando os resultados obtidos não são satisfatórios, mesmo após muitas iterações de ajuste de parâmetros.

Pelo fato de um conjunto de dados de treinamento ser uma pequena amostra dos dados de entrada, o algoritmo de aprendizado deve ser capaz de generalizar o conhecimento adquirido de forma a lidar com novos de dados não vistos previamente. Um modelo gerado que seja demasiadamente simples falha na captura de aspectos importantes de representação dos dados, situação comumente denominada *underfitting*. Por outro lado, métodos complexos podem se assemelhar muito ao conjunto de dados de treinamento, absorvendo detalhes sem importância e até mesmo ruído, o que também leva a uma generalização ruim, conhecida como *overfitting*. Um modelo com *overfitting* aprende a modelar os exemplos conhecidos, mas não é capaz e entender a relação subjacente a estes (BISHOP, 2006; STENROOS, 2017). A representação desse fenômeno para um problema de duas variáveis é observado na FIGURA 3.2.

FIGURA 3.2 – Diferentes níveis de generalização do modelo.

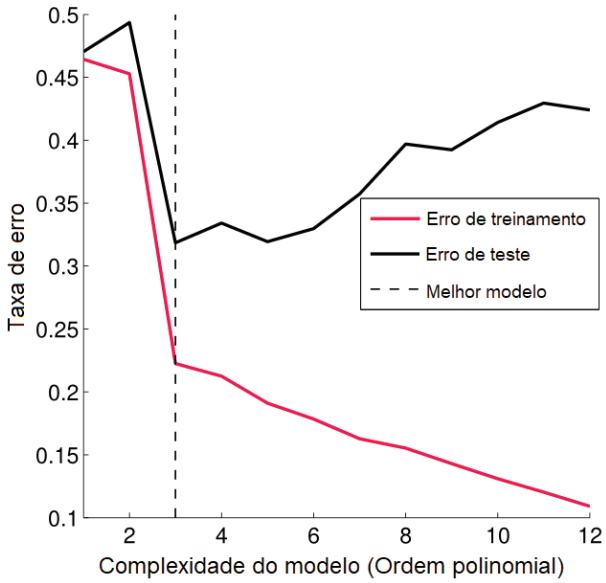


Adaptado de (ZAPLETAL, 2017).

Normalmente, o modelo de aprendizado é treinado com a maior quantidade possível de dados de entrada, de forma a obter o melhor desempenho possível. Ao mesmo tempo, sua taxa de erro deve ser verificada em novos dados distintos e independentes para, assim, verificar se a capacidade de generalização não está sendo prejudicada. Normalmente, estes dois conjuntos distintos são obtidos dividindo-se o conjunto de dados disponível em treinamento e conjunto de testes (existem diversas técnicas para realizar esta divisão, tais como 4:1, *K-fold cross-validation* ou *Leave One Out cross-validation*). O modelo é treinado apenas com dados de treinamento e o desempenho do modelo é testado nos dados de teste. Embora o verdadeiro erro de generalização nunca possa ser realmente obtido, sua aproximação pela taxa de erro de teste é suficiente para a maioria das tarefas de aprendizado de máquina. A relação entre os erros no conjunto de treinamento e teste (apresentados na FIGURA 3.3) ajudam na identificação da complexidade do modelo gerado. Um modelo que apresenta erros cada vez menores no conjunto de treinamento e cada vez maiores no conjunto de teste é um indicativo de *overfitting*, por exemplo.

O desempenho do algoritmo pode ser avaliado a partir da qualidade e quantidade de erros. Uma função de perda, como erro quadrático médio, é usada para atribuir um custo aos erros. O objetivo na fase de treinamento é minimizar essa perda (BISHOP, 2006).

FIGURA 3.3 – Relação entre a complexidade do modelo e os erros de treinamento e teste.



FONTE: Adaptado de (STANSBURY, 2013).

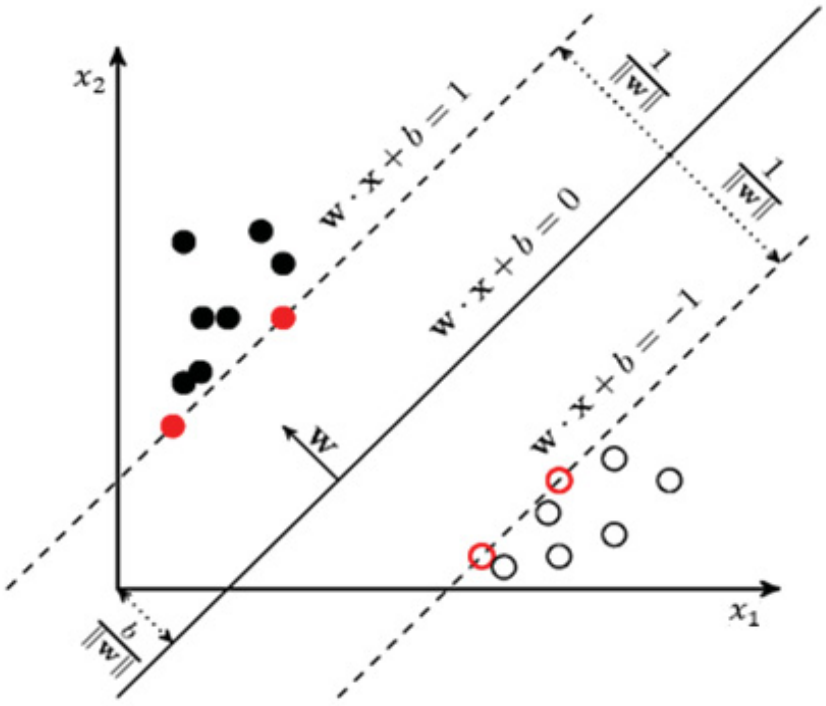
3.1 MÁQUINAS DE VETORES DE SUPORTE

As Máquinas de Vetores de Suporte (do inglês *Support Vector Machines*) constituem uma técnica embasada pela teoria de aprendizado estatístico desenvolvidas por Vapnik em 1995, com o intuito de resolver problemas de classificação de padrões.

A SVM destaca-se por pelo menos duas características: possui sólida fundamentação teórica e é bem-sucedida em aplicações práticas, como na categorização de textos (MA; GUO, 2014), na análise de imagens (BURGES, 2009) e em Bioinformática (LIN et al., 2018). Além disso, essa é uma técnica considerada mais fácil de ser aplicada do que rede neural, por exemplo. (BOUZALMAT et al., 2014).

As SVM funcionam mapeando dados para um espaço altamente dimensional de forma que estes possam ser categorizados. Após um separador entre as categorias ser localizado, os dados são transformados de modo que o separador possa ser desenhado como um hiperplano. Após isso, as características dos novos dados podem ser aplicadas para prever o grupo ao qual um novo registro deve pertencer. A SVM trabalha de tal forma que a distância entre o hiperplano e as classes seja a maior possível (SANTOS, 2002).

FIGURA 3.4 – Hiperplano de classificação de uma SVM.



FONTE: Adaptado de (BOUZALMAT et al., 2014)

A SVM com margens rígidas (FIGURA 3.4) é o modelo mais simples de SVM e somente pode ser aplicada em dados linearmente separáveis. A equação de um hiperplano é apresentada na Equação 3.1, onde w representa o vetor normal ao hiperplano h descrito, que deve ser ajustado, x é um vetor de entrada e b corresponde à distância do hiperplano em relação à origem, com $b \in \Re$ em que $w \cdot x$ é o produto escalar entre os vetores w e x (FACELI et al., 2011),

$$f(x) = w \cdot x + b = 0 \tag{3.1}$$

O hiperplano h está associado a um par de hiperplanos de apoio $h1$ e $h2$ (equações definidas na FIGURA 3.4) que são paralelos ao limite de decisão e passam pelo ponto x_i mais próximo. A distância entre esses planos de suporte é chamada de margem. Este é o parâmetro utilizado como critério de avaliação da otimização da SVM, dado que quando maior a margem, melhor tende a se desempenhar o modelo. A distância entre os dois hiperplanos de suporte é obtida por:

$$w \cdot (x_1 - x_2) = 2 \tag{3.2}$$

$$d = \frac{2}{\|w\|} \tag{3.3}$$

O objetivo do classificador SVM é maximizar o valor de d. Este objetivo é equivalente a minimizar o valor de $\|w^2\|/2$. Os valores de w e b são obtidos resolvendo este problema de otimização quadrática sob as restrições:

$$w \cdot x_i + b \geq 1 \quad se\ y_i = 1 \tag{3.4}$$

$$w \cdot x_i + b \leq -1 \quad se\ y_i = -1 \tag{3.5}$$

onde y_i é a variável de classe para x_i. A imposição dessas restrições fará com que o SVM coloque as instâncias de treinamento com y_i = 1 acima do hiperplano h_1 e as instâncias de treinamento com y_i = -1 abaixo do hiperplano h_2. O problema de otimização pode ser resolvido usando o método multiplicador de Lagrange. A função objetivo a ser minimizada, na forma Lagrangiana, pode ser escrita como:

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i (y_i (w \cdot x_i + b) - 1) \tag{3.6}$$

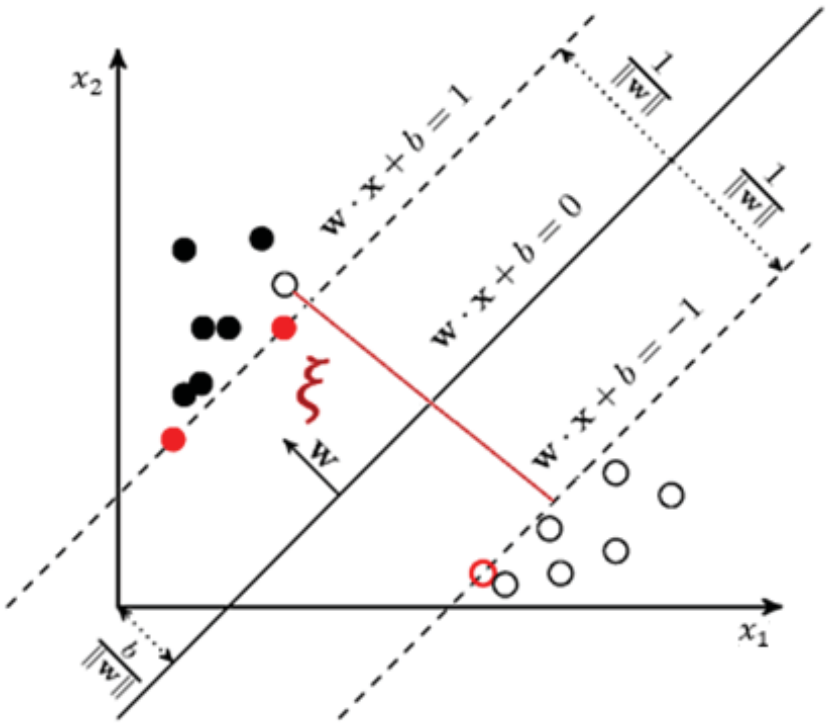
onde α_i são multiplicadores de Lagrange e N representa o número de amostras.

Em alguns casos, é preferível classificar erroneamente algumas das amostras de treinamento (erros de treinamento) para obter um plano de limite de

decisão com margem máxima. Um limite de decisão sem erros de treinamento, mas margem menor, pode levar a um sobre ajuste e não pode classificar as amostras desconhecidas corretamente. Por outro lado, um limite de decisão com poucos erros de treinamento e uma margem maior pode classificar as amostras desconhecidas com mais precisão. Portanto, deve haver uma compensação entre a margem e o número de erros de treinamento.

Para dados não são linearmente separáveis ou que apresentam ruídos é empregada a SVM com margens suaves (FIGURA 3.5), que é uma adaptação da SVM com margens rígidas introduzindo as variáveis de folga ξi. Isso permite que alguns dados possam violar a restrição imposta pelo hiperplano h. Além disso, uma penalidade para o erro de treinamento deve ser introduzida na função objetivo, a fim de equilibrar o valor da margem e o número de erros de treinamento.

FIGURA 3.5 – SVM com margens suaves.

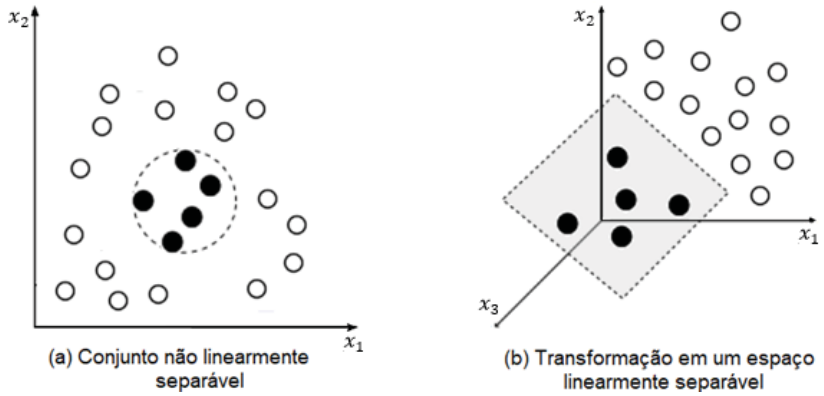


FONTE: Adaptado de (BOUZALMAT et al., 2014)

As SVMs são eficazes na classificação de conjuntos de dados linearmente separáveis. No entanto, existem casos em que não é possível realizar a divisão dos dados utilizando um hiperplano linear (FIGURA 3.6 (a)). Quando isso ocorre é mapeado o conjunto de treinamento de seu espaço

original, referenciado como entrada, para um novo espaço de maior dimensão, denominado de espaço de características, facilitando a separação dos dados por meio de uma SVM linear (FIGURA 3.6 (b)) (FACELI et al., 2011).

FIGURA 3.6 – Exemplo de conjuntos não linearmente separáveis.



FONTE: Adaptado de (COSSETIN, 2015).

O mapeamento para um novo espaço de características não linear e de alta dimensão é feito com o uso de uma função de *kernel*. O desempenho do classificador SVM é dependente da escolha de uma função de *kernel* apropriada. As funções mais comuns são apresentadas na TABELA 3.1, em que x_i e x_j são os vetores de dados para dois padrões.

TABELA 3.1 – Funções de *kernel* mais comuns.

Kernel	**Função**	**Parâmetros**
Polinomial	$(\delta(x_i \cdot x_j) + k)^2$	δ: escala k: deslocamento d: grau do polinômio
Radial Basis Function (RBF)	$e^{-\sigma\lvert x_i - x_j\rvert^2}$	σ: largura do raio
Sigmoidal	$\tanh(\delta(x_i \cdot x_j) + k)$	δ: escala k: deslocamento

FONTE: Adaptado de (COSSETIN, 2015).

3.2 K-VIZINHOS MAIS PRÓXIMOS

O algoritmo K-Vizinhos Mais Próximos (do inglês, *K-Nearest Neighbors,* KNN) tradicional faz uso das classes de *k* amostras de treinamento localizadas mais próximas (vizinhos mais próximos) da instância de teste, de acordo com uma métrica de distância definida, para classificá-la. O KNN pode ser enquadrado como uma técnica de classificação capaz de alcançar alta precisão de classificação em problemas que têm distribuição não-normal (BOARETTO, 2017).

Além disso, o KNN é uma metodologia baseada em instância, de aprendizado lento não paramétrico, o que significa que não são feitas suposições sobre a distribuição subjacente dos dados. Isso é muito útil, pois no mundo real, a maioria dos dados práticos não obedece às suposições teóricas típicas (tais como distribuições gaussianas ou linearmente separáveis). Também é um algoritmo preguiçoso, pois não faz uso os pontos de dados de treinamento para fazer uma generalização. Em outras palavras, não há fase de treinamento explícito. A falta de generalização significa que a KNN mantém todos os dados de treinamento. Mais especificamente, todos os dados de treinamento são necessários durante a fase de execução. A maioria dos algoritmos preguiçosos (especialmente o KNN) toma decisões com base em todo o conjunto de dados de treinamento, ou pelo menos, um subconjunto deles (THIRUMURUGANATHAN, 2010).

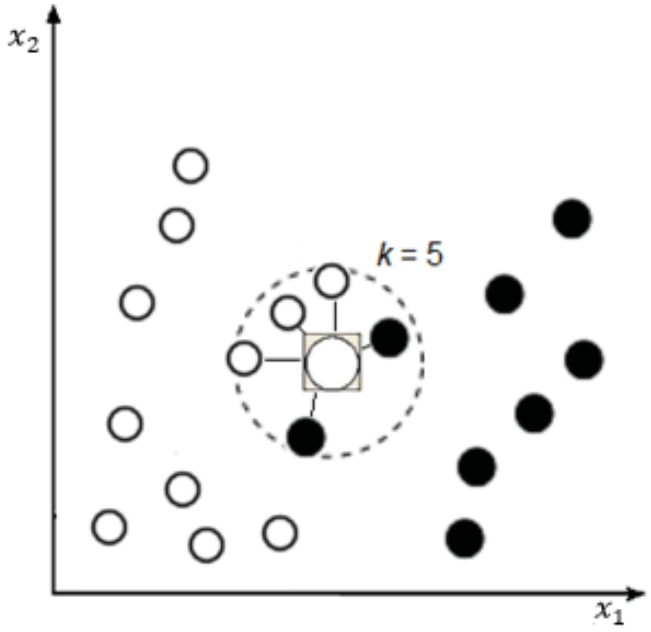
O KNN apresenta como suas principais vantagens a simples implementação e a grande velocidade de treinamento e execução. Em contrapartida, sua precisão é muito dependente da quantidade e da qualidade do conjunto de dados de treinamento e seu parâmetro *k* deve ser muito bem ajustado para bom desempenho do algoritmo.

O aprendizado do KNN consiste em armazenar todas as instâncias de treinamento com seus rótulos de classe. A classificação de uma nova instância desconhecida ocorre por meio do cálculo da distância, dada a métrica escolhida, da nova instância de dados para as demais instâncias do conjunto de treinamento, de forma a selecionar as *k* instâncias de treinamento (vizinhos) com menor distância, como mostrado na FIGURA 3.7. A partir da classe destes vizinhos, a probabilidade de cada classe para nova instância é ponderada seguindo:

$$P(y = j|\mathrm{X} = x) = \frac{1}{k}\sum_{i\epsilon\mathcal{A}} I\left(y^{(i)} = j\right) \tag{3.7}$$

onde, x representa a entrada, y a saída, $\mathcal{A}$ é o conjunto dos vizinhos mais próximos, I é a função que avalia quando a instância analisada pertence a tal classe j. No exemplo da FIGURA 3.7, o novo dado é classificado como pertencente à classe “branca”, visto que a maior parte dos dados pertencentes ao grupo de *k* vizinhos mais próximos é desta classe.

FIGURA 3.7 – Visualização da classificação de um novo dado pelo algoritmo KNN.



FONTE: Adaptado de (BOARETTO, 2017).

O ajuste do parâmetro *k* permite com que mais ou menos amostras de dados de treinamento sejam avaliados para que a decisão seja tomada. Comumente é recomendado escolher valores ímpares para *k*, de forma que não seja possível “empate” no momento de classificação do novo dado.

A definição de quais instâncias do conjunto de treinamento pertence ao conjunto dos *k* vizinhos mais próximos pode ser realizada adotando diferentes funções de distância em implementações de KNN, tais como distância Euclidiana, Hamming, Manhattan, Tanimoto, Jaccard, Mahlanobis, cosseno ou Minkowski. A distância Euclidiana, entretanto, é a mais simples e mais comumente utilizada, sendo descrita por:

$$d(x, x') = \sqrt{\sum_{i=0}^{n} (x - x')^2}, \tag{3.8}$$

onde d é a distância calculada para o dado de entrada x, com relação a outro dado x' sobre todas suas n dimensões no espaço.

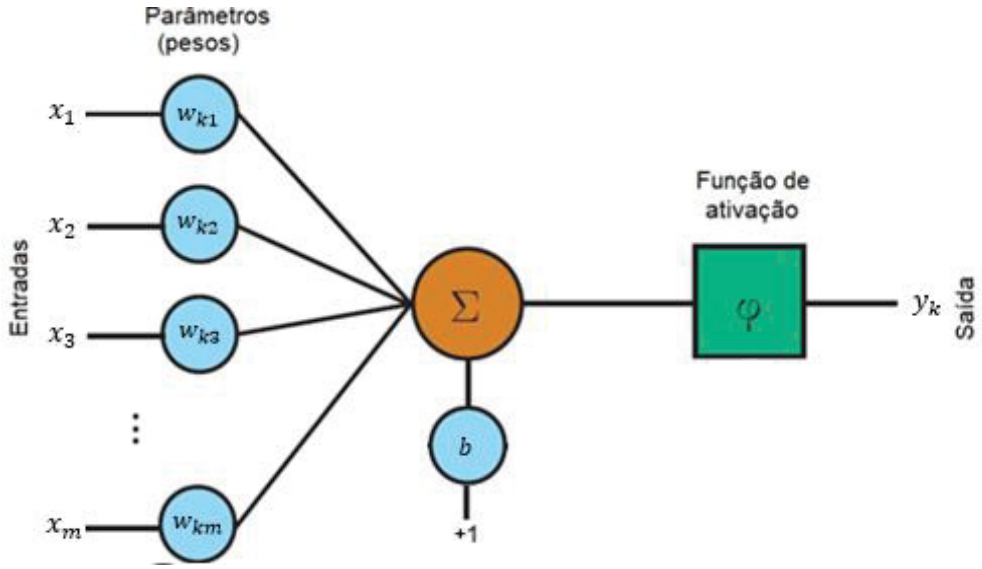
3.3 REDES NEURAIS ARTIFICIAIS

O cérebro humano é considerado o mais fascinante “processador” baseado em carbono existente, composto por aproximadamente 10 bilhões de neurônios. O sistema nervoso é formado por um conjunto extremamente

complexo de neurônios, onde estes estão conectados uns aos outros por meio das sinapses, formando uma rede neural. Sua comunicação é realizada por meio de impulsos, os quais, ao serem recebidos, são processados e após um limite de ação, é disparado um segundo impulso que produz uma substância neurotransmissora que induz uma diferença de potencial transmitida para todos os neurônios conectados em sequência (TATIBANA; KAETSU, 2017).

As redes neurais artificiais (RNAs) modelam sistemas por meio de circuitos (conexões) que possam simular o sistema nervoso humano, abrangendo a capacidade que o mesmo possui de aprender e agir perante as situações apresentadas, bem como adquirir conhecimento por meio da experiência e da observação. Uma rede neural artificial pode ser vista como um conjunto de várias unidades interconectadas (similar à estrutura do cérebro), denominadas de neurônios artificiais, cada qual contendo uma pequena porção local de memória e processamento (FLECK et al., 2016).

FIGURA 3.8 – Modelo de um neurônio artificial.



FONTE: Adaptado de (STENROOS, 2017).

Geralmente, os modelos de redes neurais possuem neurônios (também chamados de *perceptrons*) conectados em camadas, formadas por meio de uma estrutura de conexão com pesos, proporcionando uma estrutura paralela, conforme ilustrado na FIGURA 3.8. Cada um dos neurônios possui uma função de ativação φ, a qual é responsável por produzir um valor de saída y_k a partir dos m valores de entradas x_i recebidos localmente por cada unidade, ajustados

pelos pesos w_{ki} de cada neurônio. Um parâmetro de regularização b também é adicionado. A estrutura em camadas permite que as redes neurais sejam capazes de modelar funções não-lineares complexas, em troca da dificuldade de interpretação do processo de generalização interno realizado pela rede (FLECK et al., 2016). A equação para um único neurônio é descrita como:

$$y = \varphi\left(\sum_{i=1}^{m} w_i x_i + b\right) \tag{3.9}$$

Os primeiros pesquisadores descobriram que os *perceptrons* e outros sistemas lineares eram incapazes de resolver problemas que não eram linearmente separáveis. Simplesmente adicionar camadas também não ajuda, porque uma rede composta de neurônios lineares permanece linear, não importa quantas camadas ela possua (SCHMIDHUBER, 2015). Uma maneira simples e eficaz de criar uma rede não-linear se dá por meio da utilização de unidades lineares retificadas (ReLu, do inglês, *rectified linear unit*) (SCHMIDHUBER, 2015). Este tipo de função gera a saída por meio de uma função de rampa:

$$\varphi(x) = \max(0, x)\,. \tag{3.10}$$

Esse tipo de função é simples de se calcular e diferenciar (para uso em conjunto com a retropropagação). As ReLus tornaram-se populares nos últimos dez anos, muitas vezes substituindo funções de ativação sigmoidal, que possuem derivadas suaves, mas sofrem de problemas como saturação de gradiente e computação mais lenta (SCHMIDHUBER, 2015).

Para problemas de classificação multiclasse, geralmente a função de ativação *softmax* (BISHOP, 2006) é utilizada na camada de saída da rede:

$$\varphi(x) = \frac{\exp x_k}{\sum_{k=1}^{K} \exp x_k}, \tag{3.11}$$

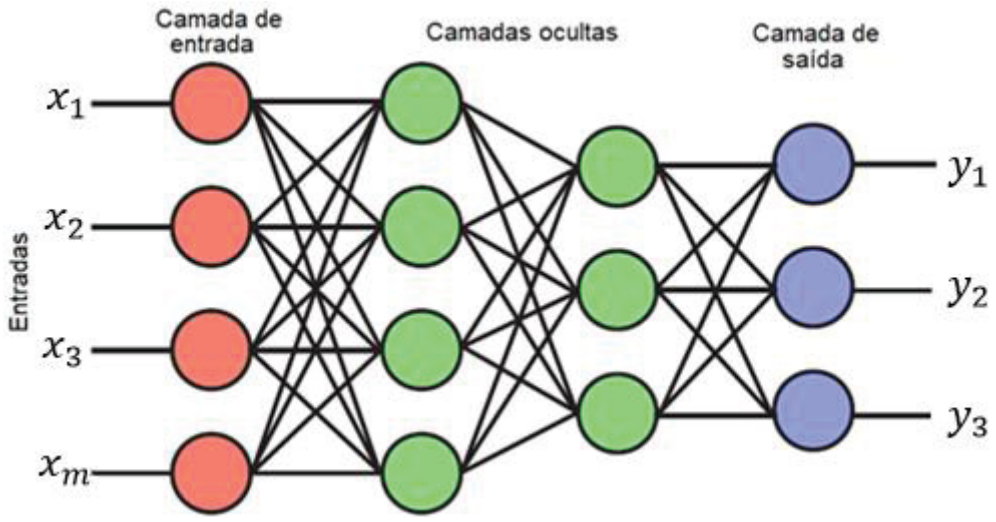
sendo x_k o vetor de dados de entrada para cada um dos K neurônios da rede. Os valores emitidos pelas unidades do tipo *softmax* podem ser aplicados como probabilidades de classe.

A arquitetura e as funções da rede são escolhidas no estágio inicial e permanecem as mesmas durante o treinamento. O desempenho da rede neural depende do valor dos pesos que cada neurônio recebe. Os pesos w_{ki} são ajustados durante o treinamento para que uma determinada saída seja alcançada. A RNA pode ser treinada utilizando diversas abordagens de

treinamento (NEOCLEOUS; SCHIZAS, 2002). Entretanto, um método de treinamento muito proeminente é o algoritmo conhecido como retropropagação (SCHMIDHUBER, 2015). Outras técnicas incluem o algoritmo de eliminação de peso, que automaticamente infere a topologia de rede, e algoritmos genéticos que tentam derivar a arquitetura de rede e treinar seus pesos (SIDDIQUE; TOKHI, 2001).

Em uma rede multicamada do tipo *feed-forward* totalmente conectada, como na FIGURA 3.9, a saída de uma camada de neurônios é empregada para alimentar as entradas de cada neurônio da próxima camada. Este tipo de rede normalmente possui três tipos de camadas: uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída (BISHOP, 2006). A camada de entrada geralmente apenas passa os dados sem modificá-los. A maior parte do cálculo acontece nas camadas ocultas. A camada de saída converte as ativações da camada oculta em uma saída, geralmente na forma de uma classe.

FIGURA 3.9 – Estrutura de uma Rede Neural do tipo *Perceptron* Multicamada.



FONTE: Adaptado de (STENROOS, 2017)

3.4 REDES NEURAIS CONVOLUCIONAIS

As Redes Neurais Convolucionais (do inglês, *Convolutional Neural Networks,* CNNs) são tipos especializados de RNAs, desenvolvidas originalmente para aplicações em processamento de imagens. Estas são, indiscutivelmente, os modelos de IA mais bem sucedidos com raízes na biologia (ZAPLETAL, 2017). Apesar de terem influências provindas de muitos campos científicos diferentes, seu núcleo de princípios fundamentais foi retirado da

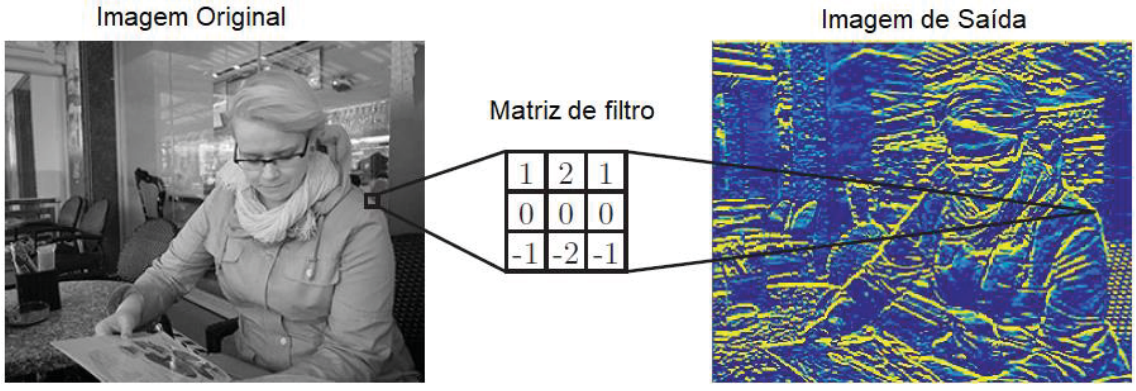
neurociência. Além do bom desempenho para processamento de imagens, este modelo de RNA também foi implementado com sucesso em aplicações de processamento de vídeo e linguagem natural.

A inspiração supracitada na biologia foi baseada nos trabalhos dos neurologistas David Hubel e Torsten Wiesel, os quais, no final de 1950, estudaram o sistema de visão de mamífero. Em seus experimentos, conectaram eletrodos no cérebro de gatos anestesiados e mediram a resposta do cérebro aos estímulos visuais (HUBEL; WIESEL, 1959). Desta forma, foram capazes de descobrir que a reação dos neurônios no córtex visual era desencadeada por uma linha estreita de luz brilhando sob um ângulo específico na tela de projeção que era apresentada ao animal. Foi determinado que os neurônios individuais do córtex visual respondem apenas a padrões específicos na imagem de entrada. Hubel e Wiesel receberam o Prêmio Nobel de Fisiologia e Medicina em 1981 por sua descoberta.

3.4.1 Operação de Convolução

A estrutura básica das CNNs foi inspirada por um conceito em biologia chamado de campo receptivo (FUKUSHIMA, 1988). Campos receptivos são um dos componentes do córtex visual animal (HUBEL; WIESEL, 1959), atuando como detectores sensíveis a certos tipos de estímulos, tais como bordas ou segmentações. Eles são encontrados em todo o campo visual e se sobrepõem uns aos outros.

FIGURA 3.10 – Detecção de bordas horizontais de uma imagem utilizando filtro de convolução.



FONTE: Adaptado de (STENROOS, 2017).

Esta função biológica pode ser aproximada computacionalmente utilizando a operação de convolução (MARR; HILDRETH, 1980). Em processamento de imagens, estas podem ser filtradas pela convolução para produzir diferentes efeitos visuais. A FIGURA 3.10 mostra como um filtro convolucional selecionado a mão detecta bordas horizontais de uma imagem, funcionando de forma semelhante a um campo receptivo.

A operação de convolução discreta entre uma imagem f e uma matriz de filtro g é definida como:

$$h[x, y] = f[x, y] * g[x, y] = \sum_n \sum_m f[n, m] g[x - n, y - m]. \quad (3.12)$$

O produto escalar do filtro g e uma sub-imagem de f (com as mesmas dimensões de g) centrados nas coordenadas (x, y) produz o valor de pixel de h nas mesmas coordenadas (GOODFELLOW et al., 2017). O tamanho do campo receptivo simulado pela operação é ajustado pelo tamanho da matriz de filtro. Alinhando o filtro sucessivamente com cada sub-imagem de f produz a matriz de pixels de saída (imagem de saída) h. No caso de redes neurais, a matriz de saída também é chamada de mapa de características, do inglês, *feature map* (ou um mapa de ativação, após o cálculo da função de ativação). A cada iteração de convolução, o tamanho da imagem de saída se torna cada vez menor em relação a imagem original (STENROOS, 2017).

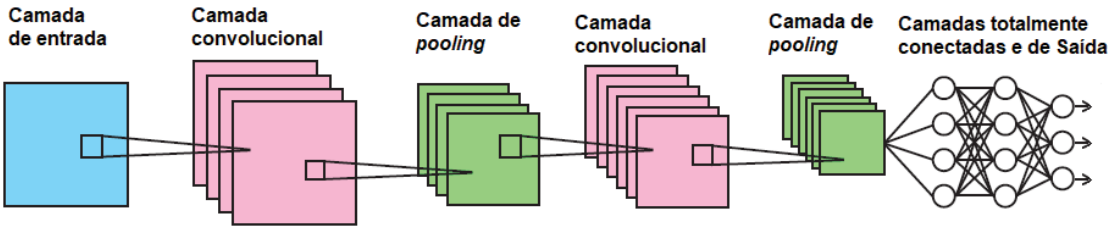
3.4.2 Estrutura Básica

A estrutura das CNNs tradicionais é tipicamente composta de três tipos diferentes de camadas, convolucionais, agrupamento (do inglês, *pooling*) e totalmente conectada. Cada tipo de camada tem regras diferentes para propagação de sinal para frente e para trás (ZAPLETAL, 2017).

Não há regras precisas sobre como a estrutura de camadas individuais deve ser organizada. No entanto, com exceção de abordagens mais recentes, as CNNs são tipicamente estruturadas em duas partes principais. A primeira, geralmente chamada de extração de recurso (do inglês, *feature extraction*), faz uso de combinações de camadas convolucionais e de agrupamento, de forma a gerar, a partir dos dados originais, o vetor de informações ou características que será analisada pela próxima parte. Esta é chamada de camada de classificação,

e é constituída por camadas totalmente conectadas, formando uma estrutura similar à de uma RNA comum. Esta estrutura é ilustrada na FIGURA 3.11.

FIGURA 3.11 – Estrutura básica de uma CNN tradicional.



FONTE: Adaptado de (STENROOS, 2017).

Uma das vantagens desse tipo de rede é o fato de receber como entrada a própria imagem, em vez de um conjunto de recursos já selecionado. A rede é capaz de aprender o conjunto de recursos que melhor modela a classificação desejada (LOPES, 2016).

Como o nome sugere, as camadas convolucionais empregam as operações de convolução efetuadas sobre os dados. A entrada da camada convolucional é uma imagem (no caso da primeira camada de rede) ou um mapa de características provindo da camada anterior. A operação de convolução é executada sobre a entrada com um filtro específico, chamado *kernel*, que é tipicamente de formato quadrado e sua largura pode variar de 3 a N pixels. O mapa de características é criado pela convolução do *kernel* sobre cada elemento do vetor de entrada. Os valores da matriz do *kernel* são tratados como parâmetros e ajustados durante o treinamento da rede. A operação de convolução substitui a operação de multiplicação de uma camada de rede neural tradicional (ZAPLETAL, 2017). Cada uma das camadas convolucionais é composta por uma série de operações de convolução, cada uma representando um neurônio da camada. Cada um destes está ligado a uma região específica da imagem havendo também uma sobreposição entre essas regiões, ou seja, uma parte de uma região pode ser uma entrada de dois ou mais neurônios (LOPES, 2016).

Como os mesmos filtros são usados para todas as partes da imagem, o número de parâmetros a serem ajustado é reduzido drasticamente em comparação com uma camada neural totalmente conectada (SCHMIDHUBER, 2015). Os neurônios da camada convolucional geralmente compartilham os

mesmos parâmetros e são conectados apenas a uma região local da entrada. Em teoria, as camadas mais próximas da entrada devem aprender a reconhecer recursos de baixo nível da imagem, como bordas e cantos, e as camadas mais próximas da saída devem aprender a combinar esses recursos para reconhecer estruturas mais significativas e abstratas (STENROOS, 2017).

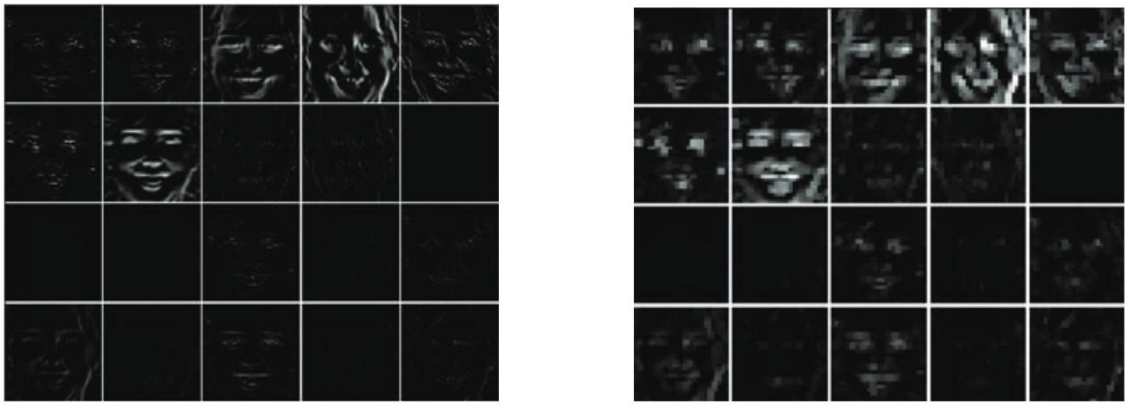
Como, geralmente, as camadas mais profundas da rede requerem menos informações específicas sobre localizações espaciais exatas de características e padrões, porém necessitam de informações de diversas localidades para reconhecer conceitos de alto nível, geralmente reduz-se o tamanho do mapa de características gerado pelas camadas convolucionais. Este processo é realizado por camadas de agrupamento (*pooling)* (GOODFELLOW et al., 2017).

Essa camada, normalmente, não constitui um processo de aprendizado, somente é usada para diminuir o tamanho dos dados da entrada. Baseia-se no princípio de que a entrada é dividida em vários elementos retangulares não sobrepostos e as unidades dentro de cada elemento são usadas para criar uma única unidade de saída. Isso diminui a dimensionalidade do vetor de saída, preservando as informações mais importantes contidas na camada de entrada. Em outras palavras, a camada de agrupamento comprime as informações contidas na entrada (ZAPLETAL, 2017). Ao reduzir a altura e a largura do volume de dados, pode-se aumentar a profundidade do volume de dados e manter os requisitos computacionais a um nível razoável. No entanto, o agrupamento pode destruir informações sobre relacionamentos entre subpartes de padrões.

O tipo de operação executada sobre cada elemento determina um tipo de camada de agrupamento. A seleção do valor máximo é o tipo mais comum de operação de agrupamento e, nesse caso, a camada é denominada *Max-Pooling* (GOODFELLOW et al., 2017).

Para o caso de imagens de expressões faciais, as camadas convolucionais da rede são capazes de identificar e extrair características marcantes da face, tais como o contorno do olhos, boca e nariz, como visto na FIGURA 3.12 (a) (LOPES, 2016). Mesmo após a aplicação da sub-amostragem via *Max-Pooling* realizada pela camada de agrupamento, vista na FIGURA 3.12 (b), estes detalhes não foram perdidos, ainda que a resolução do quadro tenha sido reduzida.

FIGURA 3.12 – Características faciais extraídas pela convolução, (a) antes da aplicação do *Max-Pooling*; (b) após a aplicação do *Max-Pooling*.



FONTE: Adaptado de (MAYYA et al., 2016).

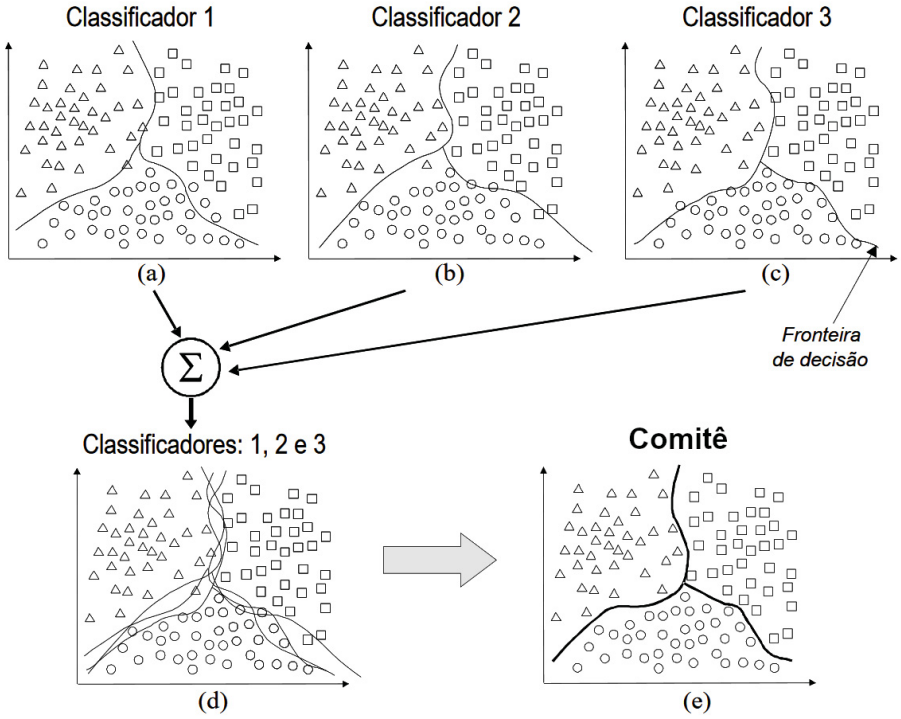
Geralmente, a última camada oculta de uma Rede Neural Convolucional é uma camada totalmente conectada. Essa camada é semelhante a camadas ocultas das RNAs. Sua entrada é dada pelos mapas gerados pelas convoluções e sub-amostragens realizadas pelas camadas anteriores, e a saída, geralmente, representa as classes desejada (LOPES, 2016).

3.5 COMITÊ DE MÁQUINAS

Em um problema padrão de classificação por AM, dado um conjunto de dados de treinamento, o algoritmo de aprendizado tem a tarefa de gerar um classificador de saída, o qual é uma hipótese sobre a real função matemática que mapeia as características de entrada a sua respectiva classe. Gerando-se vários modelos diferentes, seleciona-se o que fornece a maior precisão (ou menor erro) para a aplicação. Porém, em alguns casos, a precisão pode ser aumentada utilizando a combinação da saída de diversos classificadores (DIETTERICH, 2000).

Um comitê de máquinas é um grupo de modelos que têm suas decisões combinadas de alguma forma (tipicamente por meio de votação ponderada ou não) para classificar um novo exemplo. Consistem de um conjunto de técnicas de aprendizado de máquina que buscam agregar o conhecimento adquirido pelos modelos que o compões a fim de atingir uma solução global que resulte em um modelo mais eficiente que o de seus componentes aplicados sozinhos (VILLANUEVA, 2006).

FIGURA 3.13 – Exemplo de classificação utilizando comitês de máquinas



FONTE: Adaptado de (VILLANUEVA, 2006).

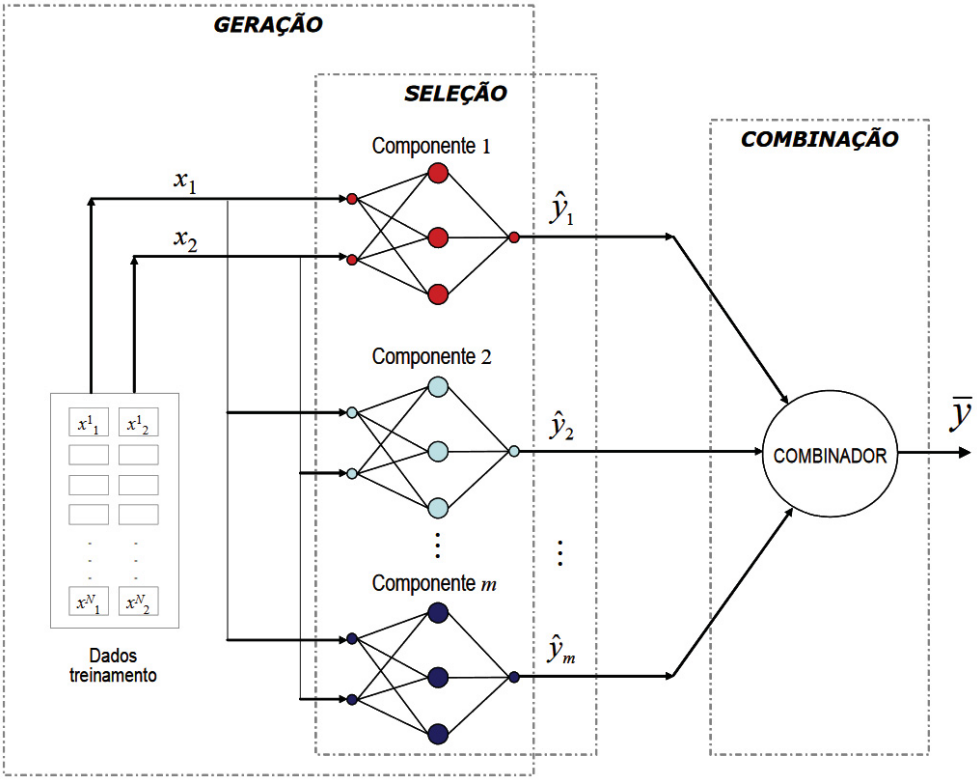
O comitê é definido pela seleção de *k* modelos e pela função que realiza a combinação dos mesmos. Para tal existem diferentes abordagens que podem ser seguidas. Os classificadores podem ser extraídos do treinamento a partir de um mesmo conjunto ou de várias sub-amostras do conjunto de dados. Pelo fato dos algoritmos de aprendizados realizarem, essencialmente, uma busca, em um espaço de hipóteses, pela hipótese que mais se adequa aos dados, quando não se possui um conjunto de dados grande o suficiente, o algoritmo pode acabar chegando a diversos pontos que são “igualmente ideais” para seu conjunto de treinamento. A construção de um comitê de máquinas permite que todas essas hipóteses sejam consideradas em uma “hipótese média” (DIETTERICH, 2000). Essa situação é representada na FIGURA 3.13.

As técnicas de geração de comitês de máquinas são compostas por uma metodologia de três etapas, treinamento (geração), seleção e combinação, conforme ilustrado na FIGURA 3.13.

Na etapa de treinamento, os modelos participantes (componentes) do comitê são gerados pelos classificadores fracos, a partir dos dados do conjunto de treinamento.

Na etapa de seleção, os componentes com melhor desempenho são selecionados. Ao usar um número maior de componentes, é possível que nem todos os componentes contribuam para o desempenho global do conjunto, portanto, um passo de refino é recomendado para reduzir o conjunto de acordo com um critério de seleção, que pode ser uma medida de erro sobre uma subamostra de dados (VILLANUEVA, 2006).

FIGURA 3.13 – Etapas de construção de um comitê de máquinas.



FONTE: Adaptado de (VILLANUEVA, 2006).

Na etapa de combinação, o método combinatório empregado é diferente de acordo com o tipo de problema em que será aplicado. Em um problema de classificação, pode ser usada uma técnica de votação, ou, no caso de um problema de regressão, geralmente é usada uma média dos resultados resultantes de cada componente.

Além disso, pode-se gerar os classificadores utilizando um mesmo algoritmo de AM, denominados modelos homogêneos, ou por meio de algoritmos diferentes, conhecidos como modelos heterogêneos. Dentre as técnicas mais frequentemente aplicadas para a combinação de modelos homogêneos se encontram os métodos de *bagging* (BREIMAN, 1996) e *boosting* (KEARNS,

1993). Já para modelos heterogêneos, o mais empregado é o método *stacking* (PUGLIESI et al., 2003).

O algoritmo *Bootstrap Aggregating* (*Bagging*) vota classificadores gerados por diferentes amostras de *bootstrap*. Uma amostra de *bootstrap* é gerada pela sub-amostragem uniforme de *m* instâncias do conjunto de treinamento. A partir destas, um modelo classificador intermediário é construído sobre cada uma das subamostras de dados. Um classificador final é construído a partir dos classificadores intermediários, cuja saída é selecionada por maioria de votos das classes de saída dos sub-classificadores (LI et al., 2014)

O algoritmo de *Boosting*, consiste de uma combinação de modelos matemáticos fracos que são construídos iterativamente de forma sequencial, cada um sendo treinado com diferentes subconjuntos do conjunto de dados original, sem substituição (LI et al., 2014). A ideia principal é que cada novo modelo gerado é capaz de aprender com os erros do modelo anterior e classificar mais corretamente, principalmente, estes dados.

O *Stacking Generalization* também chamado *Stacking*, usa o conceito de meta-aprendizado, ao invés de algoritmos de votação para combinar os modelos base. Os modelos dos algoritmos fracos são gerados em paralelo e combinados na formação de um meta-modelo, que aprende com as predições geradas pelos modelos base (LI et al., 2014).

3.6 REDUÇÃO DE DIMENSIONALIDADE

Em problemas de aprendizado de máquina, é importante possuir dados de boa qualidade e com menor presença de ruído. Os avanços na capacidade de coleta e armazenamento de dados nas últimas duas décadas resultaram em coleções de dados maiores do que se utilizava no passado. Isto significa que tanto o número de objetos de dados quanto a dimensão destes objetos também cresceram. No entanto, como afirmado por TANG et al. (2014), um maior conjunto de dados possui também desvantagens, tais como a existência de dados com mais ruído. Além disso, a existência de mais variáveis não garante que todas elas expressem informações relevantes ao problema estudado (TELI, 2007).

A grande dimensionalidade dos dados pode causar problemas, informalmente conhecidos como “maldição da dimensionalidade”, termo

introduzido por Richard Bellman, em 1954 Trabalhar com dados de alta dimensão é um processo computacionalmente intenso e lento, podendo também resultar em desempenho inferior comparado ao modelo gerado a partir de um espaço dimensional menor. A redução do número de recursos de dados pode ajudar a melhorar o desempenho de aprendizado, criar modelos mais generalizáveis, diminuir o armazenamento necessário e permitir a visualização os dados (BURGES, 2009).

O objetivo da redução de dimensionalidade é transformar os dados de alta dimensão em um espaço dimensional menor, mantendo as propriedades importantes dos dados para o problema em consideração. Um benefício secundário da redução de dimensionalidade é que alguns ruídos e redundâncias nos recursos são eliminados após a redução do espaço do recurso. Como resultado, os dados podem ser representados de maneira mais compacta e eficaz, o que pode melhorar o desempenho do classificador (COSSETIN, 2015).

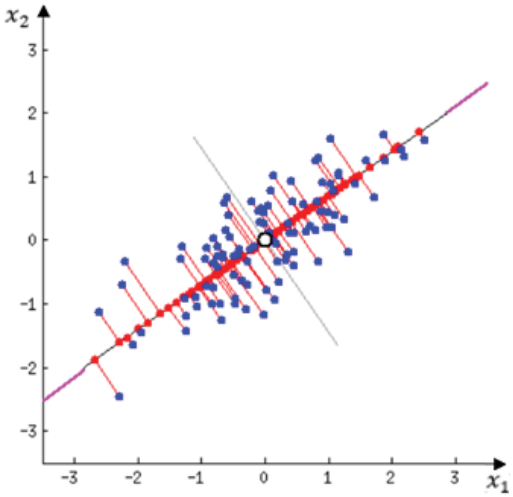
Conjuntos de dados de imagem e vídeo são inerentemente grandes e, muitas vezes, inconvenientes para manipular, principalmente em computadores com limitação de memória disponível. Mesmo com a extração de características e descritores de imagens, tais como HOG e LBP, ainda é benéfico do ponto de vista computacional que as dimensões sejam reduzidas. Além disso, nem todas elas podem ser relevantes para discriminar as diferentes expressões faciais, por exemplo.

3.6.1 Análise de Componentes Principais (PCA)

A Análise de Componentes Principais foi introduzida pela primeira vez por Pearson em 1901 e é o método de redução de dimensionalidade mais comumente utilizado na literatura. O algoritmo emprega medidas de variação para extrair as estruturas dimensionais mais relevantes na distribuição de um conjunto de dados. Assim, todas as dimensões com um determinado nível de relevância (medida de variação) podem ser extraídas, ou um número definido de dimensões mais relevantes pode ser delineado. Um exemplo de PCA aplicado em dados bidimensionais pode ser observado na FIGURA 3.14, onde a componente principal, que melhor representa os dados das dimensões fictícias x_1 e x_2, recebe as projeções dos dados, tornando-os unidimensionais. O primeiro componente principal para as dimensões x_1 e x_2 é representado pela linha com

as extremidades em magenta. Pontos vermelhos são a projeção dos dados no primeiro componente principal.

FIGURA 3.14 – Ilustração do PCA.



FONTE: Adaptado de (TELI, 2007).

Na prática, dado um conjunto de dados $X \leftarrow D^{mxn}$, o primeiro passo do algoritmo é normalizar todos os dados do conjunto, de forma que todos existam em faixas de valores similares. Este passo evita que dados que possuam faixas de valores maiores "dominem" valores com variações menores, evitando tendências na análise dos dados. Para tal normalização, procede-se primeiro calculando a média dos vetores x^n:

$$\mu^m = \frac{\sum_{i=1}^{n} x_i}{n} . \tag{3.13}$$

Cada instância de dados x^n tem, então, seu valor subtraído da média μ^m e dividido pelo desvio padrão dos dados σ:

$$x_{norm}^n = \frac{x^n - \mu^m}{\sigma}. \tag{3.14}$$

Então, a matriz de covariância é calculada para X:

$$\Sigma = \begin{pmatrix} cov(x_1, x_1) & \cdots & cov(x_1, x_m) \\ \vdots & \ddots & \vdots \\ cov(x_m, x_1) & \cdots & cov(x_m, x_m) \end{pmatrix}, \tag{3.15}$$

onde $cov()$ representa a covariância entre duas observações:

$$cov(x, y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{n - 1} \tag{3.16}$$

A covariância é uma medida do grau de interdependência (ou inter-relação) numérica entre duas variáveis aleatórias e serve como uma medida de

quanto duas variáveis mudam uma em relação a outra. Caso o sinal do valor de saída seja positivo, as variáveis têm alteração proporcional. Caso seja negativo, as variáveis têm alterações opostas. Como a covariância possui propriedade comutativa, a matriz produzida é simétrica com relação a sua diagonal principal.

A partir da matriz de covariância Σ gerada, são calculaos os autovalores λ e autovetores e, os quais são utilizados para definir os componentes principais dos dados:

$$\lambda = \det(\Sigma - \lambda I)\,, \tag{3.17}$$

onde I é a matriz identidade, definida por:

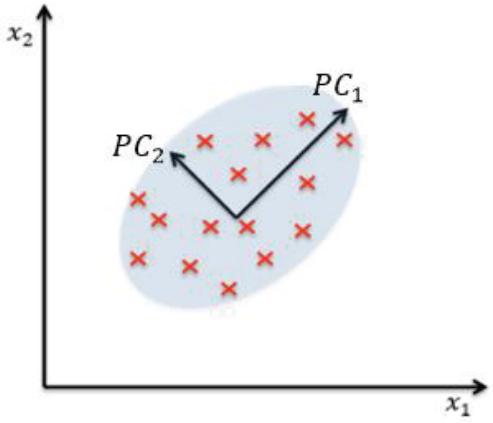
$$\Sigma e = \lambda e\,. \tag{3.18}$$

Componentes principais são novas variáveis construídas por combinações lineares ou misturas das variáveis iniciais. Essas combinações são feitas de tal maneira que as novas variáveis (os componentes principais) não são correlacionadas e grande parte das informações contidas nas variáveis iniciais é comprimida nos primeiros componentes produzidos (FIGURA 3.15). O autovalor associado a cada autovetor representa a variância explicada (quantidade de informação representada) pelo mesmo. Os autovetores são ordenados de acordo com a maior representação de variância do conjunto de dados.

O último passo é a produção das novas variáveis com dimensionalidade reduzida. Para tal, os autovetores correspondentes aos primeiros *N* componentes principais escolhidos são retidos, de forma a selecionar os que codificam maior parte possível da informação, e os dados iniciais x^n são projetados nestes autovetores para gerar os componentes dos vetores transformados z^n no novo espaço M-dimensional, onde x^n é o conjunto de dados de coordenadas de alta dimensão e z^n é a projeção no espaço de baixa dimensão dos n pontos do conjunto de dados.

A observação da relação de dados inerente aos autovalores da matriz de covariância permite estimar a dimensionalidade “intrínseca” da distribuição de dados comparando os autovalores resultantes e trabalha sob a suposição de que valores pequenos de correlação para esta “dimensão principal” denota dimensões que podem ser removidas gerando o mínimo perda de coerência e informação (BISHOP, 2006).

FIGURA 3.15 – Representação dos componentes principais PC_1 e PC_2.



Fonte: O autor (2018).

A análise de componentes principais pode ser usada para a redução da dimensão selecionando os autovetores mais importantes (correspondentes aos maiores autovalores). Os autovetores selecionados então criam uma matriz de transformação que adapta os dados originais a um espaço de menor dimensão.

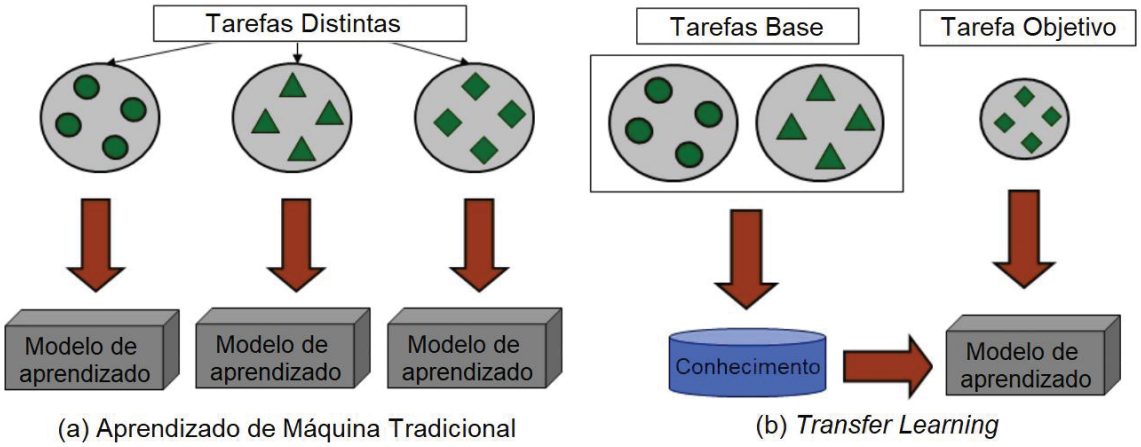
3.7 APRENDIZADO POR TRANSFERÊNCIA

O estudo do aprendizado por transferência (do inglês, *transfer learning*) foi inspirado pela capacidade dos seres humanos de aplicar experiências e conhecimentos adquiridos anteriormente em novas situações em diferentes ambientes. O paradigma da aprendizado por transferência implica em reutilizar máquinas com modelos previamente treinados para um determinado problema fonte S, com pequenas modificações, para resolver um problema-alvo diferente T. Um método de aprendizado por transferência ideal deve ser capaz de melhorar o classificador reutilizado em comparação com o que treinado a partir do zero (KANDASWAMY, 2016). Em linhas gerais, o algoritmo de transferência de conhecimento explora fatores de similaridade entre as tarefas, de forma a identificar pontos de conhecimento comuns que possam ser adaptados para a nova tarefa.

A FIGURA 3.16 apresenta a diferença fundamental entre os processos de aprendizado das técnicas tradicionais de AM e de aprendizado por transferência. As técnicas tradicionais buscam gerar modelos para cada tarefa desde o princípio, enquanto as técnicas de transferência de aprendizado tentam adaptar o conhecimento de algumas tarefas anteriores para uma tarefa objetivo.

Os métodos de aprendizado por transferência baseiam-se em uma suposição importante: os padrões extraídos no conjunto de dados original são úteis no contexto do novo conjunto de dados.

FIGURA 3.16 – Diferenças entre o processo de aprendizado de técnicas tradicionais de AM e aprendizado por transferência.



FONTE: Adaptado de (PAN, 2010).

Dado um conjunto de dados $D = \{(x_i; y_i)\}_{i=1}^{N}$, onde o domínio de variáveis de entrada é denominado X e um conjunto de rótulos Y, um classificador é definido como qualquer função $f(x) : X \rightarrow Y$ que mapeia instâncias $x_i \in X$ para algum dos rótulos em Y. O desempenho do classificador, tais como a taxa de erro de previsão ε e o tempo computacional, são medidos em um conjunto de testes X_{test} com v instâncias não rotuladas extraídas de uma mesma distribuição $P(X)$.

Por definição, o objetivo da aprendizado por transferência é adaptar o conhecimento adquirido a partir do conjunto de dados de entrada X_S de um problema base, de forma a produzir uma hipótese efetiva para uma nova tarefa ou problema (BRUZZONE; MARCONCINI, 2010). Nessa estrutura de aprendizado, existem casos onde os problemas de origem e destino podem derivar de escopos ou objetivos similares e em outros casos, escopos distintos. No aprendizado supervisionada, os rótulos Y_S de origem e Y_T de destino podem ser iguais ou diferentes.

O uso do aprendizado por transferência se torna importante em situações onde não existe grande quantidade de dados para o treinamento de um modelo de forma confiável. Sob esta condição, a grande maioria dos

algoritmos de AM voltados para aprendizado supervisionado tradicional podem “colapsar” e não conseguem apresentar bons resultados.

FIGURA 3.17 – Diferentes domínios visuais.

FONTE: Adaptado de (TORRALBA; EFROS, 2011).

Outra aplicação de aprendizado por transferência é na adaptação de domínios, o qual é o foco deste trabalho. A adaptação do domínio é um requisito comum na visão computacional, pois é comum que os dados em que a informação rotulada esteja facilmente acessível não sejam similares aos dados relevantes ao problema no qual o modelo vai ser aplicado. Mesmo que o treinamento e os dados do teste pareçam iguais, os dados de treinamento ainda podem conter um viés imperceptível para os seres humanos, mas que o modelo explorará para se alcançar um melhor ajuste a estes (TORRALBA; EFROS, 2011). A FIGURA 3.17 apresenta exemplo de treinamento de um objeto em domínios distintos. A presença de um fundo, como nas imagens do domínio 2, podem confundir um algoritmo treinado somente nos dados do domínio 1, apesar das características extraídas para identificação do objeto em si serem similares.

4 ANÁLISE DE EXPRESSÕES FACIAIS

A maioria dos sistemas computacionais ignora o fato de que as comunicações realizadas entre seres humanos sempre se dão socialmente e que as emoções têm papel importante na transmissão das informações de uma pessoa para outra. Sistemas de interação humano-computador que possuem a capacidade de sentir os estados emocionais e afetivos de um ser humano (como estresse, raiva, tédio, desatenção ou felicidade, por exemplo) e são capazes de se adaptar e responder a estes estímulos são suscetíveis a serem percebidos como mais naturais, eficazes e confiáveis. Picard (1995) sugere diversas aplicações onde o reconhecimento de emoções humanas se mostra benéfica, dentre as quais podemos citar o fato de um discurso sintetizado por meio de uma voz que passe emoções se torna mais agradável de ouvir do que uma simples dissertação monótona.

Os agentes computacionais podem implementar a capacidade de aprender novos comportamentos e atualizar as preferências de usuário por meio das emoções apresentadas. Outra aplicação é ajudar o usuário a monitorar seu nível de estresse. Em contextos clínicos, reconhecer a incapacidade de uma pessoa expressar certas expressões faciais pode ajudar a diagnosticar transtornos psicológicos de forma precoce.

A análise das emoções humanas fazendo uso de expressões faciais é parte integrante da pesquisa psicológica. Por muito tempo, acreditava-se que personalidade de uma pessoa poderia ser inferida a partir de uma análise de sua fisionomia ou aparência externa, especialmente pela região da face e dos olhos (BREUER; KIMMEL, 2017). Esta hipótese foi refutada por falta de apoio científico de muitos, dentre os quais Leonardo Da Vinci. No século XVII, John Bulwer publicou o primeiro trabalho conhecido sobre o mecanismo muscular que rege o controle das expressões faciais, em seu livro “Pathomyotomia, or, Dissection of the Significant Muscles of the Affections of the Mind” (BULWER, 1649). Alguns séculos depois, na França, Duchenne de Boulogne realizou estudos sobre a produção de expressões faciais em seres humanos (DUCHENNE; CUTHBERTSON, 1990). Este publicou imagens de expressões faciais obtidas por estimulação elétrica dos músculos faciais, conforme apresentado na FIGURA 4.1.

FIGURA 4.1 – Imagens das experiências de Duchenne de Boulogne. As Expressões faciais dos voluntários foram obtidas pelo do estimulo dos músculos faciais por eletrodos.

FONTE: Adaptado de (BREUER; KIMMEL, 2017).

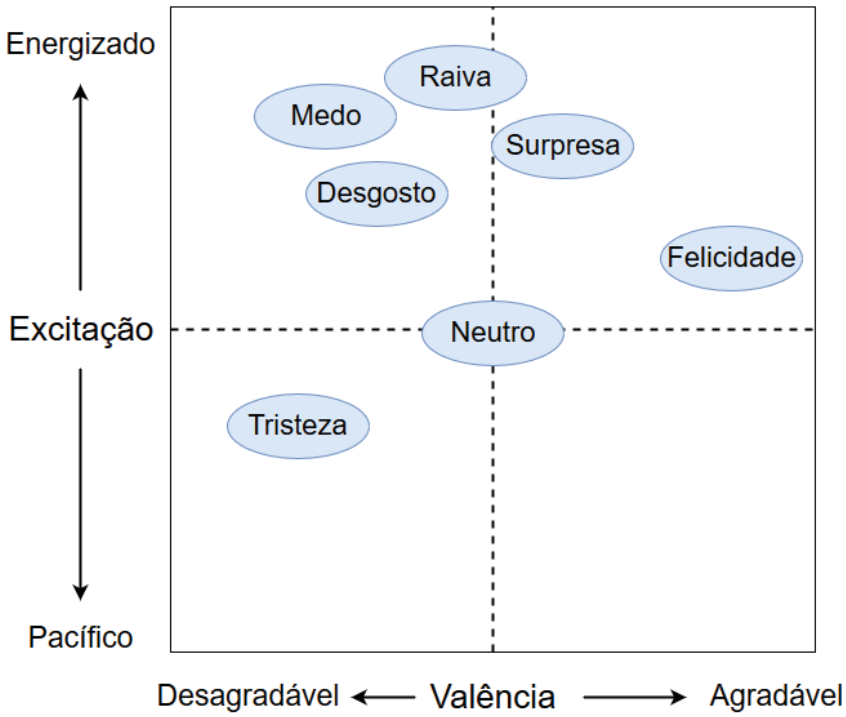
Ao final do século XIX, Charles Darwin (DARWIN; CUMMINGS, M. M., DUCHENNE, 1872) sugeriu que as expressões faciais seriam parte de reflexos comportamentais mais complexos aos estímulos ambientais. Em situações onde uma pessoa apresenta expressão de desgosto, a contração das narinas é um instinto utilizado pelo corpo para reduzir a inalação de substâncias nocivas ou venenosa. A acentuada abertura dos olhos em situações de susto ou surpresa permite o aumento do campo visual de forma a facilitar a visualização de situações perigosas ou riscos iminentes, por exemplo.

Inspirado pela base evolutiva de Darwin para expressões, Ekman, em 1982, realizou seu estudo sobre expressões faciais, onde identificou seis expressões universais primárias, ao quais são independentes de fatores pessoais, tais como gênero e cultura (EKMAN, 1994). Ekman rotulou os estados emocionais correspondentes como felicidade, tristeza, surpresa, medo, desgosto e raiva. Devido à sua simplicidade e reivindicação de universalidade, a hipótese das emoções primárias tem sido amplamente explorada na computação cognitiva.

O principal método empregado pelos pesquisadores para caracterizar emoções humanas parte da classificação das emoções em diversas categorias discretas (ZENG et al., 2009) (FIGURA 4.2), etiquetando-as em grupos tais como felicidade, tristeza, surpresa, medo, entre outras, aplicando diferentes vetores de

características como entrada para o sistema de classificação. Esta metodologia sofre com o problema de que, em muitos casos, os estímulos que representam as emoções são tem fronteiras claras que definem a mudança de uma emoção para outra, além de existirem estímulos que apresentam emoções misturadas. Outro fator relevante é de que a escolha dessas categorias pode ser restritiva ou culturalmente dependente (JAIMES; SEBE, 2007).

FIGURA 4.2 – Distribuição das seis expressões faciais básicas dentre os domínios de Valência e Excitação.

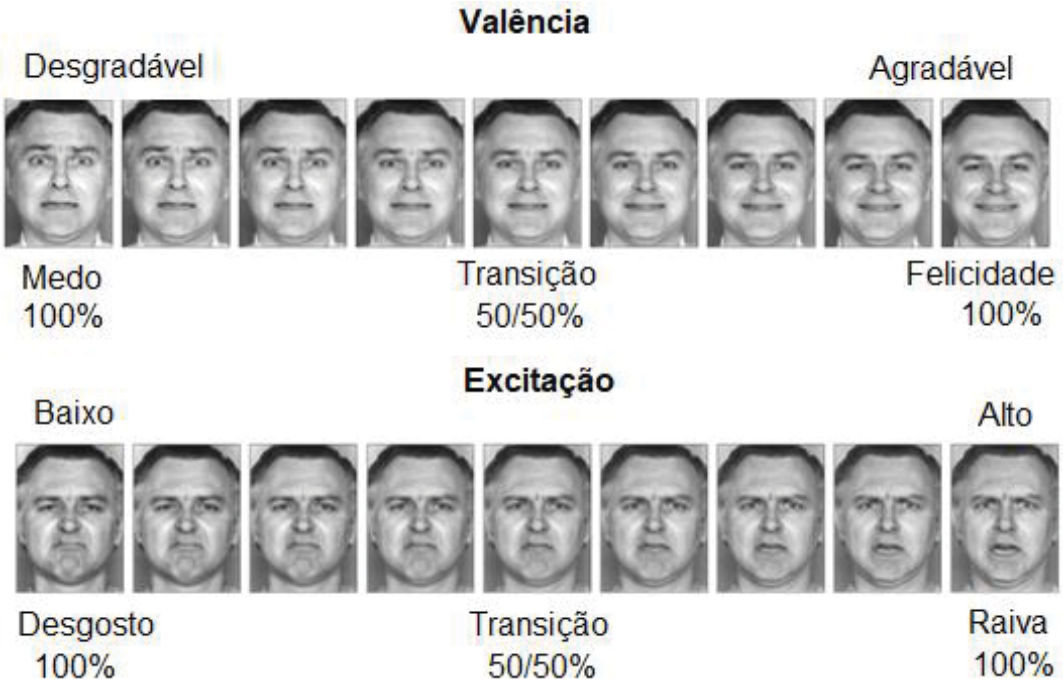


FONTE: Adaptado de (ZHANG et al., 2014).

A outra opção de abordagem é por meio da utilização de múltiplas dimensões para descrever as emoções (FIGURA 4.3). Hanajalic e Xu (2005) representam as emoções humanas por meio de três dimensões básicas, Valência (*Valence*), Excitação (*Arouse*) e Controle ou Dominância (*Dominance*). Valência é tipicamente caracterizada como um intervalo contínuo de estados afetivos que se estendem de agradável ou "positivo" para desagradável ou "negativo", enquanto a excitação é caracterizada por estados afetivos variando em uma escala contínua de energia, como excitado ou alerta, até calmo, sonolento ou pacífico. Também é possível interpretar a Valência como sendo relacionada ao “tipo” da emoção, enquanto a Excitação representa a “intensidade”. A terceira dimensão-controle, a dominância, é particularmente útil

para distinguir entre estados emocionais com excitação e valência semelhantes (por exemplo, diferenciando entre "sofrimento" e "raiva") e tipicamente varia de "sem controle" para "controle total". Dessa forma, todo o escopo das emoções humanas pode ser representado como um conjunto de pontos no espaço de coordenadas tridimensionais (PANTIC et al., 2005; ZENG et al., 2009).

FIGURA 4.3 – Variação da expressão facial nos domínios Valência e Excitação.

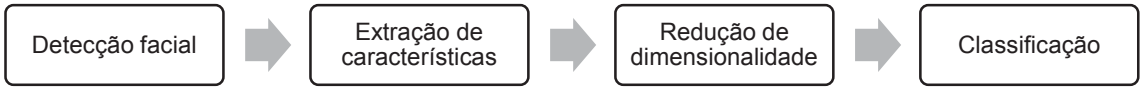


FONTE: Adaptado de (MATSUDA et al., 2013).

4.1 TÉCNICAS DE IDENTIFICAÇÃO DE EXPRESSÕES FACIAIS

O termo “identificação de expressões faciais” está ligado a sistemas que objetivam analisar automaticamente os movimentos e características faciais a partir de informações visuais, de forma a classificar os dados analisados em um tipo de expressão facial (LI; JAIN, 2011).

FIGURA 4.4 – Fases da análise automática de expressões faciais.

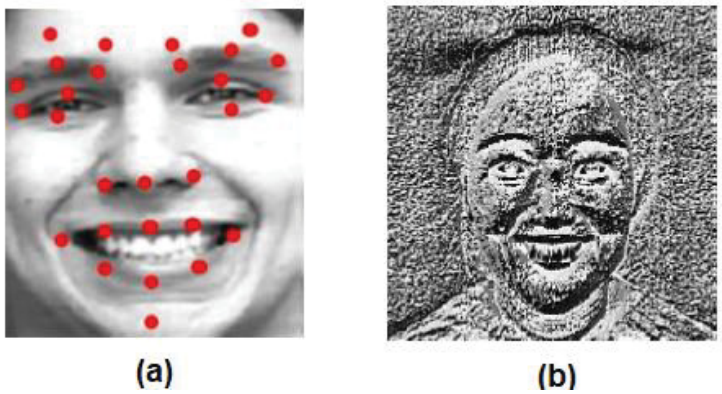


Fonte: O autor (2018).

O processo de análise automática de expressões faciais geralmente emprega quatro etapas principais, como visto na FIGURA 4.4 (LI; JAIN, 2011). A aquisição das faces dentro das imagens pode ser separada em dois passos

principais: detecção da posição das faces (ZHANG et al., 2012) e estimação de pose da cabeça (LI et al., 2013).

FIGURA 4.5 – Exemplo de extração de características baseada em (a) geometria e (b) textura.



FONTE: Adaptado de (COSSETIN, 2015).

Depois que o rosto for localizado, as alterações faciais causadas pelas expressões faciais são extraídas. A extração de dados faciais é um passo vital para o reconhecimento da expressão facial com sucesso. Soltanpour et al. (2017) realizou uma revisão dos principais métodos disponíveis na literatura. Uma extração de características ineficaz gera dados com poucas informações relevantes para o modelo, causando também um reconhecimento ineficiente da expressão. A representação da expressão facial é realizada principalmente de duas formas: utilizando características baseadas em geometria ou baseadas em textura (LI; JAIN, 2011). Métodos baseados na geometria da face (FIGURA 4.5 (a)) analisam aspectos relacionados à forma, localização e as distâncias entre componentes faciais, tais como boca, olhos, sobrancelhas e nariz (AIRES et al., 2014; TARNOWSKI et al., 2017). Estas características são extraídas da imagem da face de forma a gerar um vetor de dados que representa a geometria da face. Por outro lado, os métodos baseados em textura (FIGURA 4.5 (b)) realizam a extração de dados por meio da aplicação de filtros de imagem em regiões específicas da imagem da face (SHAN et al., 2009; ZHANG et al., 2012; CARDIA NETO, 2014).

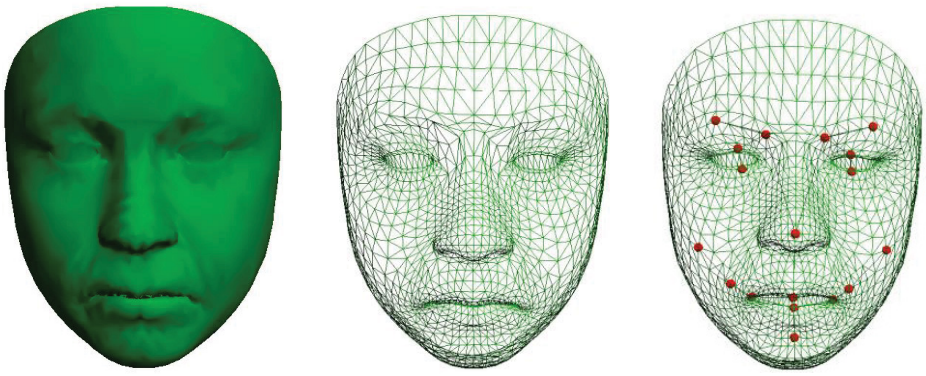
Uma vez que os vetores de característica referentes à expressão facial são gerados, o processo de classificação da expressão pode ser iniciado. Normalmente, este passo é realizado utilizando alguma abordagem de AM., porém, anteriormente à aplicação destes, são empregados métodos de redução

de dimensionalidade sobre os vetores de dados extraídos das imagens. São selecionados apenas os recursos mais discriminativos para a identificação de cada expressão, possibilitando obter uma considerável redução no custo computacional e aumento no desempenho de classificação (COSSETIN, 2015). O conjunto de características selecionado deve minimizar a variação intra-classe (mesma expressão) da expressão, maximizando a variação entre classes (expressões diferentes) (SHAN et al., 2009).

Após este pré-processamento, o método de AM é responsável por analisar os atributos restantes e gerar um modelo de conhecimento que seja capaz de identificar corretamente as expressões em dados extraídos de novas imagens não vistas anteriormente. Como descrito por Li et. al (2011), na maioria dos casos, os sistemas de reconhecimento de expressões faciais recebem uma ou mais imagens de entrada e geram uma classificação de expressão facial, tais como: neutra, raiva, tristeza, surpresa, feliz, desgosto e medo.

No caso de sistemas que fazem uso de informações tridimensionais, o modelo de face é capaz de capturar informações mais precisas quanto a geometria da superfície facial. Várias informações adicionais de superfície podem ser extraídas de modelos 3D, tais como nuvens de pontos (FIGURA 4.6, imagens de intervalo ou informações mais específicas, tais como normais, tangentes ou curvaturas, todas as quais contribuem para uma descrição mais detalhada das variações na superfície tridimensional da face (ZUCKER, 2006).

FIGURA 4.6 – Processo de localização de pontos fiduciais em dados faciais tridimensionais.



FONTE: Adaptado de (CHANTHAPHAN et al., 2016).

Enquanto os estágios para processamento, extração e classificações de informações deste tipo continuam sendo os mesmos, as técnicas utilizadas sofrem ligeiras modificações, pois dada a diferença intrínseca da forma como

estes dados são capturados, técnicas comuns de processamento de imagens não são efetivas quando aplicadas diretamente em informações de profundidade.

4.2 EXTRAÇÃO DE CARACTERÍSTICAS

Neste item são apresentadas as técnicas aplicadas para análise de dados RGB-D e extração dos vetores de características que alimentam os algoritmos de classificação baseados em AM.

As técnicas avaliadas são baseadas na extração de características locais das informações faciais tridimensionais. Esse tipo de características possui algumas vantagens sobre as extrações globais, quando aplicadas na tarefa de reconhecimento de expressões faciais, principalmente, por serem mais sensíveis a micro variações na textura facial.

4.2.1 Padrões Binários Locais

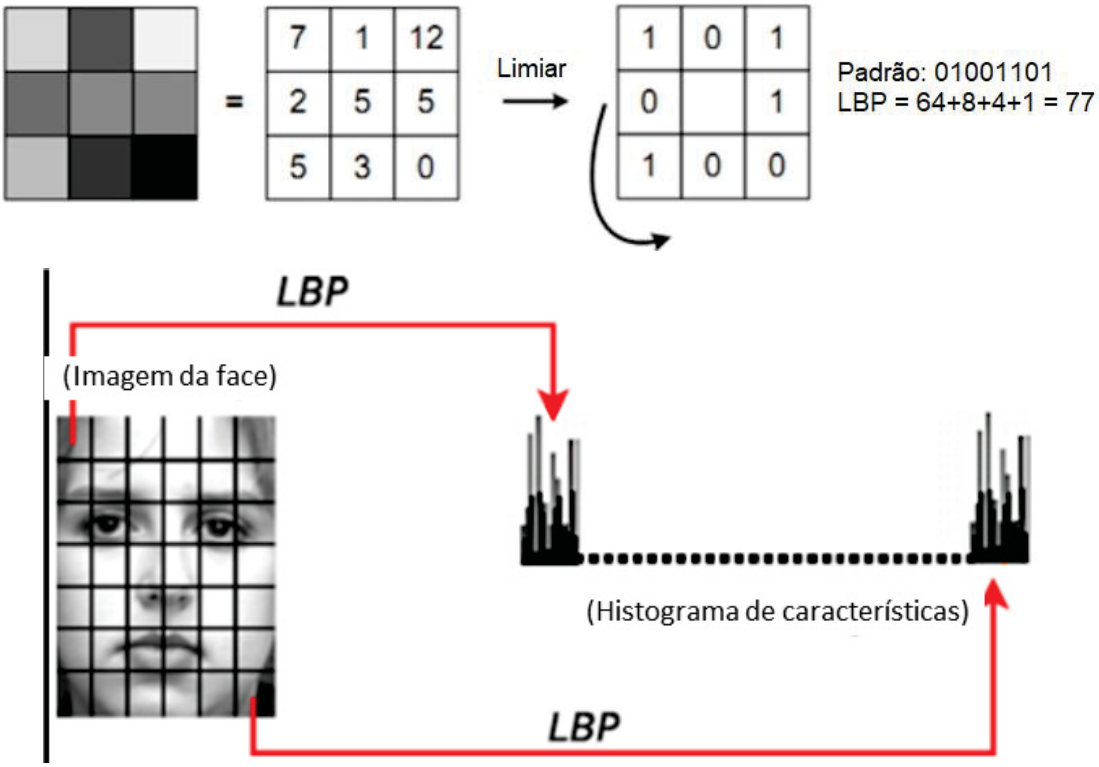
Originalmente proposto por Ojala et al. (2002) para análise de textura, a extração por Padrões Binários Locais (do inglês, *Local Binary Pattern,* LBP) têm apresentado bom desempenho aplicado em diversos segmentos, desde reconhecimento de face (AHONEN et al., 2006), análise de expressão facial (COSSETIN, 2015), a estimativa de idade (GÜNAY; NABIYEV, 2008), classificação de gênero e etnia (HUYNH et al., 2013), entre outros.

O LBP é um algoritmo que extrai informações de imagens a partir das texturas. As expressões faciais produzem pequenos padrões que são representados pelas mudanças na textura da face, como rugas e sulcos. As pequenas variações que ocorrem em função de cada expressão podem ser obtidas aplicando o LBP (COSSETIN, 2015).

Foi originalmente proposto como um operador simples que limiariza as diferenças do valor central e da vizinhança na grade 3x3 em torno de um pixel e codifica o sinal das diferenças entre este e seus vizinhos ao longo da imagem, produzindo como resultado um padrão binário de 8 bits representado para esse pixel. O histograma desses números binários na imagem inteira pode ser usado como um descritor para a imagem, como visto na FIGURA 4.7 (HUYNH et al., 2013; COSSETIN, 2015). Os padrões analisados permitem a identificação de

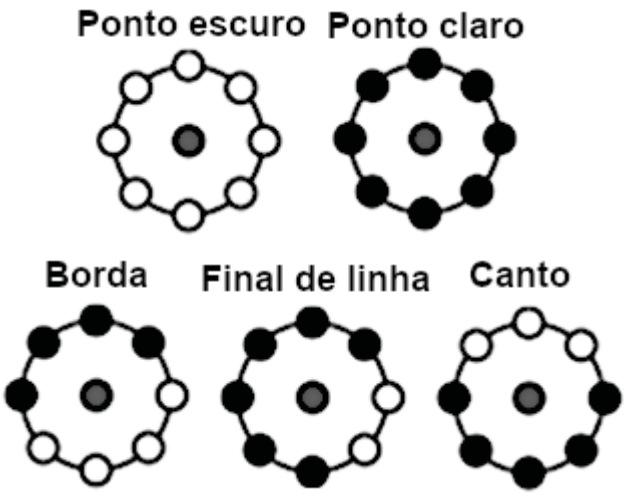
algumas características de alto nível abstracional dentro da imagem, conforme apresentado na FIGURA 4.8.

FIGURA 4.7 – Exemplo do operador LBP original.



FONTE: Adaptado de (HUYNH et al., 2013).

FIGURA 4.8 – Padrões identificáveis por meio do LBP.



FONTE: Adaptado de (HUYNH et al., 2013).

O operador foi estendido e generalizado para funcionar com qualquer raio e número de pontos na vizinhança. A notação (P, R) indica o uso de P pontos

de amostra na vizinhança, em um círculo de raio R. O valor do código LBP no pixel (x_c, y_c) é dado por:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \ , \tag{5.1}$$

onde g_c é o valor em escala de cinza do pixel central (x_c, y_c), g_p são os valores de cinza dos P pixels no raio R e s define a função de limiar como segue:

$$s(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{caso contrário} \end{cases} \tag{5.2}$$

O LBP é uma abordagem poderosa para analisar e discriminar texturas. No entanto, este processo apenas considera o sinal das diferenças ignorando os valores absolutos das diferenças, que também podem ser uma importante fonte de informação. Além disso, quando se trata de dados faciais em 3D, as diferenças de profundidade das superfícies podem levar à existência de dois padrões de textura distintos com o mesmo código binário LBP definido. As diferenças de profundidade em um mesmo ponto das superfícies faciais permitem a distinção de diferentes faces (HUANG et al., 2006).

Como apresentado na FIGURA 4.9, embora A e B sejam duas pessoas diferentes, a LBP de suas pontas do nariz são as mesmas, pois todos os pontos ao redor das pontas do nariz são “inferiores” a eles. É possível observar que, se duas regiões faciais de pessoas distintas, em um mesmo local, têm a mesma tendência de variação de profundidade, a LBP é inadequada para distingui-las. Entretanto, embora os sinais de diferenças entre as duas pontas do nariz e seus vizinhos sejam os mesmos, os valores exatos das diferenças são diferentes.

Para solucionar este problema, Huang et al. (2006) desenvolveu uma extensão para este processo, chamada de Padrão Binário Local 3D (3DLBP), capaz de codificar também as diferenças de profundidade (DD).

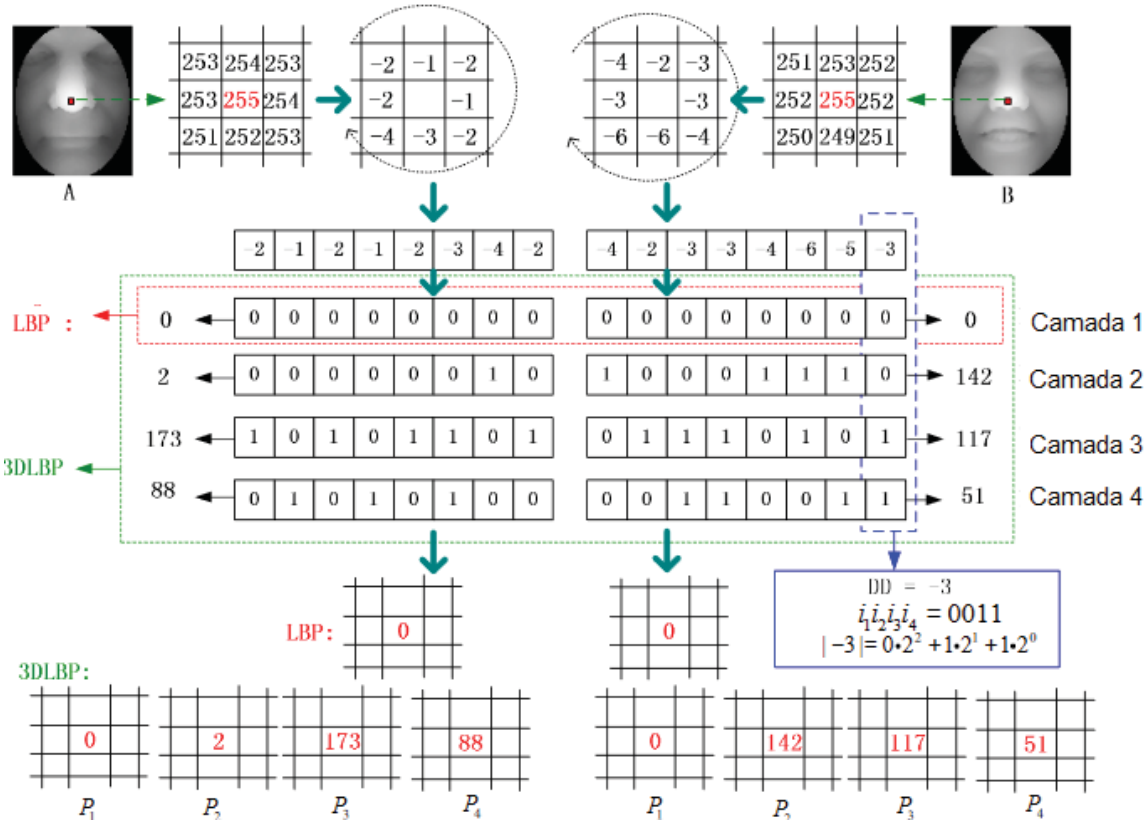
Estatisticamente, os autores observam que 93% das DD entre pontos em R=2 são menores que 7. Isso se deve à suavidade nas transições de profundidade de faces humanas. Assim, foi utilizado apenas três bits para representar a DD. Portanto, junto com um bit que representa o sinal da DD, para cada pixel vizinho ao ponto central, existem, agora, quatro bits representando a variação (i_1, i_2, i_3, i_4), onde i_2, i_3, i_4 representam o valor absoluto da DD e i_1 representa o sinal (codificado por meio do LBP original), como representado nas equações:

$$i_1(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{caso contrário} \end{cases} \quad (5.3)$$

$$|DD| = i_2 * 2^2 + i_3 * 2^1 + i_4 * 2^0 \quad (5.4)$$

Os quatro bits são então separados em quatro camadas. Dessa forma, para cada camada, os bits correspondentes de todo a DD dos pixels adjacentes são concatenados e geram um código LBP. Para correspondência, o histograma de cada código LBP é calculado, então os quatro histogramas são concatenados para formar um descritor único para a imagem.

FIGURA 4.9 – Exemplo comparativo entre LBP e 3DLBP.



FONTE: Adaptado de (HUANG et al., 2006)

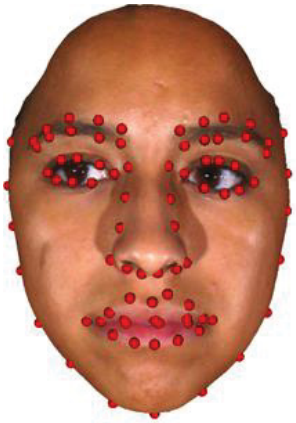
4.2.2 Pontos Fiduciais Faciais

Os Pontos Fiduciais Faciais (do inglês, *Facial Keypoints*, PFF) são pontos de referência que podem ser identificados com alta repetibilidade em uma mesma superfície, mesmo na presença de ruídos e deformações (MIAN et al., 2008). São aqueles localizados em posições específicas da face, que garantem sua existência na maioria das observações. Esses pontos normalmente marcam características salientes como: cantos e centros dos olhos, bordas da boca,

centro do nariz e sobrancelhas, não se limitando a estes somente, como apresentados na FIGURA 4.10.

As técnicas de detecção local de pontos fiduciais buscam os pontos de referência de forma individual, sem fazer uso de informação de outras regiões. Cada região tem seus pontos identificados por detectores distintos, treinados especificamente para tal região, isto é, há detectores para a boca (cantos, lábios superior e inferior), olhos (cantos, pupila), narizes (cantos, ponta, base), entre outros (HOLLANDA, 2011). Exemplos desse tipo de técnica são o Detector por Produto Interno, que utiliza filtro de correlação como métrica de similaridade no projeto de seus classificadores, e detectores baseados em filtros de Gabor, que empregam cascatas de classificadores rápidos (como *AdaBoost*) para selecionar características extraídas por *wavelets* de Gabor (VUKADINOVIC; PANTIC, 2005).

FIGURA 4.10 – Pontos fiduciais extraídos da base BU-3DFE: (a) 83 pontos de referência evidenciados em uma face 3D texturizada; (b) tabela que apresenta o número de pontos identificados para diferentes regiões da face.



(a)

Posição	# pontos
Olho direito	8
Olho esquerdo	8
Sobrancelha direita	10
Sobrancelha esquerda	10
Nariz	12
Boca	20
Borda da face	15

(b)

FONTE: Adaptado de (BERRETTI et al., 2011)

A detecção dos pontos chaves recebe como entrada uma nuvem de pontos que representam a face, os quais são amostrados em intervalos uniformes. A variação, rotação e deformação da malha representada pelos pontos extraídos podem ser empregadas para representar as expressões faciais (MIAN et al., 2008; BERRETTI et al., 2011).

É possível categorizar a expressão facial por meio dos movimentos realizados pelos músculos faciais de uma pessoa. Partindo deste principio, Ekman (2002) desenvolveu um modelo que é capaz de dividir as expressões

faciais em UAs. A fim de oferecer uma descrição abrangente do movimento muscular visível no rosto, Ekman propôs o Sistema de Codificação de Ação Facial (FACS). No sistema, uma expressão facial é uma descrição de alto nível de movimentos faciais representados por regiões ou pontos de referência chamados unidades de ação. Cada UA possui alguma base muscular relacionada e uma determinada expressão facial pode ser descrita por uma combinação de UAs.

A partir da extração dos PFFs, é possível calcular a deformação nos pontos de interesse das características faciais partindo do deslocamento das coordenadas destes pontos. Com as coordenadas obtidas, é possível comparar as deformações geométricas da imagem analisada com uma imagem neutra (imagem sem expressão). Ao fim da análise geométrica da imagem, pode-se deduzir a emoção a partir das combinações de UAs encontradas na face.

Embora muitas das abordagens de reconhecimento facial e de expressões faciais façam uso de técnicas de extração de pontos fiduciais, tais métodos encontram dificuldades em lidar com ambientes sem restrições, os quais apresentam oclusões parciais da face, variações de escala, problemas de iluminação, resolução e diferentes orientações (SILVA, 2017). O uso de informações de terceira dimensão se mostra útil para superar tais dificuldades, provendo a possibilidade de realizar prévio alinhamento da face.

5 METODOLOGIA

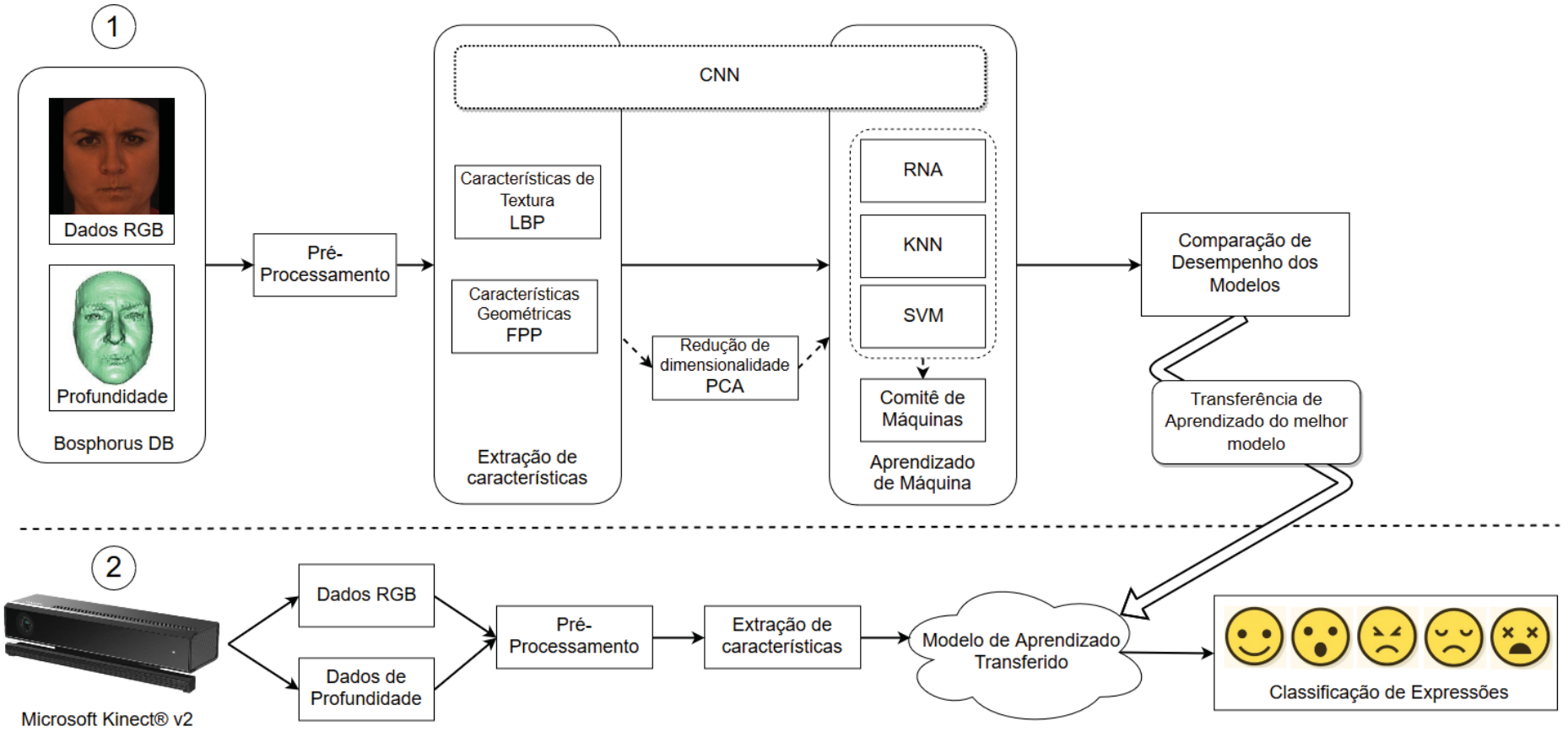
O sistema de reconhecimento de expressões faciais proposto utiliza a informação de profundidade adquirida com o sensor RGB-D Microsoft Kinect V2. Este foi desenvolvimento em dois estágios, conforme apresentado no fluxograma da FIGURA 5.1: (1) treinamento dos modelos computacionais de classificação sobre imagens de alta resolução e (2) aplicação do melhor modelo gerado em imagens adquiridas pelo sensor Kinect.

O processo de geração dos modelos de aprendizado e análise comparativa de seu desempenho, fez se um estudo de caso sobre uma base de dados previamente construída, composta de imagens de um grupo de 6 expressões faciais, e a expressão neutra, capturada por meio de um sensor tridimensional de alta resolução. A partir destes dados, quatro etapas principais foram executadas: (i) pré-processamento dos dados, (ii) extração de características representativas das expressões faciais a partir dos dados bidimensionais e tridimensionais, pela utilização de técnicas de extração por características de textura e geométrica, bem como aplicação de técnicas de redução de dimensionalidade, (iii) geração do modelos de aprendizado, empregando os conjuntos de características extraídas, por meio de duas técnicas de SVM, KNN e CNN, (iv) avaliação do desempenho dos modelos na classificação de novas imagens de expressão facial, pelas métricas de precisão de classificação (matriz de confusão) e tempo de treinamento e classificação dos modelos.

A segunda etapa deste trabalho segue com a adaptação do conhecimento obtido por meio do melhor modelo gerado na primeira etapa, para aplicação em dados, adquiridos e processados por meio do sensor Kinect.

Para efeitos comparativos, todas as análises de desempenho foram realizadas em um computador rodando o sistema operacional Windows 10 – 64 bits, equipado com uma CPU Intel Core i7-4700HQ 2.40 GHz, 16 GB de memória RAM DDR3 e placa de vídeo Nvidia GTX860M 2GB, implementados sobre o ambiente computacional MATLAB R2018a. A tecnologia Intel® Turbo Boost do processador, que possibilita o aumento da frequência base do mesmo até certo limite de temperatura, foi desativada antes do início dos testes, de forma a gerar medidas de tempo de processamento mais estáveis.

FIGURA 5.1 – Fluxograma do sistema desenvolvido.



FONTE: O autor (2018).

5.1 BASE DE DADOS

A popularização das câmeras de captura em três dimensões possibilitou um aumento na quantidade de pesquisas relacionadas ao reconhecimento de expressões faciais por meio de tais dados nos últimos anos. Juntamente com esta tendência, observou-se uma ampliação no número de bases de dados que disponibilizam tais dados. Uma comparação entre a base utilizada (BosphorusDB) e algumas bases públicas disponíveis é apresentada na TABELA 5.1.

TABELA 5.1 – Comparação entre diferentes bases de dados faciais tridimensionais.

Base	Indivíduos	Amostras	Total	Emoções	Poses
Bosphorus DB (SAVRAN et al., 2008)	105	31-54	4652	6	13 + 4 oclusões
FRGC v2	466	1-22	4007	6	N/A
BU-3DFE	100	25	2500	6, em 4 níveis	N/A
ND2006	888	1-63	13450	5	N/A
York	350	15	5250	5	N/A
CASIA	123	15	1845	5	N/A
GavabDB	61	9	549	5	N/A

FONTE: Adaptado de (SAVRAN et al., 2008).

5.1.1 Bosphorus Database

A base de dados Bosphorus DB (SAVRAN et al., 2008), escolhida para o treinamento dos algoritmos de AM neste trabalho, consiste em capturas faciais tridimensionais de 105 indivíduos em várias poses, expressões e condições de oclusão facial. Os indivíduos participantes possuem idade compreendida entre 25 e 35 anos, sendo 60 homens e 45 mulheres, totalizando 4652 varreduras faciais.

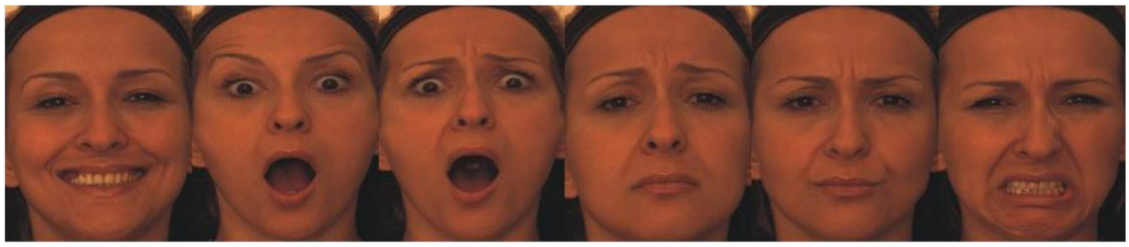
A aquisição dos dados foi realizada com o sensor tridimensional Inspeck Mega Capturor II 3D, um sensor comercial baseado em luz estruturada. O sensor 3D possui sensibilidade de aproximadamente 0.3mm em cada uma das dimensões, gerando aproximadamente 35 mil pontos de coordenadas espaciais por varredura. As imagens de textura foram obtidas em alta resolução (1600 × 1200) sob condições perfeitas de iluminação.

A base também fornece as posições de diversos PFFs manualmente marcados em ambas as imagens 2D e 3D. Para cada varredura de rosto, 24 pontos são marcados nas imagens de textura, desde que sejam visíveis na varredura fornecida, tais como a ponta do nariz, os cantos dos olhos e da boca, entre outros.

Quanto às expressões faciais, a base possui duas categorias distintas: expressões simuladas, baseadas em UAs do FACS, e expressões livres, como encontradas tipicamente na vida real. Para o primeiro tipo, um subgrupo de unidades de ação é simulado, agrupadas em três conjuntos: i) 20 UAs da parte inferior da face, ii) cinco UAs da parte superior da face e iii) três combinações de UAs. As UAs representam blocos de construção de expressões faciais e, portanto, podem constituir uma base flexível para identificação destas.

No segundo conjunto, expressões faciais correspondentes a certas expressões emocionais foram coletadas: felicidade, surpresa, medo, tristeza, raiva e desgosto (FIGURA 5.2). Para alcançar expressões mais naturais, atores e atrizes profissionais foram incorporados entre os indivíduos.

FIGURA 5.2 – Expressões faciais, da esquerda para direita: felicidade, surpresa, medo, tristeza, raiva e desgosto.



FONTE: O autor (2018).

5.2 PRÉ-PROCESSAMENTO

O primeiro passo para desenvolvimento do sistema foi a análise e seleção dos dados fornecidos pela base Bosphorus DB. Como a base fornece varreduras faciais anguladas ou contendo regiões de oclusão, que não apresentam expressões faciais, estes dados foram removidos.

Utilizando somente os dados do conjunto de expressões faciais naturais, previamente identificadas durante o desenvolvimento da base, selecionou-se 752 varreduras de 4666 disponíveis para criação do padrão-ouro a ser empregado para treinamento dos modelos de AM, este conjunto foi chamado de “*conjunto reduzido”*. Estas varreduras representam menos de 20% dos dados disponibilizados pela base. A pequena quantidade de dados em cada classe de expressão facial pode representar um conjunto específico para o treinamento de um modelo de aprendizado genérico o suficiente para ser adaptado para outro domínio de aplicação. Além disso, a quantidade de exemplos para cada classe de expressão facial não estava igualmente

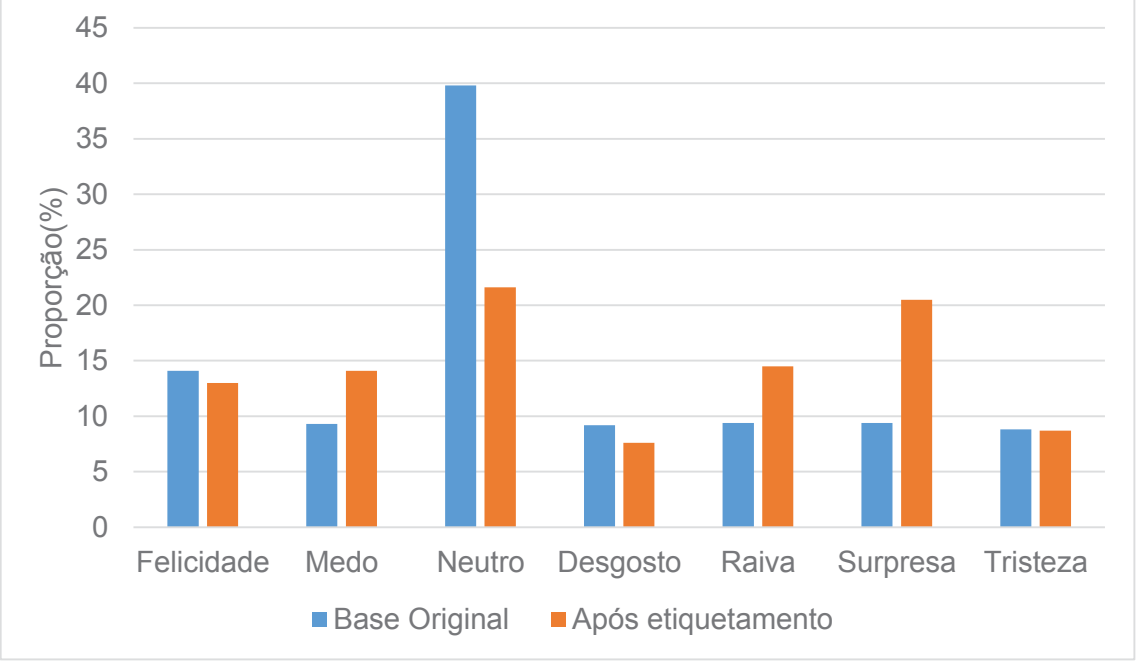
distribuída, como visto na TABELA 5.2. Decidiu-se, portanto, também fazer uso das varreduras onde são emuladas somente unidades de ação facial. Como as UAs, isoladamente, não necessariamente representam uma emoção facial, foi necessário filtrar somente as varreduras que continham uma ou mais UAs representativas das expressões avaliadas neste trabalho.

TABELA 5.2 – Composição dos dados antes e depois do processo de etiquetamento manual.

Expressão facial	**Base original**		**Após etiquetamento manual**	
	Imagens rotuladas	Proporção (%)	Imagens rotuladas	Proporção (%)
Felicidade	106	14,1	286	13,0
Medo	70	9,3	311	14,1
Neutro	299	39,8	474	21,6
Desgosto	69	9,2	168	7,6
Raiva	71	9,4	318	14,5
Surpresa	71	9,4	450	20,5
Tristeza	66	8,8	192	8,7
Total	752	100	2199	100

FONTE: O autor (2018).

FIGURA 5.3 – Distribuição de exemplos para cada classe de expressão facial, antes e depois do pré-processamento da base de dados.



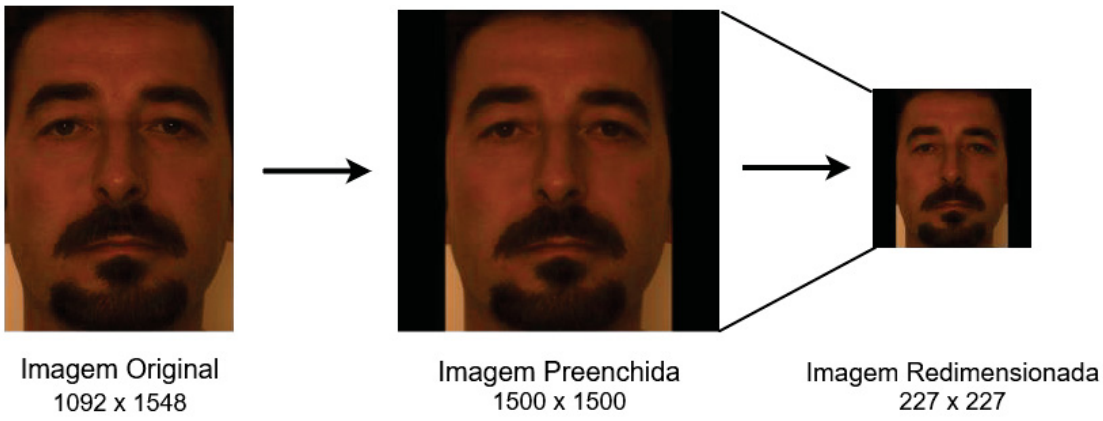
FONTE: O autor (2018).

Realizado este filtro, foi necessário realizar o etiquetamento manual da expressão existente nestas varreduras, visto que somente era identifica a/as UAs emuladas no momento da captura. A identificação da emoção representada pela UA ou conjunto de UAs da varredura, foi realizada seguindo como guia os critérios desenvolvidos por Ekman (1982). Desta forma, foi possível utilizar 47% dos dados disponibilizados pela base, como visto na TABELA 5.2, aumentando a diversidade do conjunto de dados selecionado e melhorando a distribuição dos exemplos entre as classes, como apresentado na FIGURA 5.3.

Como os dados fornecidos pela base já haviam sido previamente recortados para a região de interesse, não foi necessário implementar nenhum algoritmo para rastreio facial e corte. Entretanto, tanto as imagens quanto as matrizes de dados de profundidade apresentavam dimensões diferentes entre varreduras, devido ao processo realizado para corte durante a construção da base. As imagens 2D, por exemplo, apresentavam de 900 a 1400 *pixels* de largura e de 1200 a 1900 *pixels* de altura. Dados com dimensões diferentes dificultam a aplicação em diversas técnicas de AM, principalmente para CNNs.

Como as proporções da grande maioria das imagens eram diferentes, não foi possível simplesmente realizar o redimensionamento dos dados para uma resolução comum, sem introduzir uma distorção severa nos dados. Portanto, realizou-se um redimensionamento proporcional de todas as imagens para 1200 *pixels* de altura, produzindo imagens que ainda possuíam largura variável. Na sequência foi realizado o preenchimento das laterais das imagens com bordas pretas, de forma a produzir imagens quadradas com 1200x1200 *pixels*. A introdução de bordas pretas nos entornos da imagem não afeta a extração de características. Finalmente, de forma a reduzir o tempo computacional necessário para treinamento dos algoritmos de AM, foi produzido um novo conjunto de imagens de menor resolução, como 227x227 *pixels*. Este valor foi escolhido pelo fato de ser o valor padrão empregado pela rede AlexNet (WU et al., 2017), uma das principais redes neurais convolucionais treinadas para reconhecimento de objetos disponível na literatura. A visualização do processamento realizado é apresentada na FIGURA 5.4.

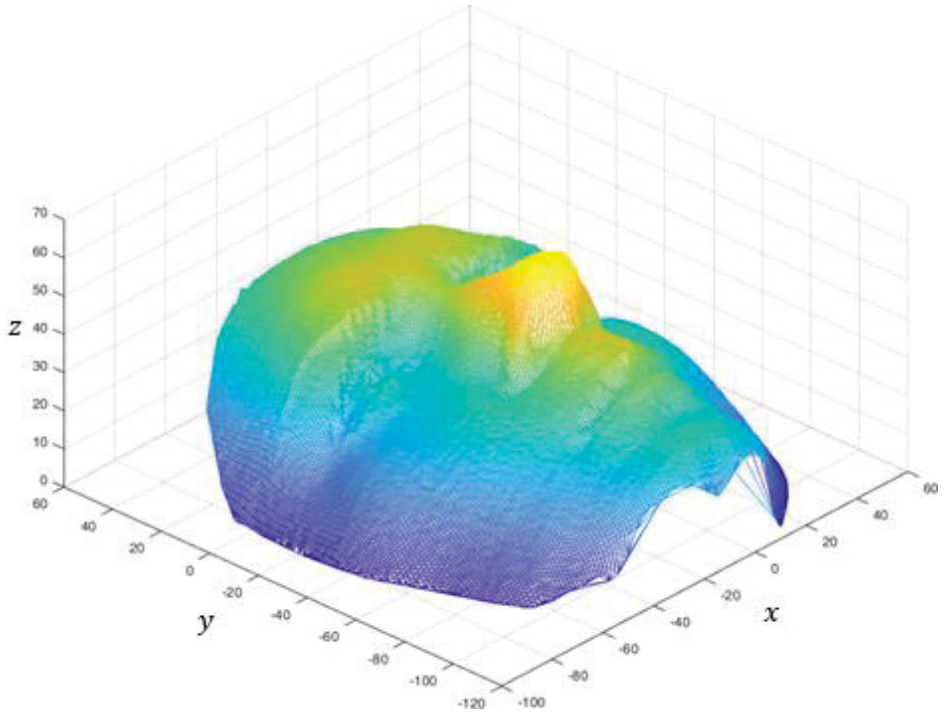
FIGURA 5.4 – Processo de pré-processamento das imagens 2D da base Bosphorus DB.



FONTE: O autor (2018).

No caso dos dados de profundidade, os dados foram disponibilizados pela base de dados no formato de uma nuvem de pontos no espaço tridimensional, da forma como foram capturadas pelo sensor, em um arquivo ".bnt". A FIGURA 5.5 apresenta um exemplo de dados capturados, onde x reprenta a largura, y, a altura e z, a profundidade. Para obter uma representação de mais fácil visualização, os dados foram convertidos para um vetor de dados de profundidade, permitindo a sintetização de uma "imagem topográfica" da face, onde as regiões mais claras representam pontos mais próximos do sensor e regiões mais escuras, pontos mais distantes.

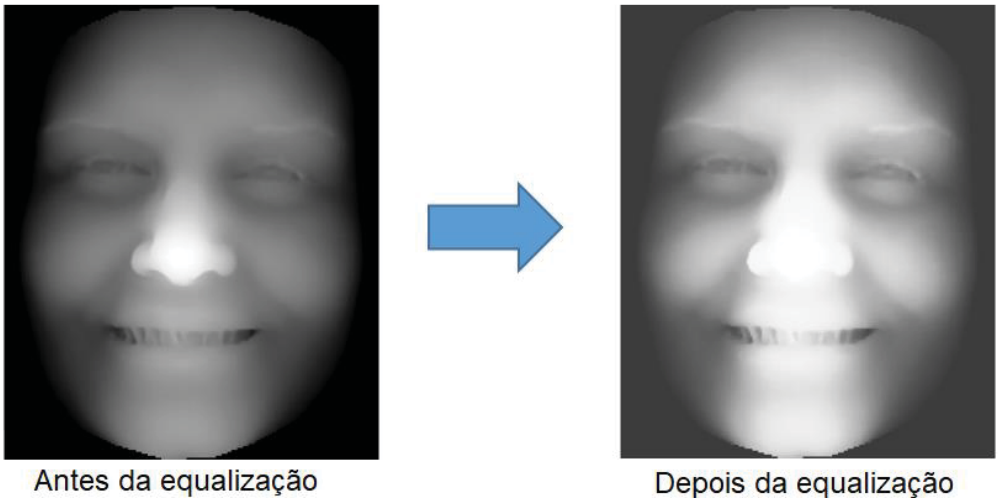
FIGURA 5.5 – Dados de profundidade extraídos da base de dados.



Fonte: O autor (2019).

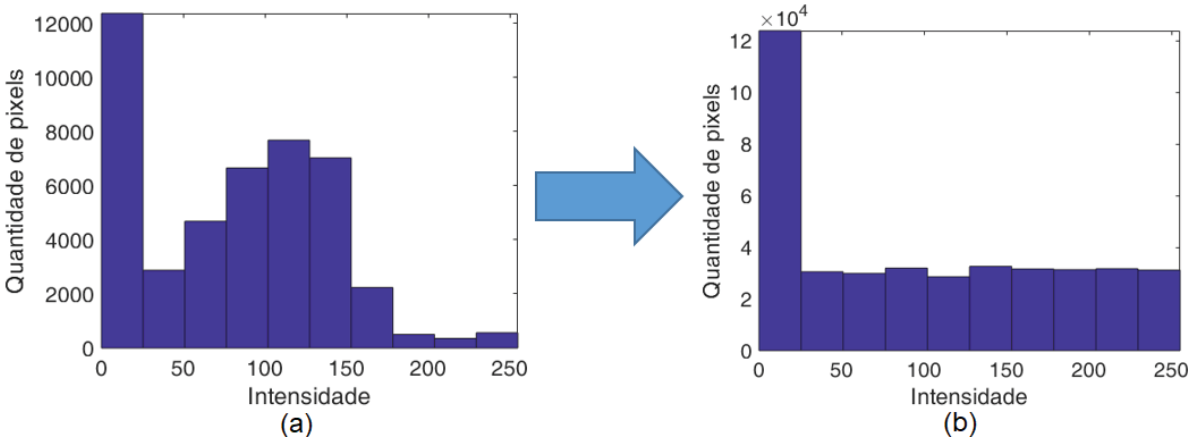
O plano xy de referência, a partir do qual as profundidades são calculadas, é o plano definido pelo valor z mínimo da nuvem de pontos 3D e o plano definido pelos dois eixos principais da PCA. Como as imagens da base de dados foram capturadas em um ambiente controlado, a posição facial dos indivíduos sofreu muito pouco efeito de rotação e translação entre capturadas, fazendo com que diferentes nuvens de pontos tenham, se não iguais, os planos xy de referência semelhantes. Ainda assim, tentou-se eliminar as pequenas diferenças no plano de referência, a partir de uma equalização adaptativa do histograma da imagem 2D gerada (FIGURA 5.6), como descrito por VRETOS et al. (2011). Esse processo conseguiu ressaltar características relevantes da imagem como visto pelos histogramas das imagens na FIGURA 5.7.

FIGURA 5.6 – Comparação das imagens de profundidade antes e depois da equalização adaptativa de histograma.



FONTE: O autor (2019).

FIGURA 5.7 – Histogramas das imagens (a) antes e (b) depois do processo de equalização.



FONTE: O autor (2019).

Tal como com os dados 2D, as nuvens de pontos capturadas pelo sensor de profundidade não apresentação dimensões constantes de largura e altura (as resoluções variavam de 220x180 até 270x200 *pixels*). Então, realizou-se o mesmo procedimento de preenchimento e redimensionamento dos dados para obter matrizes de mesma dimensão e proporção. Os valores inseridos no preenchimento tridimensional foram baseados na média dos 10 menores valores encontrados na matriz da varredura, os quais devem representar o anteparo de fundo da varredura. Estas imagens também foram redimensionadas para a mesma resolução de 277x277 *pixels*, mantendo o padrão adotado.

Como último passo de pré-processamento, a nuvem de pontos foi tratada de forma a definir como profundidade zero todos os pontos que não apresentavam deslocamento positivo no eixo de profundidade z, indicando que estes não são correspondentes à pontos da face capturada. Em seguida, foi realizada a conversão dos pontos de profundidade para uma escala de cinza de 8 *bits*, assumindo como valor máximo (255) o ponto com maior valor de coordenada z, realizando então o mapeamento dos demais pontos para esta faixa. Os arquivos de imagens gerados foram então salvos no formato ".bmp" para facilitar a importação no algoritmo de extração de características 3DLBP.

5.3 EXTRAÇÃO DE CARACTERÍSTICAS

A fase de extração de características faciais a partir dos dados da base foi realizada em sob três abordagens distintas, extração de características geométricas (por meio da análise da variação das coordenadas dos PFFs), extração de características de textura facial (por LBP) e a extração automática de características via as camadas convolucionais de uma CNN (AlexNet).

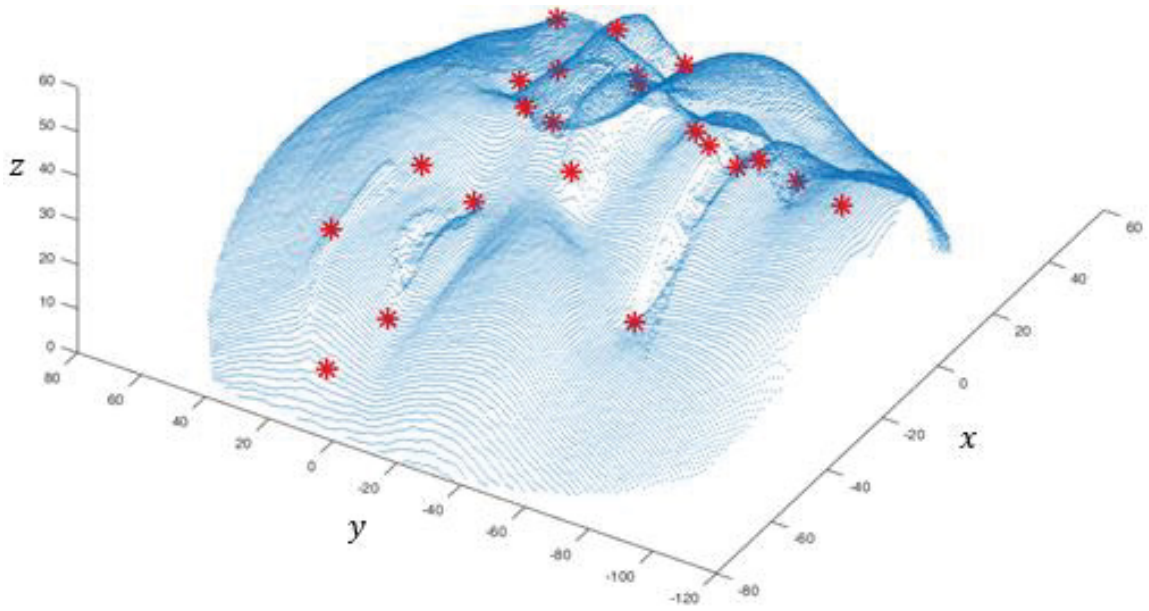
A extração se deu em dois passos. Primeiramente, para se obter uma base do desempenho esperado para a tarefa de classificação de expressões faciais, realizou-se a extração e treinamento dos modelos de aprendizado a partir somente dos dados 2D das imagens RGB fornecidas, possibilitando identificar o nível de melhoria no desempenho gerado pela inclusão da dimensão de dados adicional. Em segunda instância, foram realizadas as adaptações necessárias aos algoritmos de extração de características e de aprendizado para comportarem o processamento da dimensão de dados de profundidade, caracterizando as imagens RGB-D em 3D.

5.3.1 Características Geométricas

A base Bosphorus DB disponibiliza, juntamente com as imagens RGB e a nuvem de pontos de cada captura facial, a localização manual de 24 pontos fiduciais faciais tanto em 2D quanto em 3D. Como pode apresentado na FIGURA 5.8, estes pontos representam posições de características proeminentes da face, tais como posição das sobrancelhas dos cantos dos olhos, cantos da boca, nariz e queixo. Como, para este trabalho, foram excluídos os dados contendo rotação da posição da cabeça (visto que estes não apresentam diferentes expressões faciais), dois dos 24 pontos disponíveis não são utilizados, sendo estes identificadores dos lóbulos das orelhas direta e esquerda.

Os pontos fiduciais foram disponibilizados por meio de arquivos do tipo ".lm2" e ".lm3", respectivamente, contendo o conjunto de coordenadas referentes à etiquetação dos pontos na imagem RGB, para duas dimensões, e dentro na nuvem de pontos, para três dimensões. Dada a estrutura destes arquivos, foi necessária a criação de scripts MATLAB capazes de interpretar os dados e fornecer uma matriz de 22x2 ou 22x3 contendo as coordenadas ordenadamente estruturadas. A base de dados indica que podem existir pontos não marcados entre os dados (no caso de estes não serem visíveis, por exemplo), porém na amostra selecionada dos dados para aplicação neste trabalho este problema não ocorreu.

FIGURA 5.8 – Marcação dos pontos fiduciais faciais sobre a nuvem de pontos tridimensional.

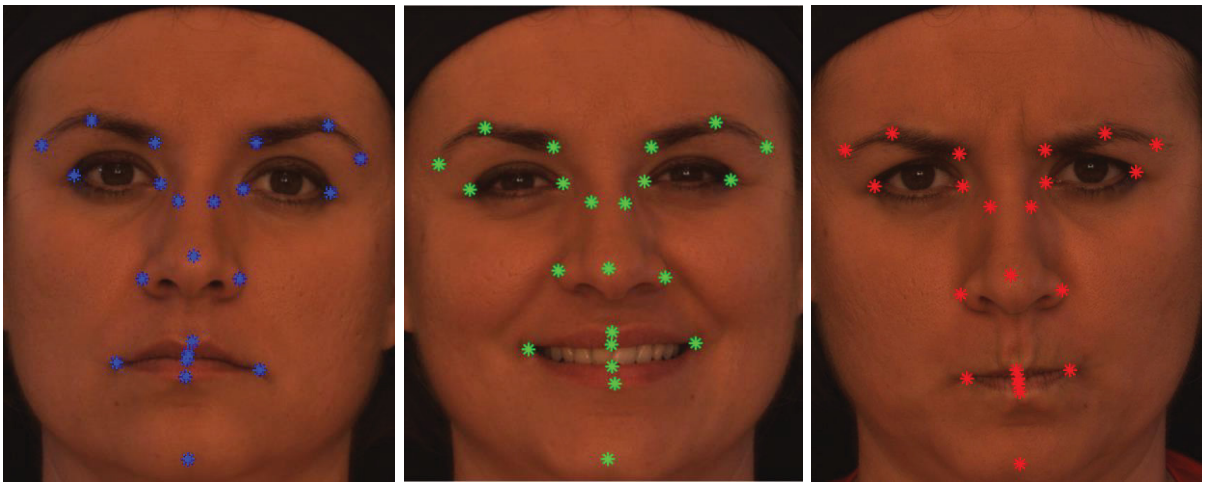


FONTE: O autor (2019).

Para a diferenciação de expressões faciais, é comum utilizar deformações das estruturas faciais como características representativas, tais como movimentação das sobrancelhas, alteração no formato e abertura da boca, abertura dos olhos, entre outros. Desta forma, a alteração no posicionamento dos pontos fiduciais destacados é uma forma mais eficaz de caracterizar uma expressão. Como visto na FIGURA 5.9, a deformação de componentes faciais como a boca e sobrancelhas são visivelmente diferentes em expressões distintas, como felicidade e raiva, por exemplo.

Entretanto, a localização dos pontos faciais por si só não caracteriza uma boa abordagem para classificação de expressões faciais, pois as posições absolutas destes pontos variam amplamente de acordo com o formato facial do indivíduo, pelo posicionamento da face na imagem e até mesmo com leves alterações no ângulo da cabeça. Dessa forma, a posição dos PFFs nas varreduras com expressão neutra foi utilizada para definir uma base de cada PFF para cada um dos indivíduos, a partir da qual a diferença de deslocamento para cada um dos PFFs nas varreduras das demais expressões faciais foi computada. Como existe mais de uma varredura da expressão neutra para cada indivíduo, a média de cada um dos PFFs foi computada para definir a posição base.

FIGURA 5.9 – Comparação entre a posição relativa dos PFFs em uma expressão neutra (à esquerda), de felicidade (centro) e de raiva (à direita).

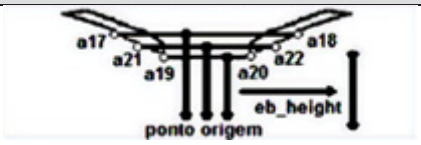
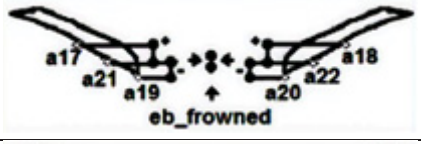
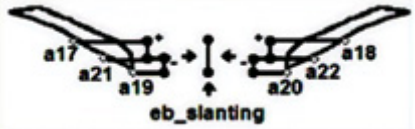
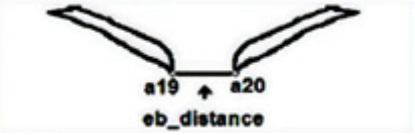
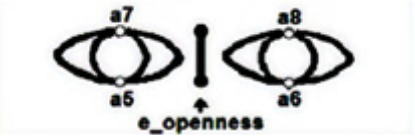
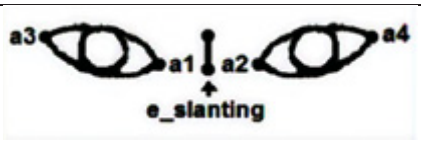
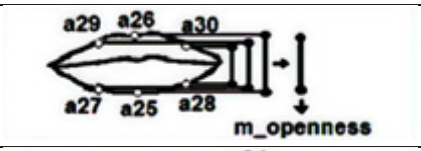
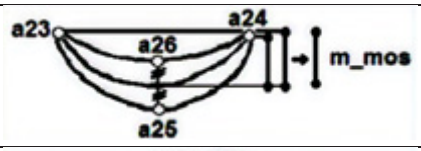
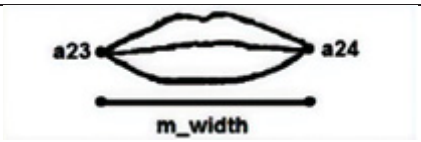
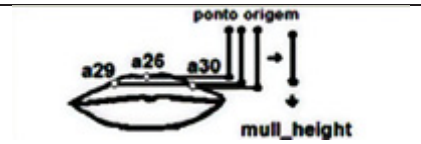
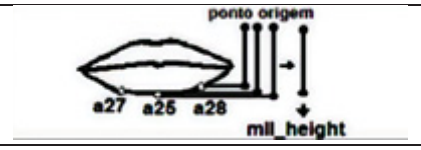


FONTE: O autor (2019).

Além disso, a posição dos PFFs pode ser analisada não só individualmente pela movimentação com relação ao mesmo ponto em uma expressão neutra, mas também na variação da distância relativa entre uma dupla ou grupo de pontos que caracterizam um componente facial específico. Por exemplo, na FIGURA 5.9, por meio da distância dos pontos que representam a posição dos lábios superior e inferior

pode-se inferir se a boca do indivíduo está fechada, aberta ou franzida. Dessa forma, além dos deslocamentos com relação à posição neutra, também forma adicionada ao vetor de características, as distâncias relativas dos pontos que representam deformações relevantes para o reconhecimento das expressões. Na FIGURA 5.10 encontram-se as variáveis de deformação facial utilizadas, bem como o critério empregado para a geração de um valor de intensidade para tal deformação. Quanto maior for o deslocamento do ponto fiducial no eixo indicado, mais este contribui na intensidade da deformação. Com o cálculo destas informações, 12 novos campos de dados foram adicionados ao vetor de características para cada instância da base de treinamento. Haviam quatro instâncias de dados com marcações faltantes. Estas foram completadas com valores nulos (zero).

FIGURA 5.10 – Variáveis da deformação das características faciais.

Deformação	Critério	Exemplo
Altura das sobrancelhas *eb_height*	$\frac{a17.y + a19.y + a21.y}{6} + \frac{a18.y + 20.y + a22.y}{6}$	
Franzimento das sobrancelhas *eb_frowned*	$\frac{a19.y + a17.y + a20.y + a18.y}{2} - a21.y - a22.y$	
Inclinação das sobrancelhas *eb_slanting*	$\frac{(a17.y - a19,y) + (a18,y - a20.y)}{2}$	
Distância entre sobrancelhas *eb_distance*	$a20.x - a19.x$	
Distância entre sobrancelhas e olhos *eeb_distance*	$\frac{(a17.y - a7.y) + (a8.y - a18.y)}{2}$	
Abertura dos olhos *e_openness*	$\frac{(a7.y - a5.y) + (a8.y - a6.y)}{2}$	
Inclinação dos olhos *e_slanting*	$\frac{(a3.y - a1.y) + (a4.y - a2.y)}{2}$	
Abertura da boca *m_openness*	$\frac{a26.y - a25.y}{2} + \frac{a29.y - a27.y}{4} + \frac{a30.y - a28.y}{4}$	
Intensidade do sorriso *m_mos*	$\frac{a23.y + a24 - 2 * a26.y}{2} + \frac{a25.y - a26.y}{4} + \frac{a25.y - a26.y}{4}$	
Alargamento da boca *m_width*	$a24.x - a23.x$	
Altura do lábio superior *mul_height*	$\frac{a29.y + a26.y + a30.y}{3}$	
Altura do lábio inferior *mtl_height*	$\frac{a27.y + a25.y + a28.y}{3}$	

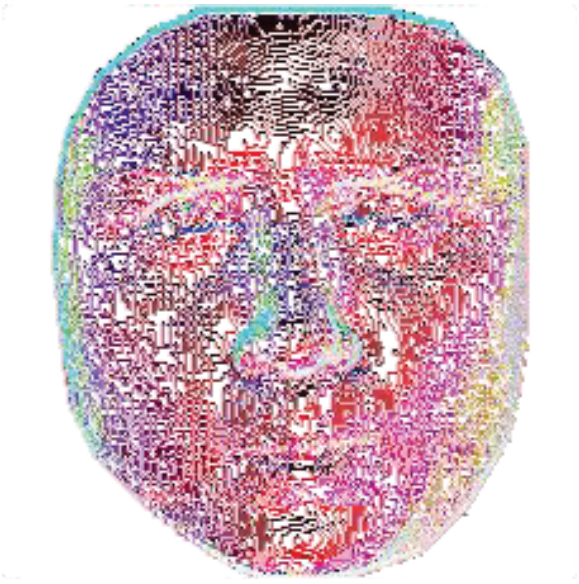
FONTE: Adaptado de (AIRES et al., 2014)

5.3.2 Características de Textura

Apesar da extração dos histogramas de LBP estar disponível diretamente no MATLAB por meio do método *extractLBPFeatures*, esta é limitada à análise de imagens em duas dimensões somente. Portanto, para este caso, utilizou-se a implementação de 3DLPB realizada por CARDIA NETO (2014). Para execução de tal, utilizou-se Python 3.68, além das bibliotecas PCL 1.9.1, Keras 2.24 e Pytorch 1.1.0.

Assim como no caso dos PFFs, o estudo sobre as características de textura foi realizado tanto para os dados em 3D quanto para somente duas dimensões, ignorando os dados de profundidade. Como o 3DLBP é uma extensão do método LBP normal, foi possível executar a extração de características uma única vez sobre a base de dados e obter os vetores de características para ambos os casos de estudo. Como apresentado no item 4.2.1, os histogramas extraídos via 3DLBP possuem quatro níveis, P_1 *a* P_4, sendo que os dados extraídos pela LBP original são os mesmo apresentados no nível P_1. Uma visualização das características extraídas pelo 3DLBP é apresentada na FIGURA 5.11.

FIGURA 5.11 – Visualização das características de textura extraídas pelo 3DLBP.

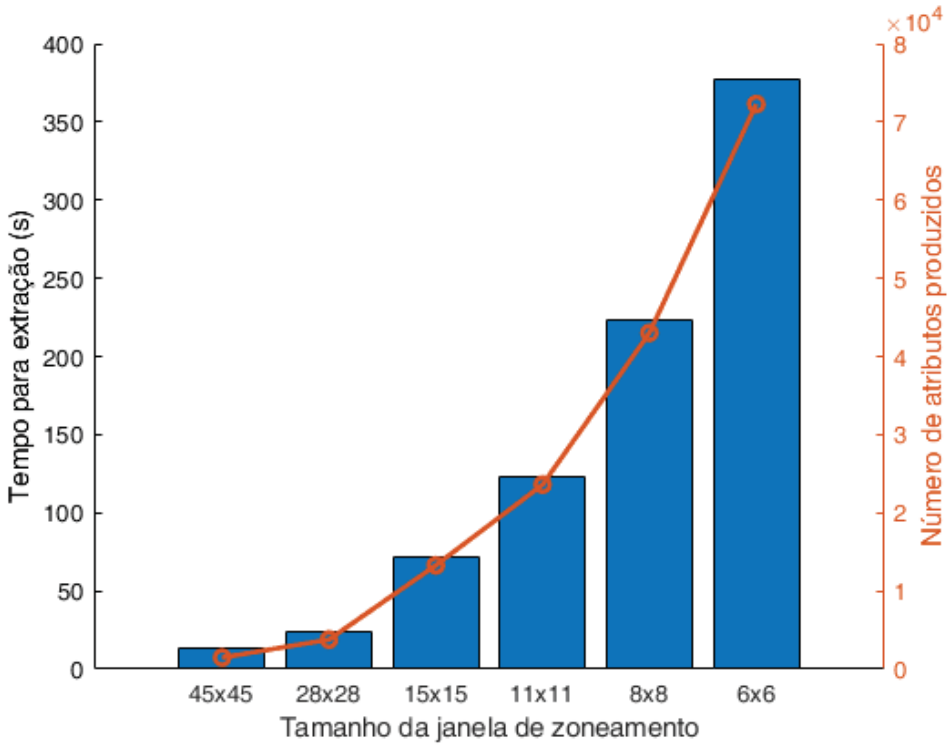


FONTE: O AUTOR (2018).

Como a técnica de extração por LBP é parametrizada com o tamanho da janela de análise, a partir da qual é gerado o histograma de variação do ponto central, buscou-se verificar como diferentes tamanhos de janela afetam o tempo de extração das caraterísticas, bem como a acurácia dos modelos treinados.

Para tal, utilizou-se seis tamanhos diferentes de janelas de zoneamento, 6x6, 8x8, 11x11, 15x15, 28x28 e 45x45 pixels, sob as imagens redimensionadas em 227x277 pixels. No caso de duas dimensões, como é apresentado na FIGURA 5.12, o número de histogramas gerados e, consequentemente, o número de atributos produzidos varia de acordo com a mudança do tamanho da janela. Para as imagens 2D, o número de atributos extraídos variou de 1475, para uma janela de 45x45 pixels, até 72275, para uma janela de 6x6 pixels. O tempo de extração também foi mais extenso para janelas menores, partindo de 12 segundos (45x45) e aumentando exponencialmente até 367 segundos (6x6).

FIGURA 5.12 – Comparação entre o tempo de extração e número de atributos extraídos para diferentes janelas de zoneamento do LPB em duas dimensões.



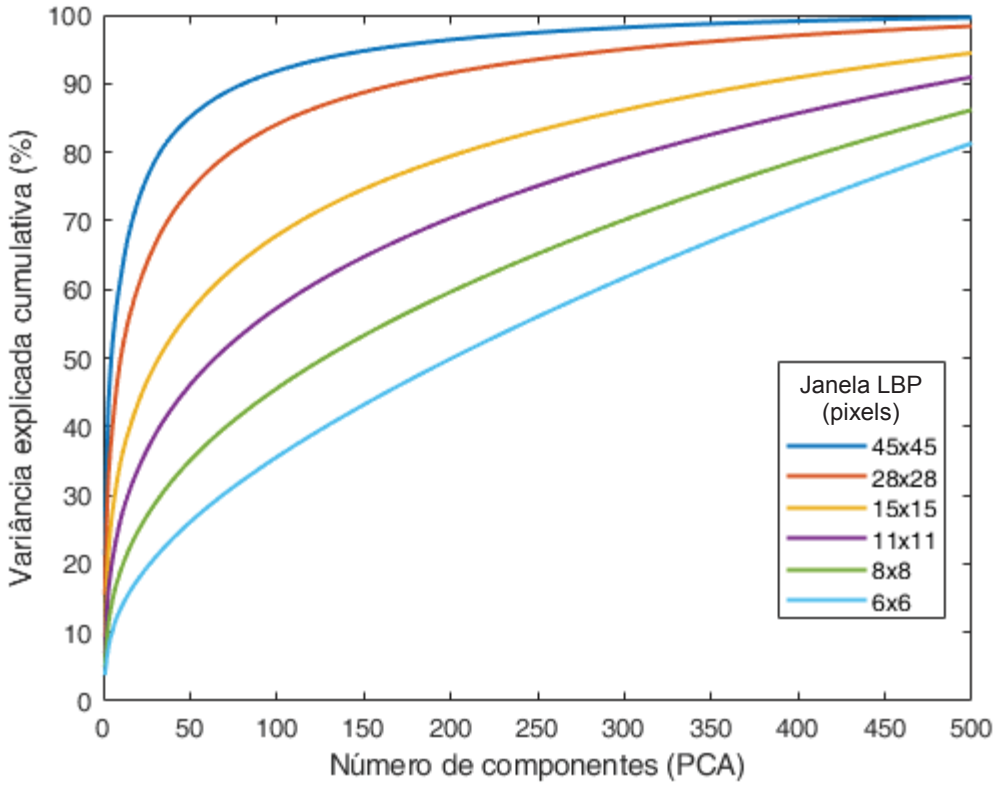
FONTE: O autor (2018).

Dada a esperada alta dimensionalidade dos atributos extraídos por este método, utilizou-se a técnica PCA para realizar a análise e seleção dos componentes que melhor explicam a variância dos dados, assim, minimizando a perda de informação pelo remapeamento dos dados a um espaço de menor dimensão. A FIGURA 5.13 apresenta os valores da variância explicada cumulativa obtida pelo PCA para cada uma das janelas de zoneamento utilizadas. Este demonstra quanta variação do conjunto de dados (e, consequentemente, quanta informação) é capaz de ser representada a medida que mais componentes principais identificados pela

técnica são selecionados. Idealmente, deve-se realizar o balanço entre quanto se admite de informação permita frente a quantas dimensões são reduzidas do conjunto de dados.

Foi possível identificar que com 500 componentes, no mínimo 70% de toda a informação dos dados é representada para todas as janelas avaliadas. Interessante notar que o tamanho da janela tem influência muito grande na quantidade de componentes necessários para representar os atributos gerados. Isto se deve pelo fato de uma janela maior ser capaz de capturar apenas características mais gerais da imagem, enquanto janelas menores permitem maior detalhamento na análise de regiões localizadas. Por exemplo, para a janela de 45x45 *pixels*, a seleção de somente 100 componentes principais dos dados é capaz de codificar em torno de 90% de toda a informação do conjunto de dados. Já para uma janela de 11x11 *pixels*, são necessários 500 componentes para que a mesma quantidade relativa de informação seja representada.

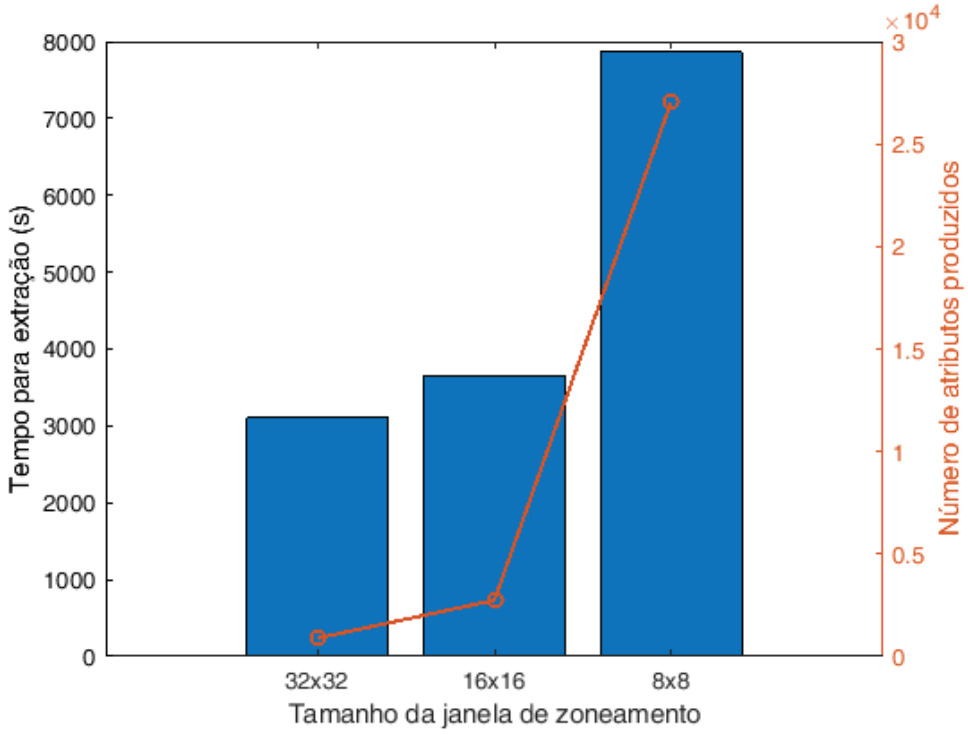
FIGURA 5.13 – Variância explicada por componentes do PCA aplicado em atributos de diferentes tamanhos de janela de zoneamento do LBP em duas dimensões.



FONTE: O autor (2018).

Para os dados em três dimensões, devido a maior complexidade de processamento e informação envolvida, tanto o tempo de processamento quanto o tamanho do vetor de características gerado foram vastamente maiores. Como apresentado na FIGURA 5.14, o tempo de processamento médio para as janelas de extração foi de 4879,1 segundos. Por tal motivo, menos combinações de tamanhos de janela foram avaliadas. O número de atributos extraídos variou de 871 a 27041. O PCA também foi aplicado nestes vetores de dados de forma a produzir 90% da variância explicada.

FIGURA 5.14 – Comparação entre o tempo de extração e número de atributos extraídos para diferentes janelas de zoneamento do LPB em três dimensões.



FONTE: O autor (2019).

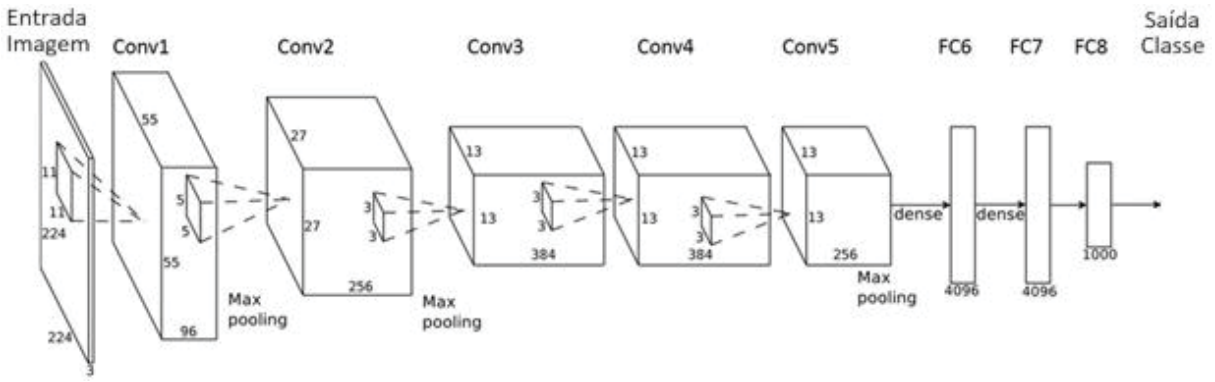
5.3.3 Extração via CNN

Para a extração de características automaticamente via as camadas convolucionais de uma CNN, utilizou-se a arquitetura da rede AlexNet (KRIZHEVSKY; HINTON, 2012). A estrutura da rede, apresentada na FIGURA 5.15, contém cinco camadas convolucionais (Conv 1- 5), seguidas de três camadas totalmente conectadas (FC 6 – 8). Devido à complexidade da tarefa de definição do número e tipos de camadas, bem como o ajuste dos parâmetros pertinentes para cada uma delas, optou-se pela aplicação de um modelo já empregado pela comunidade científica, em contrapartida de uma abordagem própria. Acredita-se que isso pode

levar a melhores resultados, uma vez que as redes pré-treinadas são mais otimizadas do que poderia ser desenvolvido e possuem maior poder de detecção de características.

A escolha da rede AlexNet se deu pela dificuldade em encontrar um modelo largamente utilizado desenvolvido puramente para detecção facial ou reconhecimento de emoções faciais. A AlexNet foi desenvolvida buscando realizar a detecção e classificação de objetos dentre um conjunto de 1000 classes, tais como animais, objetos domésticos e veículos. Mesmo que a rede não tenha sido desenvolvida para lidar diretamente com características faciais, espera-se que os atributos de nível inferior aprendidos por essas redes, tais como bordas e curvas, possam ser transferidos e aplicados no conjunto de dados estudado. Além disso, a rede possui tamanho pequeno o suficiente para ser alocada na memória da GPU disponível.

FIGURA 5.15 – Estrutura da abordagem de CNN AlexNet.



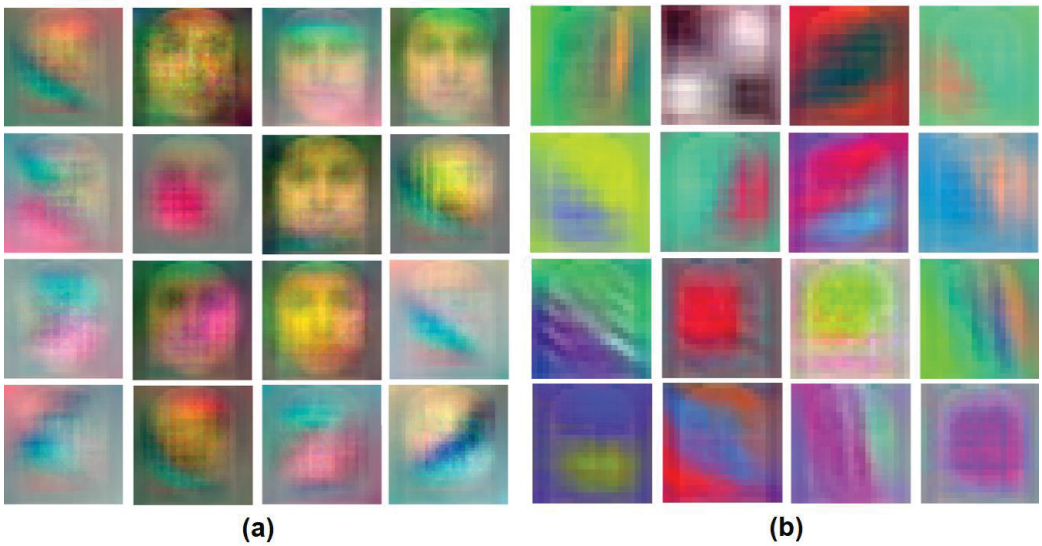
FONTE: Adaptado de (KRIZHEVSKY; HINTON, 2012).

Como visto na FIGURA 5.15, a estrutura utilizada é formada por cinco camadas convolucionais com os tamanhos das janelas de convolução variando entre 11x11 pixels e 3x3 pixels. As camadas Conv1, Conv2 e Conv5 possuem uma camada do tipo *Max Pooling* na saída, para gerar agrupamento dos atributos. Exemplos de características extraídas pelas camadas convolucionais podem ser observados na FIGURA 5.16. Percebe-se que, à medida em que os dados avançam as camadas convolucionais, as características extraídas se tornam mais abstratas e menos humanamente inteligíveis.

Para a extração de características a partir dos dados em 3D, foram implementadas adaptações na estrutura da rede, de forma a inserir uma nova dimensão de filtros convolucionais. As convoluções foram aplicadas sobre os dados de duas dimensões por vez. Como o processo se tornou computacionalmente muito

lento com o uso das imagens 2D em RGB, pois cada canal de cor gerava uma dimensão de camadas convolucionais separada, as imagens em 3D foram convertidas para escala de cinza antes de alimentarem a entrada da rede.

FIGURA 5.16 – Exemplo de características extraídas pelas camadas convolucionais Conv1 (a) e Conv2 (b) da CNN AlexNet para dados 2D.



FONTE: O autor (2018).

Além do treinamento da CNN AlexNet completa, as características abstraídas pelas camadas convolucionais foram utilizadas para treinar um modelo de SVM.

5.4 MODELOS DE CLASSIFICAÇÃO

Os algoritmos de AM aplicados para treinar os modelos de conhecimento capazes de classificar as expressões faciais foram implementados utilizando o ambiente computacional MATLAB R2018a. Estes recebem como entrada as características faciais extraídas descritas anteriormente.

Cinco abordagens distintas foram avaliadas neste trabalho, Máquinas de Vetores de Suporte, K-Vizinhos Mais Próximos, Redes Neurais Artificiais, Comitê de Máquinas e a Rede Neural Convolucional AlexNet. Com exceção da última, modelos distintos foram treinados com as características de texturas e geométricas. No caso da AlexNet, como as camadas convolucionais da CNN eliminam a necessidade de extração de características manualmente, esta foi treinada somente com os próprios atributos produzidos, extraídos das próprias imagens da base de dados.

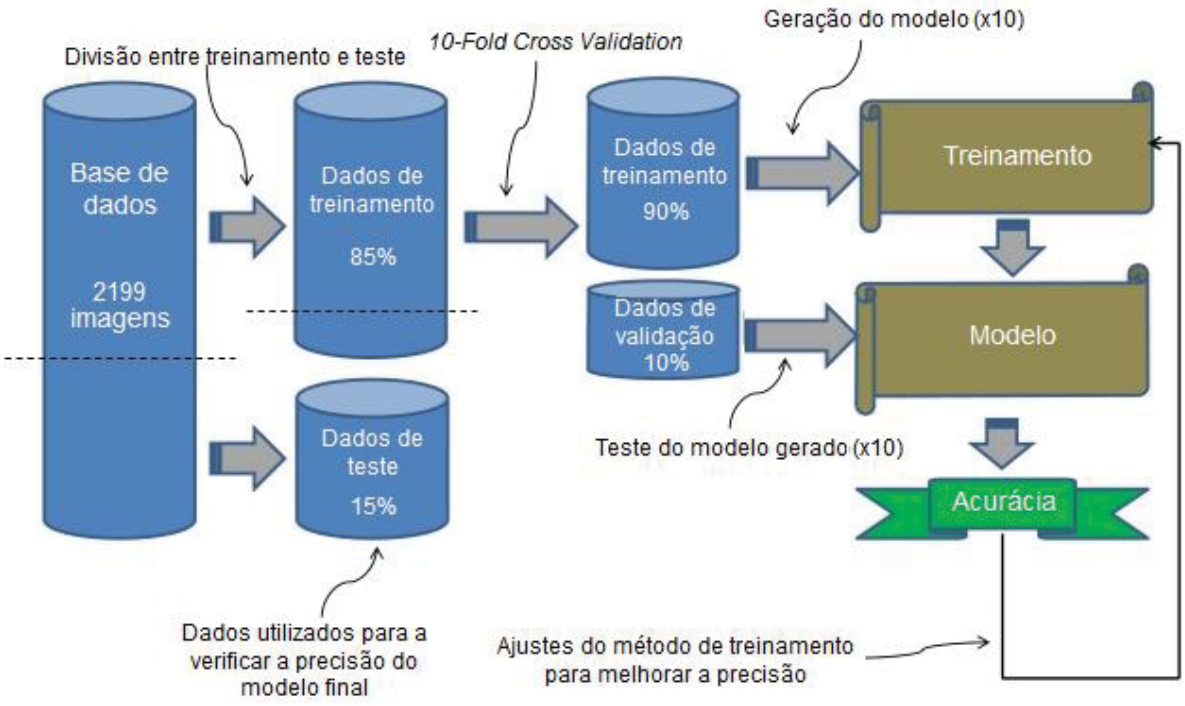
5.4.1 Divisão dos dados

A divisão entre os conjuntos foi realizada da seguinte forma:

- 85% dos dados foram destinados para o conjunto de treinamento e validação;
- 15% dos dados foram destinados para o conjunto de testes.

Como observado na FIGURA 5.17, o conjunto de treinamento é utilizado para a geração dos modelos de aprendizado, porém estes passam previamente por um processo de validação cruzada. Contrariamente, os dados do conjunto de teste nunca são fornecidos aos algoritmos durante o treinamento, sendo empregados somente para a análise final de precisão dos modelos gerados.

FIGURA 5.17 – Processo de divisão do conjunto de dados.



FONTE: O autor (2019)

A divisão dos dados foi realizada de forma estratificada, tentando manter a mesma proporção de instância de dados por classe, como visto no histograma da FIGURA 5.18.

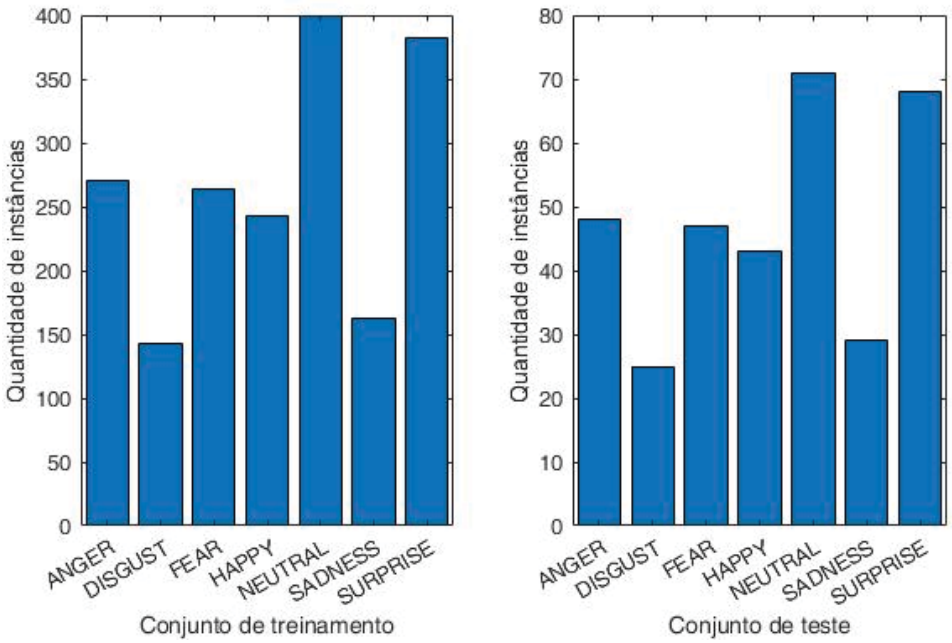
5.4.2 Validação Cruzada

De forma a minimizar a influência da aleatoriedade da divisão do conjunto de dados para treinamento e teste do modelo, a técnica de (do inglês, *Cross Validation*,

CV) foi aplicada a todos os algoritmos. O CV é uma técnica de validação de modelos de AM amplamente utilizada na comunidade científica (BROWNE, 2000). CV foi originalmente empregada para avaliar a capacidade preditiva de equações de regressão linear para prever um critério de pontuações em uma bateria de testes (BOARETTO, 2017).

O *Holdout* é a técnica mais simples de CV e consiste em dividir o conjunto de dados em dois subconjuntos, treinamento e testes, geralmente esta divisão é feita da forma 70%-30%, com a maior parte dos dados destinados para o conjunto de treinamento. Isto permite com que o modelo de aprendizado gerado seja avaliado sem nenhum viés, pois os dados de teste nunca foram “vistos” anteriormente pelo modelo. Devido a aleatoriedade da divisão dos dados, esta técnica pode nem sempre representar o desempenho correto do modelo.

FIGURA 5.18 – Proporção de dados por classe após a divisão para treinamento e teste.



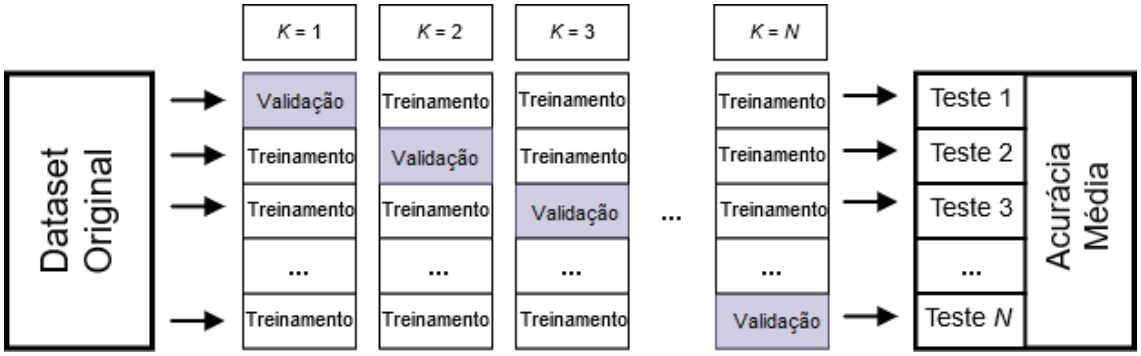
FONTE: O autor (2019)

Uma técnica mais precisa para se medir o desempenho de um modelo de aprendizado é intitulada Validação Cruzada *K-fold,* utilizada neste trabalho. Esta técnica consiste na divisão do conjunto de dados original em K subconjuntos de tamanho igual. Desta forma, o método de treinamento e teste é repetido K vezes, alternando-se o conjunto de *holdout* para teste. Após a obtenção dos K índices de desempenho, a média destes é efetuada para se obter a acurácia média do modelo. Este método faz com que a forma como os dados são divididos não tenha tanta

importância, pois cada ponto de dados é empregado como conjunto de testes exatamente uma vez. Já a desvantagem é a necessidade da execução do algoritmo de treinamento, que pode ser computacionalmente pesado, diversas vezes (GEISSER, 1975).

Neste trabalho utilizou-se a técnica de validação cruzada com dez divisões (*10-fold Cross Validation*). O conjunto de treinamento é divido em dez subgrupos (como visto na FIGURA 5.19), dos quais 9 são aplicados para o treinamento do modelo e o último (não visto) para análise da acurácia. A partir da acurácia média obtida dos dez modelos gerados é possível realizar ajustes nos parâmetros de treinamento e a geração de novos modelos. O tempo de treinamento para cada abordagem de AM foi calculado a partir da média dos 10 treinamentos realizados pelo processo de CV.

FIGURA 5.19 – Validação Cruzada *K-Fold.* Neste trabalho, utilizou-se *K*=10.



FONTE: O autor (2018)

5.4.3 Análise de Desempenho

O principal parâmetro de pontuação adotado como métrica para medir o desempenho dos algoritmos é a acurácia da matriz de confusão. Esta gera um conjunto de 4 variáveis, que permite entender o desempenho da classificação produzida pelo modelo (predito) em comparação com as classes originais (observados), dado um problema de classificação binária com duas classes nomeadas como P (Positivo) e N (Negativo). As variáveis em verde, VP (Verdadeiro Positivo) e VN (Verdadeiro Negativo) representam as classificações corretas dentro das duas classes deste problema exemplo, devendo ser maximizadas. Já as variáveis apresentadas em vermelho, FN (Falso Negativo) e FP (Falso Positivo) representam classificações incorretas.

TABELA 5.3 – Exemplo de matriz de confusão para duas classes.

		Predito	
		P	N
Observado	P	VP	FN
	N	FP	VN

FONTE: O autor (2018).

Este modelo é facilmente escalável para tratar problemas multiclasse, como o reconhecimento de expressões faciais. Para o caso de estudo deste trabalho, sete classes foram utilizadas para classificação (Desgosto, Felicidade, Medo, Neutro, Raiva, Surpresa e Tristeza), gerando a seguinte matriz de confusão:

TABELA 5.4 – Matriz de confusão multiclasse para a classificação de expressões faciais.

		Predito						
		Desgosto	Felicidade	Medo	Neutro	Raiva	Surpresa	Tristeza
Observado	Desgosto							
	Felicidade							
	Medo							
	Neutro							
	Raiva							
	Surpresa							
	Tristeza							

FONTE: O autor (2018).

No modelo da TABELA 5.4 é possível identificar as frequências de classificação para cada classe do modelo. As classes “observadas” se referem às classes reais dos dados, nas quais o modelo deveria classificar cada uma das instâncias do conjunto de dados. Já as classes “preditas” são as que o modelo classificou cada uma das instâncias de dados. Idealmente, espera-se obter valores maiores na diagonal principal da tabela (indicada em verde), a qual representa dados corretamente classificados em suas respectivas classes.

A acurácia de uma matriz de confusão é calculada para cada uma das classes separadamente (como visto na Equação 6.1) e consiste no número de todas as previsões corretas realizas sobre o conjunto de dados de teste para tal classe, divididas pelo número total de instâncias de dados desta classe presente no conjunto

de dados de teste. O valor obtido varia de 0, que representa nenhum acerto, a 1, que representa que todas as instâncias classificadas corretamente.

$$Acurácia = \frac{Dados\ \ corretamente\ classificados}{Total\ de\ dados} \tag{6.1}$$

Da mesma forma, o cálculo da acurácia pode ser realizado para todo o conjunto de dados de teste por meio do somatório de todas as instâncias de dados corretamente classificadas em suas devidas classes dividido pelo total de instâncias do conjunto.

5.4.4 Treinamento dos modelos

Neste trabalhou avaliou-se o desempenho dos modelos de aprendizado produzidos pelos algoritmos SVM, KNN, RNA, CNN e um comitê de máquinas. Todos os algoritmos foram implementados por meio de funções disponibilizadas pelo MATLAB R2018a.

Para todos os algoritmos, os hiperparâmetros foram obtidos por meio de otimização Bayeseana, disponível no MATLAB pela função *bayesopt.* A otimização foi realizada para o conjunto de dados de PFFs e para o conjunto de dados LBP. A execução do processo de otimização em conjunto com a validação cruzada fez com que um tempo expressivamente elevado fosse necessário para a avaliação de cada um dos algoritmos. De forma a possibilitar a comparação do tempo necessário por cada um dos algoritmos para efetivamente realizar o treinamento do modelo, após a identificação dos melhores parâmetros para o conjunto de dados de treinamento, uma nova execução do algoritmo foi realizada, fixando, neste momento, os parâmetros encontrados.

Nos próximos itens são apresentados os parâmetros e adaptações realizadas em cada um dos algoritmos trabalhados.

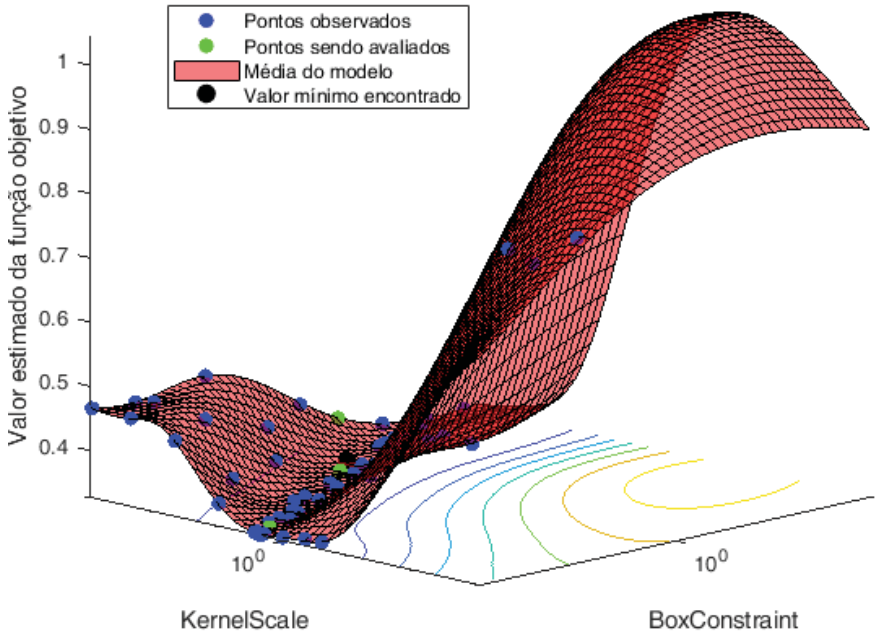
5.4.4.1 Máquinas de Vetores de Suporte

Para o treinamento das SVMs foram empregadas as funções *templateSVM*, para definição dos parâmetros do algoritmo, e *fitcecoc*, para treinamento do modelo.

A otimização para a SVM avaliou os parâmetros *Coding*, *BoxConstraint*, *KernelScale, KernelFunction, PolynomialOrder* e *Standardize*, levando em torno de 3200 segundos para execução. O parâmetro de configuração mais relevantes para a

SVM é a função de *kernel* (*KernelFunction*), que modifica a forma como o mapeamento das funções de separação dos dados é realizada. O MATLAB fornece quatro opções de funções *kernel* implementadas, Gaussiana, *Radial-Basis Function* (RBF), Linear e Polinomial. A função gaussiana apresentou maior nível de precisão nos testes realizados com o conjunto de dados de treinamento, sendo então a escolhida para implementação.

FIGURA 5.20 – Hiperplano de otimização dos parâmetros *KernScale e BoxConstaint* da SVM.



FONTE: O autor (2019).

A FIGURA 5.20 apresenta um dos hiperplanos da função objetivo da SVM otimizados pela função Bayeseana entre os parâmetros *KernelScale e BoxConstraint*. Os demais parâmetros modificados do padrão são apresentados na TABELA 5.5.

TABELA 5.5 – Parâmetros de treinamento modificados para a SVM.

Parâmetro	PFF	LBP
BoxConstraint	76,31	135,31
Coding	onevsall	onevsall
Iteration_Limit	10^6	10^6
KernelFunction	*Gaussian*	*Gaussian*
KernelScale	981,84	1394,34
PolynomialOrder	-	-
Standardize	false	false

FONTE: O autor (2019) .

5.4.4.2 K-Vizinhos Mais Próximos

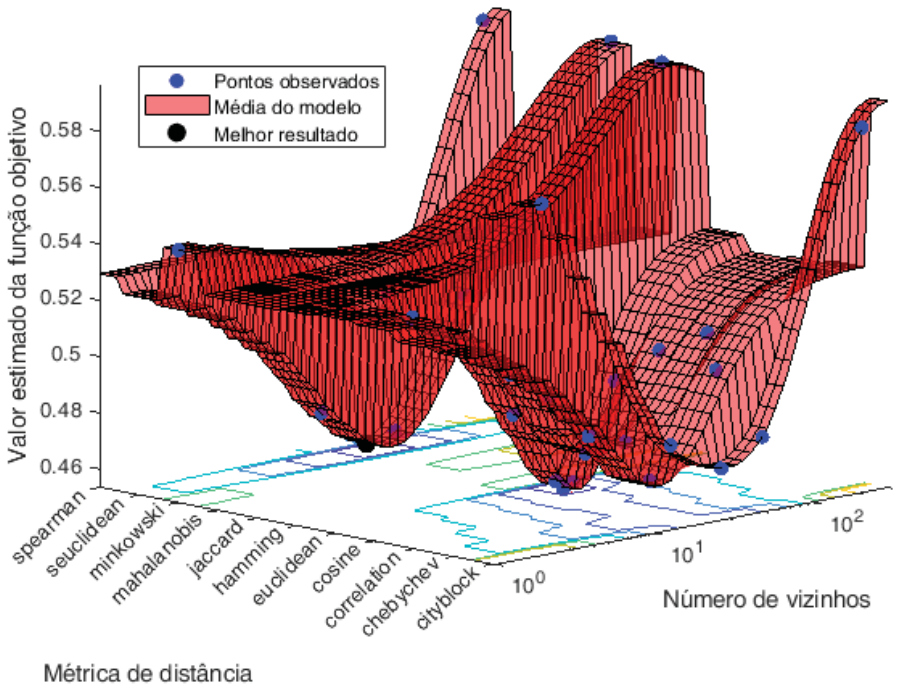
O algoritmo KNN foi treinado por meio da função *fitcknn*. Os testes com conjunto de dados de treinamento foram realizados com o parâmetro *OptimizeHyperparameters* definido para a opção '*all*'. Esta opção permite que o próprio MATLAB realize a otimização de todos os parâmetros de treinamento e identifique os que apresentam melhor desempenho, como visto na FIGURA 5.21. O tempo de otimização médio foi de 126 segundos. Os parâmetros utilizados são apresentados na TABELA 5.6.

TABELA 5.6 – Parâmetros de treinamento modificados para o KNN.

Parâmetro	PFF	LBP
Distance	*cosine*	*seuclidian*
Coding	onevsall	onevsall
DistanceWeight	inverse	inverse
Standardize	false	false
NumNeighbors	13	18

FONTE: O autor (2018).

FIGURA 5.21 – Resultado da otimização dos parâmetros do KNN para LBP.



FONTE: O autor (2018).

5.4.4.3 Redes Neurais Artificiais

Duas abordagens distintas de RNAs, do tipo *feedforwardnet*, foram avaliadas. A primeira é uma rede ligeiramente complexa, com duas camadas ocultas, cada uma com 256 neurônios (NN1). A segunda é uma rede minimizada, com apenas uma camada oculta de 10 neurônios (NN2).

- Duas abordagens de RNA avaliadas:
    - **NN1** – 1 camada oculta com 10 neurônios:
    - **NN2** – 2 camadas ocultas com 256 neurônios cada;

Para ambas as estruturas, os mesmos parâmetros de treinamento foram modificadas do padrão, apresentados na TABELA 5.7.

TABELA 5.7 – Parâmetros de treinamento modificados para as RNAs.

Parâmetro	PFF / LBP
divideParam.trainRatio	*85/100*
divideParam.validationRatio	15/100
performParam.regularization	0,5
useGPU	*yes*

FONTE: O autor (2018).

5.4.4.4 AlexNet

Como a CNN AlexNet foi previamente treinada com objetivos diferentes do reconhecimento de expressões faciais, foi necessário realizar adaptações na estrutura de classificação. A camada final do tipo *softmax* foi alterada para produzir como saída apenas 7 classes ao invés das 1000 usadas originalmente. Para possibilitar a entrada da camada de profundidade das imagens, no caso tridimensional, uma nova camada de entrada de mesmo tamanho (227x227 pixels) foi adicionada à rede.

Além disso, pelo fato da base de dados utilizada ser menor do que a ImageNet, empregada originalmente, foram realizados ajustes nos hiperparâmetros que controlam a taxa de aprendizado, a fim de obter uma taxa mais rápida no início do treinamento, que decai conforme as épocas de treinamento são executadas, permitindo a convergência da rede. O tamanho do *batch* de treinamento foi reduzido devido a limitações de memória na placa gráfica disponível. Os parâmetros modificados são apresentados na TABELA 5.8.

Devido ao extenso tempo de treinamento da rede, neste caso não foi utilizada a validação cruzada. No entanto, optou-se pela ativação do parâmetro *Shuffle,* que

mistura novamente os dados de treinamento e teste a cada nova época de treinamento da rede.

Para a implementação da CNN de forma a ser capaz de trabalhar com os dados de profundidade, as camadas convolucionais da rede foram substituídas por camadas que realizam a convolução em três dimensões. Como o MATLAB R2018a não possui suporte a camadas convolucionais 3D, utilizou-se a *toolbox* mdCNN, disponibilizada por Garty (2019) Por meio desta, foi realizada a reimplementação da rede AlexNet, mantendo os mesmos parâmetros de cada uma das camadas.

TABELA 5.8 – Parâmetros de treinamento modificados para a AlexNet.

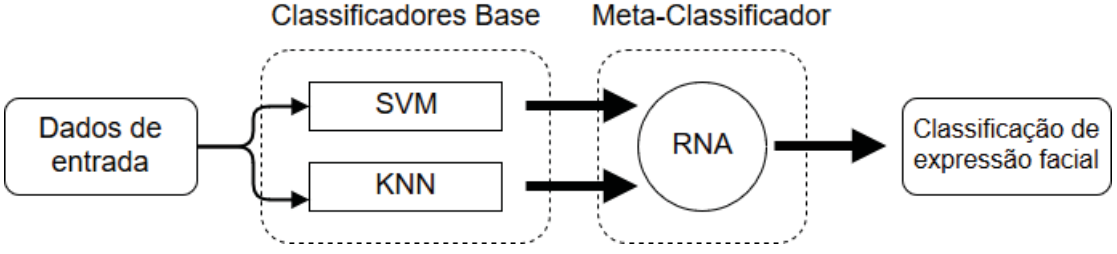
Parâmetro	Valor
InitialLearningRate	*0,0005*
LearnRateDropFactor	0,8
LearnRateDropPeriod	15
MaxEpochs	100
MiniBatchSize	*5*
ValidationPatience	100
Shuffle	*every-epoch*
ValidationPatience	*500*
ValidationFrequency	*20*

FONTE: O autor (2018).

5.4.4.5 Comitê de Máquinas

O modelo de comitê de máquinas utilizado é baseado na metodologia de *stacking*, onde grupos classificadores fracos do tipo SVM e KNN são adotados para gerar um grupo de “votos” para cada classe, os quais são ponderados por uma RNA que é capaz de gerar a classe de saída, como representado na FIGURA 5.22. Os parâmetros de treinamento para os classificadores específicos permaneceram os mesmos apresentados anteriormente, para os 15 classificadores de cada tipo.

FIGURA 5.22 – Estrutura de *Stacking* proposta.

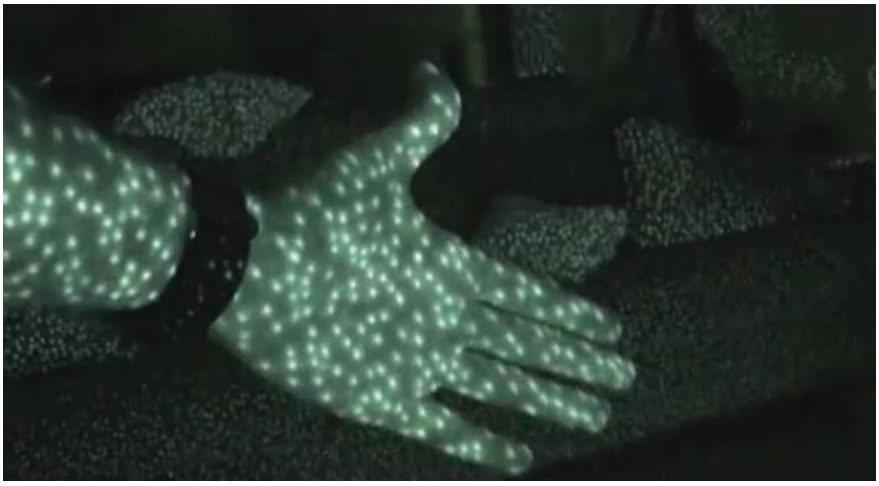


FONTE: O autor (2018).

5.5 AQUISIÇÃO DE IMAGENS TRIDIMENSIONAIS

A aquisição das imagens tridimensionais de baixa resolução para transferência do modelo de conhecimento foi realizada com o uso do sensor Kinect v2. A aquisição de dados de profundidade é realizada por meio de dois componentes, um emissor de infravermelho e uma câmera (sensor). A técnica adotada pelo Kinect v2 para aquisição dos dados de profundidade consiste na emissão de um padrão pontos infravermelhos, projetados sobre as superfícies dentro do seu campo de visão, como demonstrado na FIGURA 5.23. A reflexão destes é, então, capturada por um sensor infravermelho e processada de forma a analisar a diferença de tamanho gerada pela distância do objeto até a câmera. Como o posicionamento do sensor é ligeiramente deslocado em relação ao emissor, as distorções observadas no mapa de pontos são aplicados para calcular a profundidade em cada pixel da câmera RGB (WU, 2017).

FIGURA 5.23 – Mapa de pontos infravermelhos emitidos pelo sensor Kinect para obtenção da imagem de profundidade.



FONTE: https://jahya.net/blog/how-depth-sensor-works-in-5-minutes/

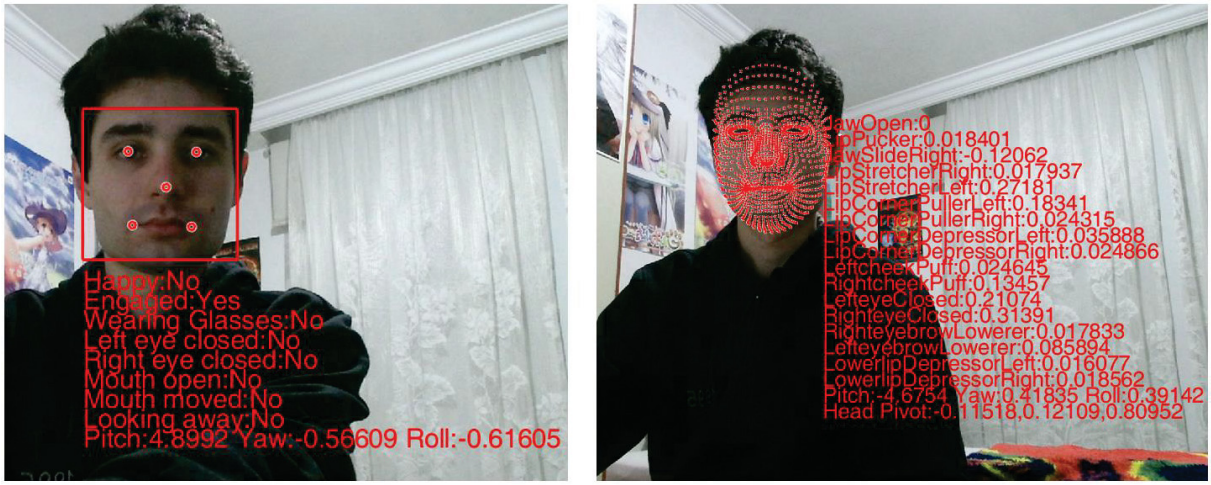
5.5.1 Interface com o Kinect v2

Juntamente com os drivers do sensor Kinect v2, é fornecido o kit de desenvolvimento (Kinect For Windows SDK 2.0) com recursos e APIs para interface com o dispositivo de forma simples por meio de código escrito em C#, C++ e Java, permitindo fácil acesso aos quadros RGB-D capturados pelo sensor, bem como informações de alto nível previamente processadas por algoritmos, tais como o

rastreamento corporal, análise de componentes facial, reconstrução de ambientes em 3D e métodos para calibração da câmera.

A API Kinect Face Tracking (FIGURA 5.24 (a)) contém um mecanismo de rastreamento de face, capaz analisar a entrada da câmera do Kinect e detectar a posição e ângulo da cabeça, além de cinco pontos calculados diretamente pelo sensor em tempo real (olhos, nariz e cantos da boca). A API Kinect HD Face (FIGURA 5.24 (b)) faz uso de toda a informação RGB-D e é capaz de gerar um modelo tridimensional da face no formato de uma nuvem de pontos e também identificando e quantificando a intensidade de características da face, tais como abertura/fechamento dos olhos e boca e detecção de sorriso. Além disso, a API Kinect HD Face é capaz de identificar 87 PFFs na imagem RGB e 13 pontos adicionais na imagem de profundidade, dentre estes, os cantos da boca, o centro de cada olho, o centro do nariz e a caixa delimitadora ao redor da cabeça.

FIGURA 5.24 – Dados extraídos via (a) API Kinect Face Tracking e (b) API Kinect HD.



(a) **(b)**

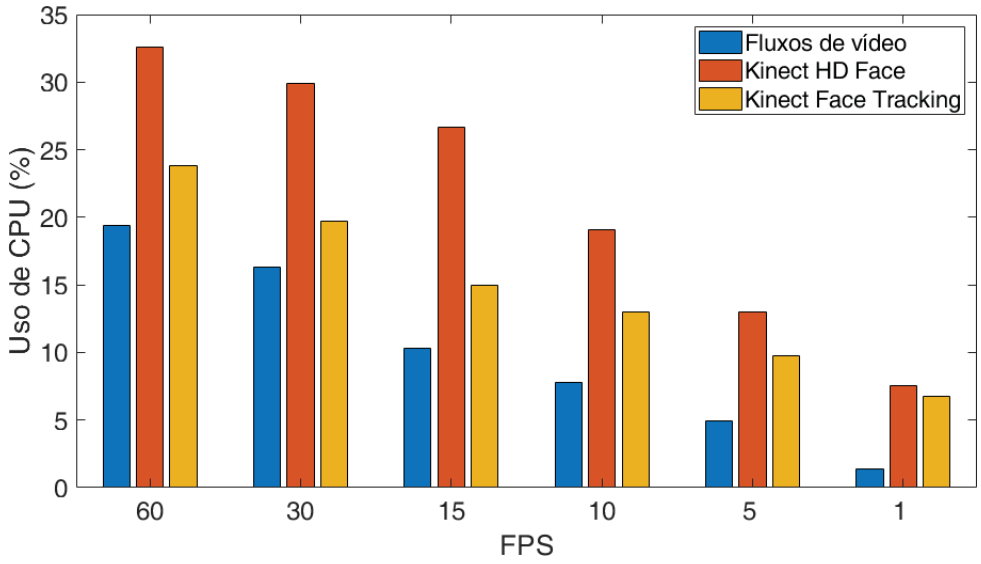
FONTE: O autor (2019).

A utilização das funcionalidades disponibilizadas pela SDK do Kinect dentro MATLAB foi realizada por meio do *toolbox* Kin2 (TERVEN; CÓRDOVA-ESPARZA, 2016).Esta *toolbox* encapsula a maioria das funcionalidades do Kinect for Windows SDK 2.0 em uma única classe com métodos de alto nível, sendo escrita principalmente em C++ com funções MATLAB Mex, fornecendo acesso a *frames* RGB, profundidade e infravermelho; mapeamento de coordenadas; rastreamento de seis corpos em tempo real com 25 articulações e estados de mãos; face e processamento de face de alta definição; e reconstrução 3D em tempo real. O desempenho de um aplicativo que

faz uso da Kin2 perde em média 30% em relação a um aplicativo C++ nativo. No entanto, o uso desta biblioteca reduz em uma ordem de grandeza o tempo de desenvolvimento para prototipagem e pesquisa.

Visto que um dos motivos do uso do Kinect como sensor de varredura tridimensional é a possibilidade de aplicação em sistemas de tempo real, dada a velocidade de varredura do sensor, realizou-se um estudo do impacto em processamento causado pela inclusão das diversas camadas de *software* na captura dos quadros. Verificou-se a utilização de CPU somada dos processos do MATLAB e do driver de captura do Kinect, durante um minuto, realizando a captura e processamento em três cenários distintos: captura dos fluxos de vídeo RGB e profundidade; captura dos fluxos e extração da malha facial pela API Kinect HD Face; captura dos fluxos e extração de pontos fiduciais e características pela API Kinect Face Tracking. Os dados foram capturados em 6 alterativas de velocidade, dado por valores de quadros por segundo (do inglês, Frames per Second, FPS). Segundo apresentado na FIGURA 5.25, mesmo realizando a captura à taxa máxima de 60 quadros por segundo o custo computacional necessário para a captura é baixo. Percebe-se também que a queda de uso de processamento segue um padrão linear em relação à taxa de FPS selecionada. Na comparação com extração de características, a API HD Face apresenta, como esperado, um custo computacional mais elevado, dado a maior quantidade de dados extraídos, bem como os processamentos necessário para geração da malha de pontos faciais fornecida.

FIGURA 5.25 – Comparação de custo computacional para diferentes formas de captura de quadros pelo toolbox Kin2.

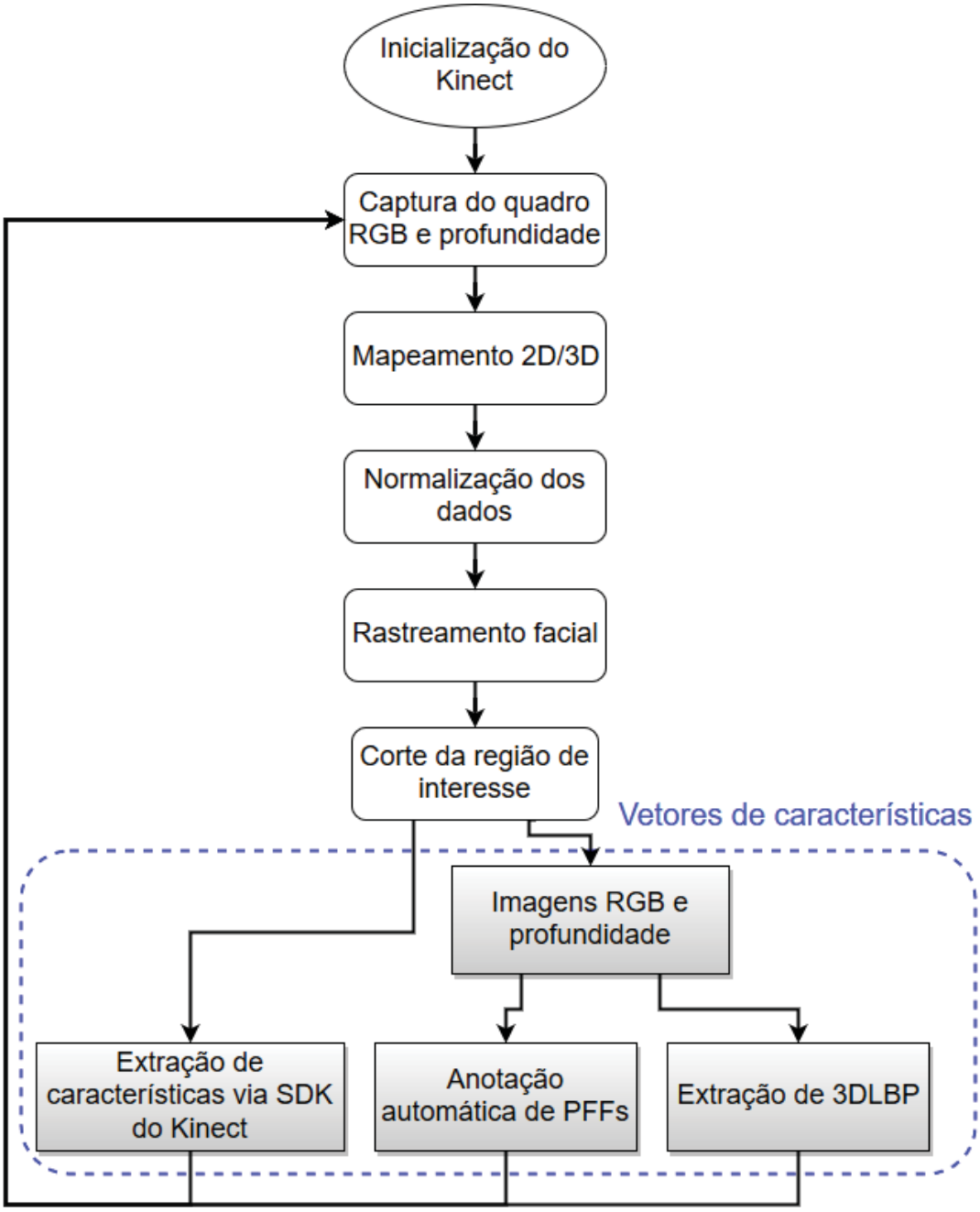


FONTE: O autor (2019).

5.5.2 Fluxo de captura e tratamento de quadros

A captura de imagens e extração do vetor de características que alimenta os algoritmos de AM foi realizada segundo o fluxograma apresentado na FIGURA 5.26.

FIGURA 5.26 – Processo de captura, tratamento e extração de características das imagens adquiridas por meio do sensor Kinect.



FONTE: O autor (2019).

Inicialmente, são executadas as rotinas do *toolbox* Kin2, de forma a instanciar a comunicação com o Kinect, definir quais os fluxos de dados da câmera são

requisitados, bem como os parâmetros de configuração necessários. Foram empregados apenas quatro dos sete fluxos de dados disponíveis. Os fluxos *color* e *depth*, representam dados de imagem puros, sendo armazenados em suas resoluções máximas (1920x1080 e 512x424, respectivamente). Já os fluxos *face* e *HDFace* representam as estruturas de dados que encapsulam as características faciais extraídas pelas APIs Kinect Face Tracking e Kinect Face HD. Os demais fluxos de dados, *infrared, body* e *body_index*, não configuram dados relevantes para o escopo deste trabalho, sendo, portanto, descartados.

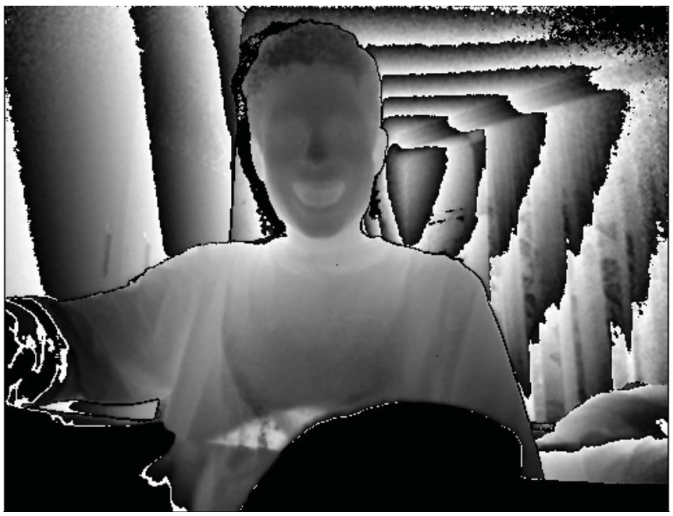
Durante a captura dos quadros de profundidade do Kinect, verificou-se que o sensor IR tem algumas limitações quanto à distância do objeto que está sendo capturado. O sensor é incapaz de detectar objetos posicionados a menos 0,5m de distância do sensor, não permitindo então que a imagem do rosto seja capturada de forma a preencher todo o campo de visão do sensor. Segundo o estudo de YANG et al. (2015), a região com maior precisão de captura do sensor IR se encontra em uma faixa de até 2m de distância por 1m de largura, a partir do centro do sensor. Desta forma, buscou-se posicionar o indivíduo para captura de forma centralizada ao Kinect, a aproximadamente 1~1,5m de distância deste, com anteparo de fundo, preferencialmente liso, de 3 à 4m de distância. Não se fez necessário deixar o Kinect verticalmente alinhado com rosto a ser capturado, porém evitou-se a captura em ângulos muito grandes.

Como as câmeras RGB e IR do Kinect possuem um leve deslocamento de posicionamento, as imagens coloridas e de profundidade capturadas pelo sensor não são perfeitamente alinhadas, tendo angulações de captura ligeiramente diferentes. Além disso, dada a diferença de resolução, a imagem colorida cobre uma área maior de captura do que a imagem de profundidade, como visto nas FIGURA 5.27 e FIGURA 5.28. Desta forma, como as cenas representadas pelas duas imagens são diferentes, um passo de alinhamento e mapeamento é necessário para que as coordenadas de uma imagem sejam compatíveis com os dados apresentados na outra. Felizmente, o *toolbox* Kin2 fornece métodos para realizar a conversão de forma simples por meio da função *mapColorPoints2Depth*, permitindo o alinhamento da imagem.

FIGURA 5.27 – Quadros RGB capturado pelo sensor Kinect à 1920x1080.

FONTE: O autor (2019).

FIGURA 5.28 – Quadro de profundidade capturado pelo sensor Kinect à 512x424.



FONTE: O autor (2019).

Devido ao baixo custo e simplicidade da câmera infravermelha do Kinect, os quadros de profundidade gerados possuem bastante ruído, com alguns problemas de cintilação e ausências de medição em certas regiões da imagem. O ruído nos dados se manifesta como pontos que possuem (incorretamente) profundidade zero (representados como pontos em azul escuro na FIGURA 5.29) continuamente aparecendo e sumindo da imagem. Parte do ruído é causado pela difusão dos raios de luz infravermelha que está sendo refletida pelo objeto atingido. Outra parte é causada pela sombra gerada pelos objetos mais próximos do Kinect sobre os objetos de fundo, o que não permite que a luz infravermelha os atinja para realizar a medição de distância.

FIGURA 5.29 – Ruídos não-normalizados presentes em um quadro de profundidade capturado. Os pontos incorretamente representados são apontados pelas setas vermelhas.

FONTE: O autor (2019).

Como o Kinect é incapaz de mensurar a profundidade nestes pontos, a imagem é preenchida com valores nulos. As regiões com as informações ausentes precisam ser preenchidas antes de usar as imagens de profundidade. Este processo de preenchimento é feito substituindo os pixels de valor zero pela moda dos 25 pixels adjacentes (FIGURA 5.30). O uso da moda dos valores retorna bordas mais nítidas do que o uso de valores médios, evitando que a imagem seja “borrada”.

FIGURA 5.30 – Quadro de profundidade normalizado.

FONTE: O autor (2019).

A API de rastreamento facial da SDK do Kinect fornece, em todo quadro de imagem em que for localizada uma ou mais faces, uma matriz contendo as

coordenadas de localização de uma “caixa” localizada ao em torno da face. Essa matriz é aplicada de forma a recortar as imagens colorida e de profundidade, fazendo que estas contenham somente a região de interesse, ou seja, o rosto do indivíduo rastreado, como apresentado na FIGURA 5.31 (a). De forma a manter o padrão utilizado pela base Bosphorus DB, como foram treinados os algoritmos, foi realizada a segmentação e remoção da região de fundo da imagem cortada, como apresentado na FIGURA 5.31 (b). No caso da imagem de profundidade, foi aplicado um valor de corte para realizar a separação do fundo, de forma que valores de profundidade maiores que 2000 (aproximadamente 2m de distância do sensor) foram tratados como região de fundo e ignorados. As imagens recebidas foram redimensionadas para 277x277 pixels.

FIGURA 5.31 – Quadro RGB (a) após recorte para as coordenadas de localização do rosto e (b) após remoção da região de fundo.

FONTE: O autor (2019).

Após todo o pré-processamento realizado nos quadros capturados pelo Kinect, quatro grupos de características foram extraídas para classificação pelos modelos de AM: as imagens RGB-D puras alimentam a abordagem de CNN; também a partir das imagens RGB-D foram extraídas as características de textura 3DLBP e realizadas as anotações automáticas dos 22 PFFs similares aos da base Bosphorus DB, por meio do uso de uma SVM treinada sobre os PFFs fornecidos pela base. Também foram realizados testes a partir da aplicação dos PFFs e UAs extraídos pela SDK do Kinect. Os resultados obtidos com o emprego da metodologia apresentada neste capítulo são apresentados e detalhados no próximo capítulo.

6 RESULTADOS EXPERIMENTAIS

Todos os experimentos foram realizados em um sistema executando Windows 10 – 64 bits, equipado com uma CPU Intel Core i7-4700HQ 2.40 GHz, 16 GB de memória RAM DDR3 e placa de vídeo Nvidia GTX860M 2GB, implementados sobre o ambiente computacional MATLAB R2018a.

Nesta seção, são discutidas as configurações dos experimentos realizados e os resultados obtidos, bem como uma avaliação crítica do método proposto.

6.1 AVALIAÇÃO SOBRE A BASE DE DADOS

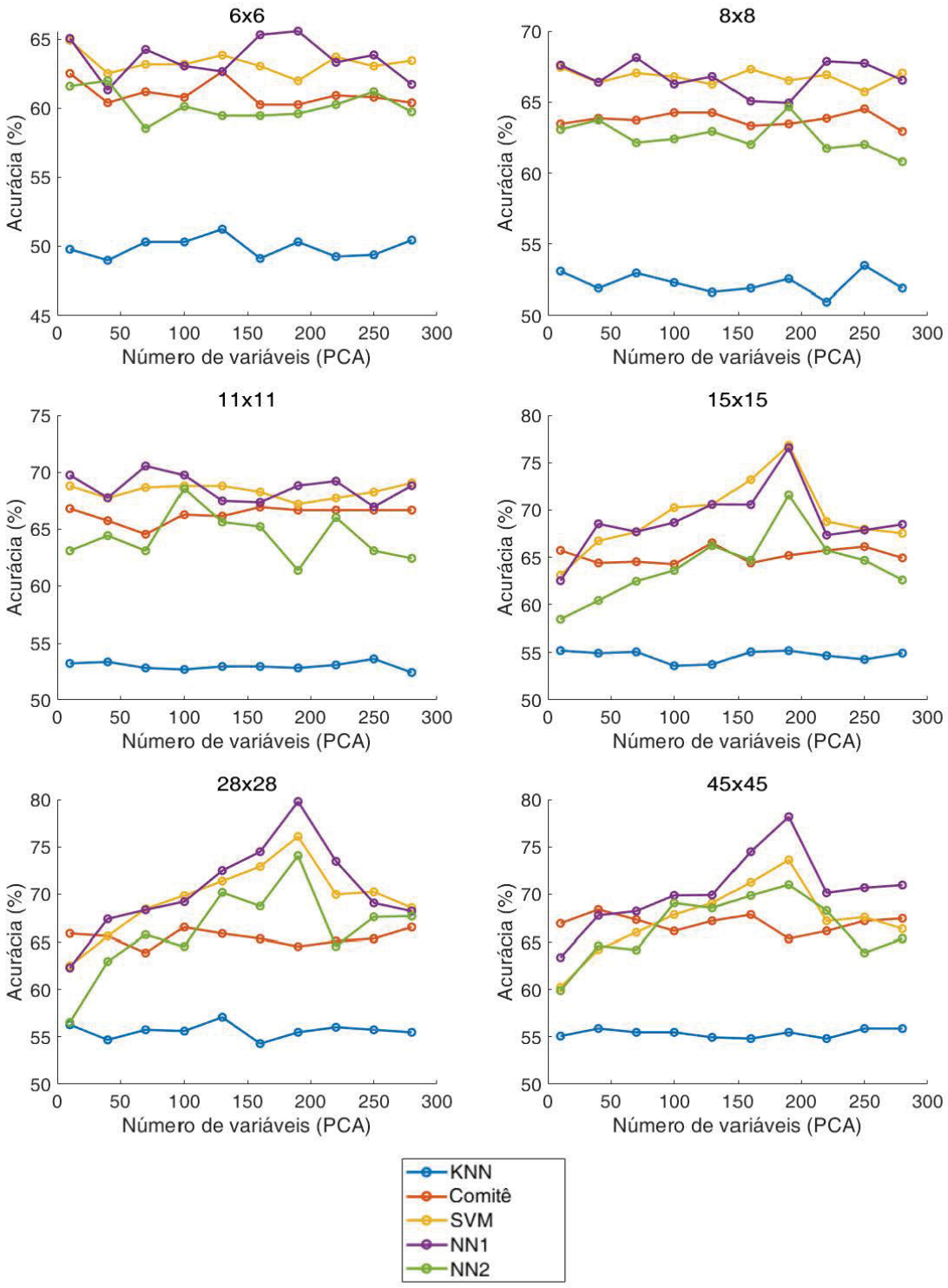
Uma lista extensa de experimentos foi realizada de forma a comparar o desempenho dos modelos de classificação gerados para o reconhecimento das expressões faciais, bem como qual conjunto de características utilizado era o mais adequado para o aproveitamento das informações de profundidade das imagens.

Inicialmente, a combinação dentre conjuntos de características extraídas e algoritmos de AM foi avaliada sob a capacidade de classificação na própria base de dados, representada pelo subconjunto de teste não visto anteriormente pelos modelos treinados.

6.1.1 Métodos 2D

A avaliação dos modelos para duas dimensões se iniciou pelas características extraídas pelos LBP. Como esta foi a única das técnicas avaliadas que gerou vetores de dados de alta dimensionalidade, a análise da aplicação da técnica de redução PCA foi realizada somente para este caso. Como havia sido verificado anteriormente, a seleção de somente as 500 componentes principais dos dados para todos os tamanhos de janela de zoneamento estudados, permitia que mais de 70% da informação presente nos dados fosse representada no subconjunto produzido. Portanto, o PCA foi configurado para realizar a extração deste número de componentes. De forma a identificar o melhor tamanho de janela de zoneamento, bem como o melhor número de componentes principais a ser utilizado, todos os algoritmos foram treinados para todos os tamanhos de janela estudados. A avaliação do PCA foi realizada pelo treinamento com a redução de componentes variando entre 10 e 300, como visto na FIGURA 6.1.

FIGURA 6.1 – Comparação de acurácia entre os algoritmos de AM treinados usando diferentes janelas de zoneamento para o LBP e diferentes níveis de redução de dimensionalidade via PCA.

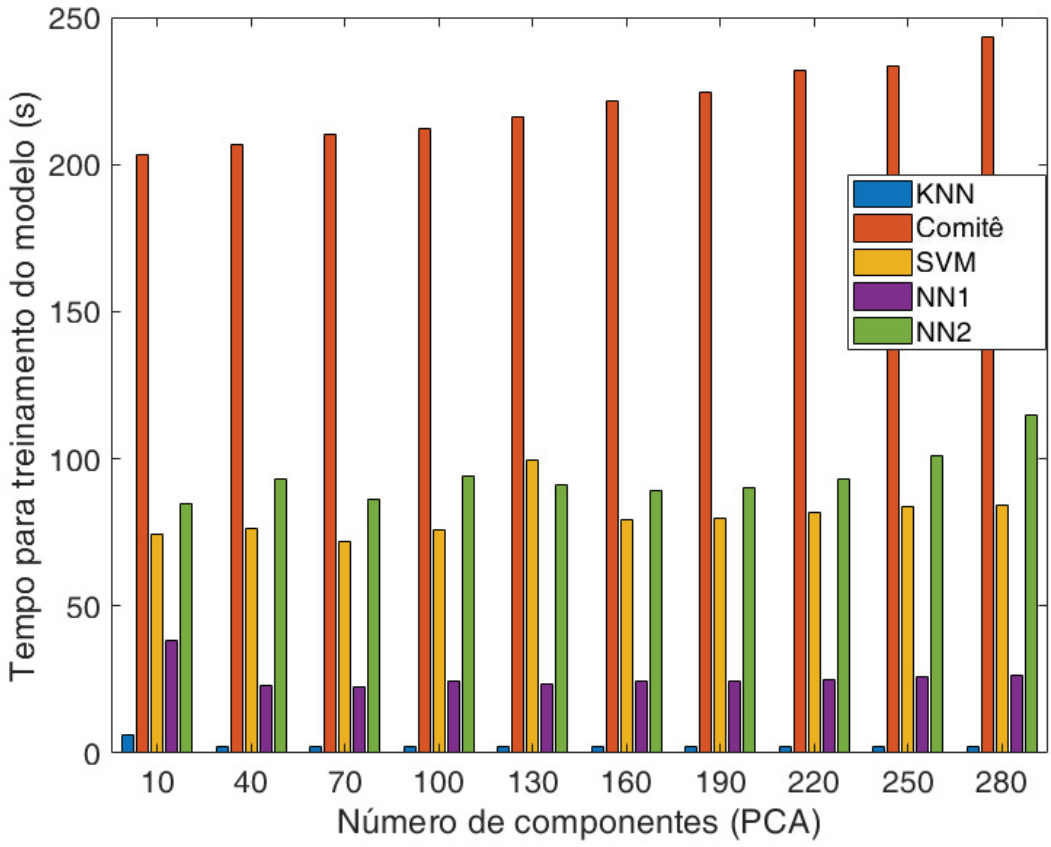


FONTE: O autor (2018).

Diferentemente do esperado, o tempo de treinamento não foi muito afetado pelo aumento do número de características. A abordagem mais afetada foi o comitê de máquina, devido ao grande número de preditores internos que este deve treinar.

Os demais algoritmos sofreram pouca ou nenhuma variação visível. A FIGURA 6.2 apresenta a variação tempo médio calculado dentre as 10 iterações do CV, para a janela de 28x28.

FIGURA 6.2 – Comparação entre o tempo médio de treinamento dos algoritmos de AM a partir de dados com diferentes dimensionalidades.



FONTE: O autor (2018).

TABELA 6.1 – Melhores resultados obtidos na avaliação LBP+PCA, por tamanho de janela de zoneamento. O melhor resultado encontra-se destacado.

Janela de zoneamento	Abordagem de AM	Número de componentes	Acurácia (%)	Tempo de treinamento (s)
6X6	NN1	190	65,12	25,40
8X8	NN1	220	68,00	27,34
11x11	NN1	70	70,90	32,87
15x15	SVM	190	75,84	74,65
28x28	**NN1**	**190**	**79,04**	**27,98**
45x45	NN1	190	78,77	37,01

FONTE: O autor (2018).

Como apresentado na TABELA 6.1, para as características faciais extraídas via LBP, a Rede Neural mais simples (NN1), com apenas 10 neurônios em uma camada oculta, apresentou o melhor desempenho para todos os tamanhos de janela,

com exceção de uma, onde o desempenho foi ligeiramente superado pela SVM. Esta, porém, apresentou um tempo de treinamento 2,5 vezes maior. Na comparação entre os tamanhos de janelas de zoneamento do LBP, foi observado que à medida em que a janela aumentou, capturando informações mais gerais da imagem, a precisão de classificação também se tornou maior, porém após a janela de 28x28, este crescimento não foi mais observado. Quanto ao número de componentes reduzidos pelo PCA, o melhor desempenho para todas as janelas foi maior na região de 150 a 250 componentes, observando uma ligeira queda quando mais componentes foram inseridos. Esta faixa de número de componentes se encontra próximo de explicar 90% da variância dos dados extraídos, como apresentado na FIGURA 5.13.

A utilização dos dados de PFFs foi realizada de forma direta, pois o vetor de dados extraído, mesmo após o processamento, não excedeu 48 dimensões para duas dimensões e 72 dimensões para 3D. Neste conjunto de dados, entretanto, foi necessário lidar com informações faltantes, pois em diversas varreduras da base Bosphorus DB alguns dos pontos não foram marcadas, de forma que existiam amostras com apenas 12 pontos detectados. Os pontos que não foram detectados tiveram seu deslocamento a partir do ponto central Neutro definido com zero, ou seja, considerou-se que não houve movimentação do ponto.

Para as características extraídas a partir das camadas convolucionais da AlexNet, não foi necessário realizar nenhuma modificação na estrutura implementada pelo MATLAB. Como um teste adicional, realizou-se a transferência de conhecimento das características extraídas pela AlexNet, previamente às camadas totalmente conectadas e realizou-se o treinamento de uma SVM, a partir destes dados de treinamento, por meio dos mesmos parâmetros definidos anteriormente.

O resumo dos melhores resultados obtidos é apresentado na TABELA 6.2. Dentre as técnicas avaliadas somente com dados, 2D, a abordagem utilizando uma SVM treinada sobre os dados extraídos pelas camadas convolucionais da AlexNet mostrou um resultado ligeiramente superior a própria AlexNet. Ambas as abordagens, entretanto, se mostram superiores ao único estudo que utilizou somente os dados 2D da Bosphorus DB.

Os tempos de execução medidos para a AlexNet compreendem tanto a fase de extração de características quanto à classificação. Não foi encontrado uma forma de medir os tempos isoladamente.

TABELA 6.2 – Resumo dos melhores resultados obtidos com o treinamento utilizando somente dados 2D em RGB.

Abordagem		**Acurácia (%)**		**Tempo de treinamento (s)**
Extração	AM	Validação	Teste	
LBP + PCA	KNN	75,00	**56,90**	8,42
	SVM	83,00	**75,80**	974,65
	NN1	88,00	**79,04**	27,98
	NN2	89,98	**74,12**	198,32
	Ensemble	80,65	**68,88**	212,65
PFFs	KNN	59,68	**62,24**	17,20
	SVM	70,35	**71,30**	938,17
	NN1	99,05	**54,02**	34,22
	NN2	98,02	**69,09**	145,00
	Ensemble	86,99	**76,05**	122,02
Convolução	SVM	97,70	**81,51**	30,00
	AlexNet	99,99	**80,81**	2545,00[1]
Abordagens da Literatura				
Estudo		**Extração**	**AM**	**Acurácia Final (%)**
(MOHAMMADI et al., 2014)		Dicionário de componentes	SRbC	**72,41**
(AHMED. et al., 2016)		PFFs	SVM	**92,6**

FONTE: O autor (2019).

6.1.2 Métodos 3D

Aplicando os dados RGB-D, os mesmos procedimentos de testes foram realizados.

Os demais resultados obtidos por meio do treinamento com os dados de profundidade são apresentados na TABELA 6.3. É possível perceber que, de forma geral, a precisão obtida pela inclusão de uma nova dimensão de observação aumenta significativamente em comparação com os dados 2D. Mais uma vez, as características extraídas via convolução apresentaram a melhor taxa de acerto nos dados do conjunto de teste.

Entretanto, o uso somente de dados geométricos da face apresentou, na maioria dos casos, resultados inferiores aos de duas dimensões. O aumento do tempo de treinamento para a maioria dos algoritmos de AM também segue o padrão esperado, visto que o tamanho dos vetores de dados utilizados para o treinamento também aumentou.

[1] Esta medida inclui o tempo de extração de características pela camada convolucional.

Como visto na TABELA 6.3, a abordagem de treinamento da SVM utilizando vetores de características extraídas da CNN não foi executada para dados tridimensionais. Isto se deve ao fato da dificuldade de extrair somente as informações convolucionais obtidas da rede do *toolbox* mdCNN.

TABELA 6.3 – Resumo dos melhores resultados obtidos com o treinamento dados 3D em RGB-D.

Abordagem		**Acurácia (%)**		**Tempo de treinamento (s)**
Extração	AM	Validação	Teste	
3DLBP + PCA	KNN	79,86	**72,12**	72,34
	SVM	88,95	**85,22**	1858,00
	NN1	72,00	**62,72**	132,68
	NN2	91,50	**81,25**	890,22
	Ensemble	62,86	**55,12**	1200,43
3D PFFs	KNN	63,00	**12,80**	7,28
	SVM	74,01	**72,84**	48,00
	NN1	63,00	**25,80**	42,01
	NN2	78,25	**79,88**	127,03
	Ensemble	74,01	**42,04**	100,28
Convolução	SVM	-	**-**	-
	AlexNet	73,03	**86,67**	10642,00[2]
Abordagens da Literatura				
Estudo		**Extração**	**AM**	**Acurácia Final (%)**
(VRETOS et al., 2011)		Momentos *Zernike*	SVM	**60.5**
(ZHAO et al., 2013b)		2D (LBP) + PFFs	Bayesian Belief Network	**85.6**
(MANISHA, DR JAGJIT SINGH, 2015)		DRHP e LDSP	Kernel PCA/GDA + CNN	**96,25**
(VIERIU et al., 2015)		LBP	Random Forest	**66,50**
(ZHAO et al., 2016)		*Deformable Partial Facial Model*	SVM	**90.0**
(DERKACH; SUKNO, 2018b)		*Graph Laplacian Features*	SVM	**77,33**

FONTE: O autor (2019).

Mesmo não apresentando os melhores resultados, alguns modelos baseados em características extraídas por meio de 3DLBP obtiveram taxas de acerto muito próximas à CNN AlexNet.

Apesar da melhoria em comparação às abordagens 2D, o melhor resultado obtido ainda é significativamente inferior à abordagens encontradas na literatura, tais como (SAVRAN; SANKUR, 2017) e (KUSUMA; CHUA, 2011). Importante ressaltar,

[2] Esta medida inclui o tempo de extração de características pela camada convolucional.

entretanto, que ambos estudos fazem uso de fusões de vetores de características extraídos por técnicas diferentes. O pequeno conjunto de testes utilizado (330 imagens) também contribui para uma acurácia ligeiramente menor.

6.1.3 Análise comparativa

A fim de facilitar a comparação entre as duas melhores abordagens obtidas, verificando a melhoria possibilitada pela análise dos dados em 3D, a TABELA 6.4 apresenta os valores de acurácia obtidos por classe de expressão facial, para cada uma das classes.

Por meio dos dados apresentados, é possível observar que a abordagem por meio de dados 3D realmente aumentou o nível de precisão em quase todas as classes de expressões faciais. A rede AlexNet conseguiu apresentar resultados muito bons (>90%) em 4 das 7 classes analisadas, quando apresentada a dados tridimensionais.

TABELA 6.4 – Comparação da acurácia obtida pelos melhores métodos em 2D e 3D.

Expressão Facial	**Acurácia (%)**	
	Abordagem 2D (Convolução AlexNet + SVM)	Abordagem 3D (AlexNet)
Desgosto	78,00	82,00
Felicidade	90,48	90,48
Medo	69,23	76,92
Neutro	95,00	95,00
Raiva	81,08	78,38
Surpresa	74,07	96,30
Tristeza	86,36	96,97

FONTE: O autor (2019).

Como apresentado na matriz de confusão, na TABELA 6.5, uma das classes com melhor desempenho de classificação foi a da expressão de Surpresa. Esta, apesar de compartilhar suas principais características faciais com outras classes de expressões (abertura de olhos e boca), as apresenta em intensidades mais elevadas que outras expressões. Acredita-se que este fato tenha contribuído com o acerto na classificação. O mesmo ocorre com a expressão de Tristeza, onde o franzimento da boca representa uma característica marcante e importante para o reconhecimento da expressão. A menor intensidade desta característica em expressões como Desgosto e Medo causou a classificação incorreta de algumas instancias destas classes.

Outro fato interessante de ser observado foi a redução de desempenho na classe da expressão Raiva. Dado o aumento de desempenho nas demais classes, acredita-se que esta redução se deu por peculiaridades específicas do modelo gerado por *overfitting*, dado pequeno tamanho do conjunto de dados utilizado neste trabalho.

TABELA 6.5 – Matriz de confusão da abordagem utilizando AlexNet adaptada sobre dados 3D.

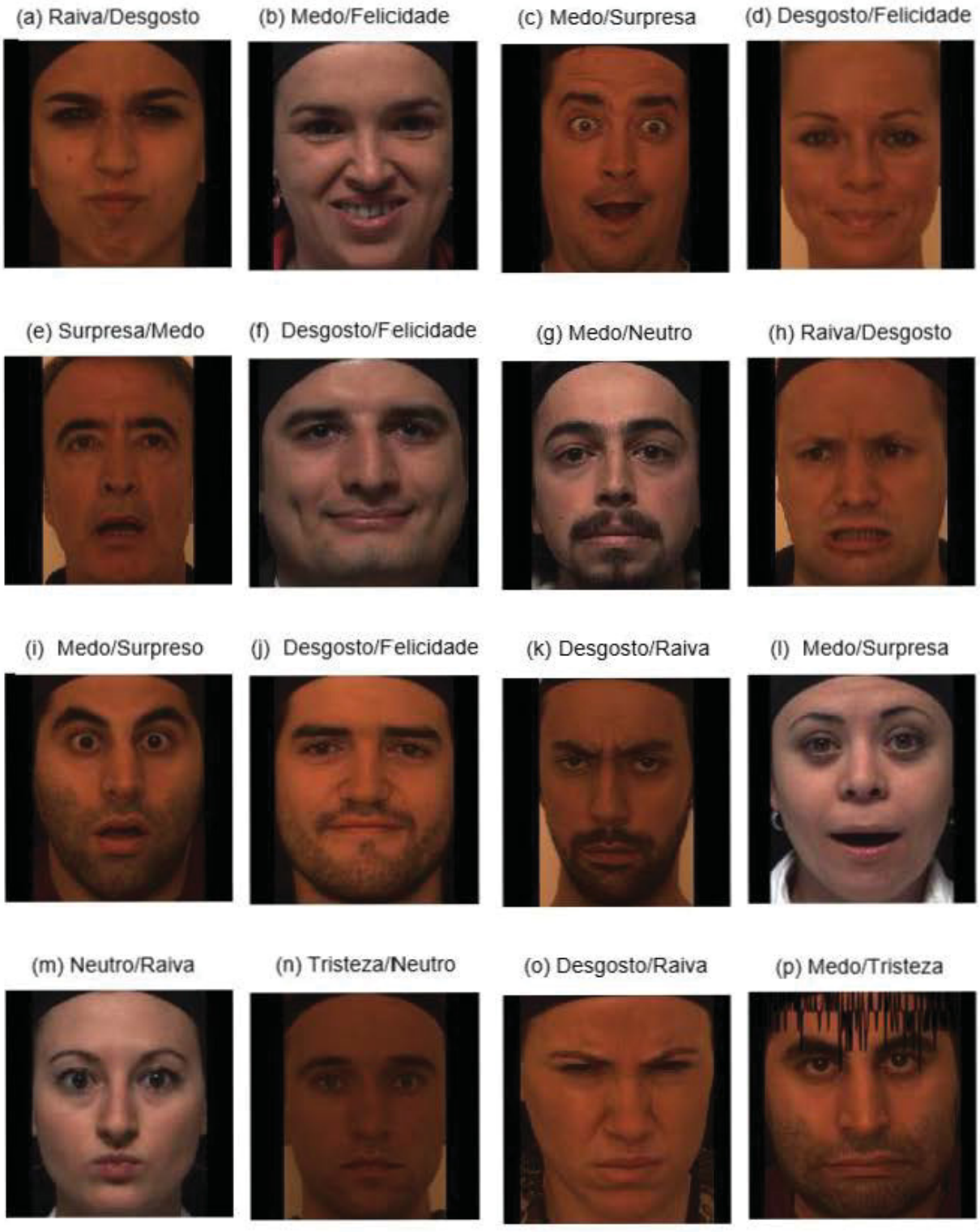
Original \ Predito	Desgosto	Felicidade	Medo	Neutro	Raiva	Surpresa	Tristeza	**Total**
Desgosto	40	3	0	0	2	2	2	49
Felicidade	2	21	0	0	0	0	0	23
Medo	2	1	40	1	1	2	5	52
Neutro	0	1	1	38	0	0	0	40
Raiva	9	3	1	0	58	2	1	74
Surpresa	0	0	1	0	0	26	0	27
Tristeza	1	0	1	0	0	0	64	66
								331

FONTE: O autor (2019).

As classes onde houveram maior erro de classificação são Medo, Desgosto e Raiva. Analisando alguns exemplos de instâncias mal classificadas (FIGURA 6.3), percebe-se que muitas das características faciais presentes na expressão destas emoções são similares entre si, tais como movimentação das sobrancelhas e boca franzida. Em alguns dos erros de classificação, a distinção entre tais emoções pode ser interpretada ambiguamente até mesmo por um humano. A exclusão de imagens com distinção tênue entre emoções do conjunto de treinamento pode contribuir para que o algoritmo isole melhor as características e intensidades representativas de cada emoção. Propõe-se, portanto, uma revisão do etiquetamento das faces na base de dados, buscando a melhoria na classificação destas emoções específicas.

Visto que a abordagem empregando a CNN AlexNet adaptada para dados tridimensionais obteve o melhor desempenho quando aplicada ao conjunto de testes da base de dados, esta foi selecionada para ser aplicada no sistema de reconhecimento por meio dos dados capturados pelo Kinect.

FIGURA 6.3 – Exemplo de instâncias do conjunto de teste incorretamente classificadas pela CNN AlexNet. As classes são apresentadas a forma *Classe Predita / Classe Real*.



FONTE: O autor (2018).

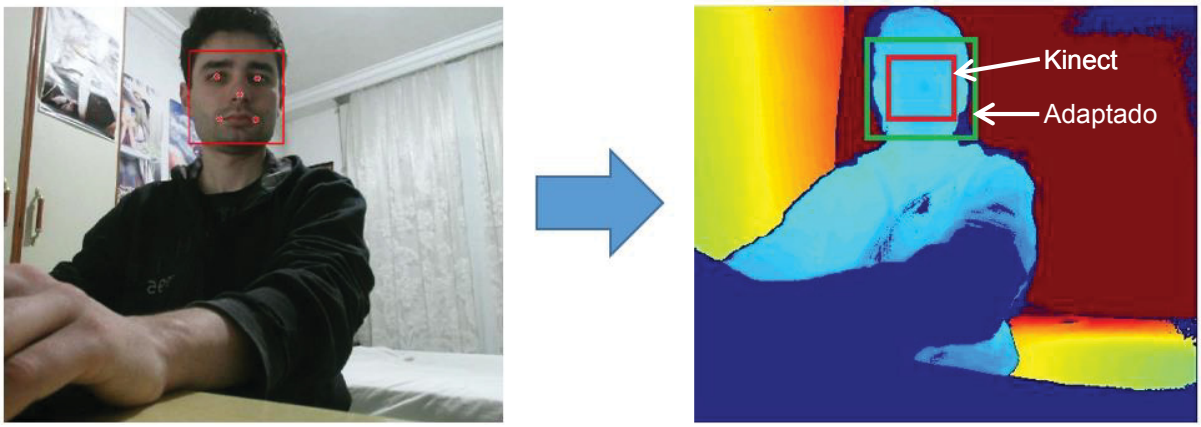
6.2 CLASSIFICAÇÃO DOS DADOS DO KINECT

Para aplicação do modelo de conhecimento gerado sobre os dados do Kinect foi necessário primeiramente realizar a captura de uma pequena base de dados para validação. A base de avaliação foi composta por capturas de 12 indivíduos, com idades variando de 14 a 86 anos de idade, sendo 3 mulheres e 9 homens. Para cada uma das sete expressões faciais analisadas, foram realizadas 3 capturas por indivíduos, buscando ligeiras diferenças entre as capturas, tais como leve rotação de cabeça, diferente intensidade de expressão e condição de iluminação. Tal abordagem busca simular uma aplicação em ambiente descontrolado. Todas as capturas totalizaram uma base de 252 imagens RGB-D.

Durante a aquisição dos quadros de imagem, enfrentou-se algumas limitações impostas pelo *hardware* do Kinect. O primeiro deste foi a limitação da sua faixa de captura, não permitindo a aproximação do indivíduo a menos de 75 cm e perdendo muita fidelidade de detalhes à medida que a distância aumenta para alguns metros. Além disso, devido ao campo de visão da lente da câmera, mesmo a pouco mais de um metro de distância, a face capturada passa a ter uma resolução muito pequena. Desta forma, a captura das faces foi realizada com o sujeito posicionado entre 1 a 1,2m da câmera.

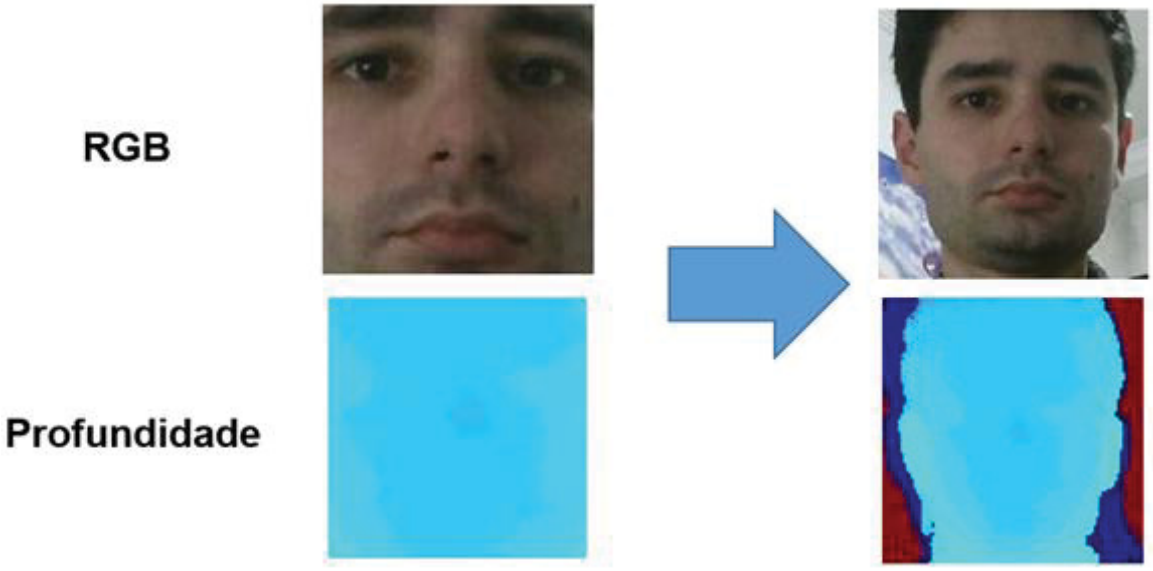
Além disso, houve dificuldade no passo de mapeamento das coordenadas da imagem 2D para a imagem gerada pelo vetor dos dados de profundidade. O método disponibilizado pela biblioteca Kin2 produziu coordenadas ligeiramente incorretas, o que causava o corte excessivo da região de interesse da imagem de profundidade, como apresentado nas FIGURAS FIGURA 6.4 e FIGURA 6.5. Para solucionar este problema foi adicionada uma variável de compensação às coordenadas de forma a alargar a área a ser recortada. Em algumas situações, entretanto, esta compensação acabou por permitir que uma certa quantidade do plano de fundo indesejável fosse incluída aos dados, não representando perfeitamente o contorno da face extraída. A presença de plano de fundo pode ser prejudicial à capacidade de classificação do algoritmo de IA.

FIGURA 6.4 – Mapeamento das coordenadas da face da imagem RGB para profundidade, realizada pelo Kinect, em vermelho, e adaptada, em verde.



FONTE: O autor (2019).

FIGURA 6.5 – Corte incorreto (à esquerda) e correto (à direita) da região de interesse da imagem de profundidade do Kinect.



FONTE: O autor (2019).

A classificação de uma nova amostra pela CNN AlexNet utilizando as quatro camadas de entrada levou, em média 5,89 segundos. A CNN obteve uma taxa de acurácia de 72,62% classificando as capturas do Kinect.

Apesar do baixo desempenho na adaptação do conhecimento, o modelo apresentou boa capacidade de identificação das expressões de Felicidade e Surpresa, principalmente em capturas onde houve grande abertura ou movimentação na região da boca. Já expressões que apresentam diferenças mais sutis, tais como Tristeza, Desgosto e Neutro, apresentaram as piores taxas de classificação, como pode ser visto na TABELA 6.6 e na matriz de confusão da TABELA 6.7.

TABELA 6.6 – Comparação da acurácia facial da CNN AlexNet por expressão na base BosphorusDB e nas capturas do Kinect.

Expressão Facial	**Acurácia (%)**	
	Bosphorus DB	Kinect
Desgosto	82,00	55,56
Felicidade	90,48	75,00
Medo	76,92	52,78
Neutro	95,00	88,89
Raiva	78,38	83,33
Surpresa	96,30	91,67
Tristeza	96,97	61,11

FONTE: O autor (2019).

TABELA 6.7 – Matriz de confusão da abordagem utilizando AlexNet adaptada sobre os dados capturados pelo Kinect.

		Predito							
		Desgosto	Felicidade	Medo	Neutro	Raiva	Surpresa	Tristeza	**Total**
Original	Desgosto	20	0	8	3	1	0	4	36
	Felicidade	2	27	0	3	1	3	0	36
	Medo	5	0	19	2	1	1	8	36
	Neutro	1	1	1	32	0	1	0	36
	Raiva	5	0	1	0	30	0	0	36
	Surpresa	0	2	1	0	0	33	0	36
	Tristeza	1	2	6	5	0	0	22	36
									252

FONTE: O autor (2019).

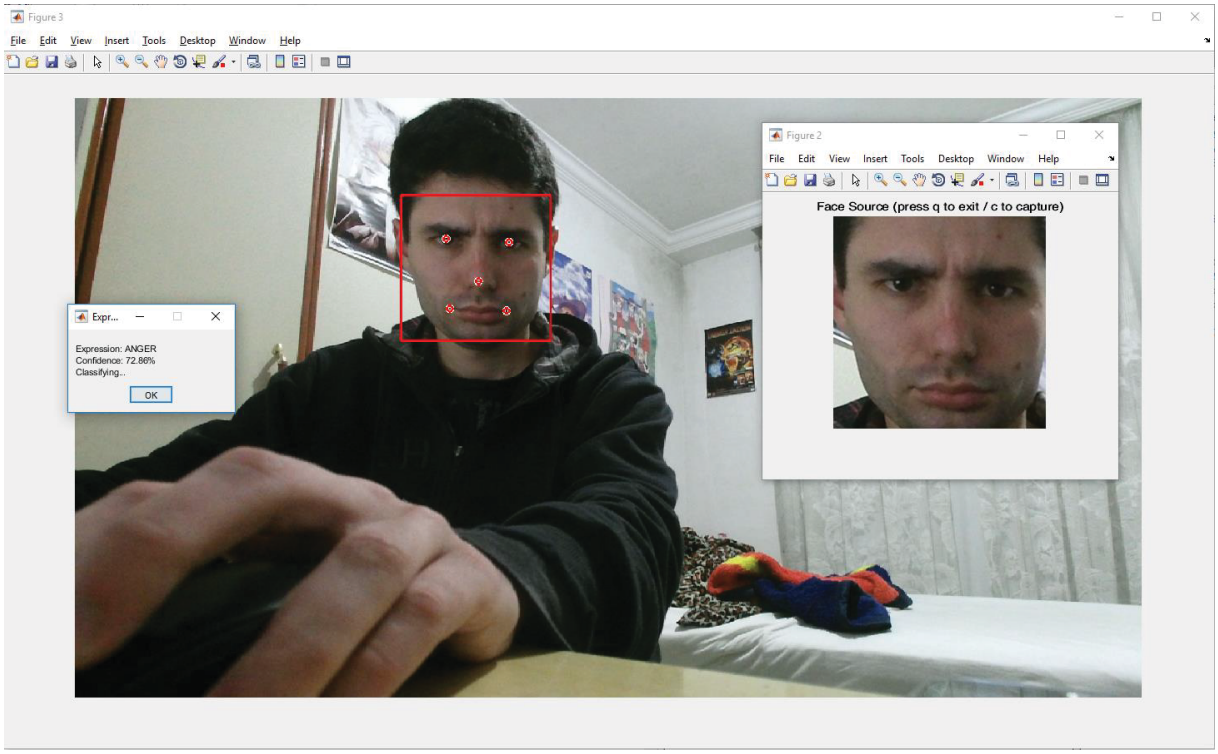
Após a análise de precisão do modelo, foi desenvolvido um protótipo de sistema de classificação em tempo real, por meio do fluxo de vídeo fornecido pelo Kinect, apresentado na FIGURA 6.6. A implementação recebe o fluxo de vídeo diretamente da câmera e realizada a captura de um quadro para classificação a cada seis segundos, a partir do momento em que é sinalizada a identificação de uma face pelo Kinect.

O sistema, entretanto, apresenta limitações com relação à dois fatores:

- Velocidade de captura, devido à falta de otimização dos processos de pré-processamento do quadro para extração da face dentro da imagem;
- Precisão de classificação, devido à baixa acurácia do modelo aplicado para a classificação.

Além disso, como o modelo não foi treinado levando em conta imagens faciais que não estejam diretamente voltadas para a câmera, em muitos momentos onde existe movimentação da cabeça do indivíduo uma classificação incorreta é realizada.

FIGURA 6.6 – Protótipo do sistema de classificação de expressões faciais desenvolvido.



FONTE: O autor (2019).

Mesmo com tais limitações o sistema é capaz de validar a aplicação do Kinect como sensor de aquisisção de imagens tridimensionais para tal aplicação, de forma a ser considerada bem-sucedida a aplicação proposta neste trabalho.

7 CONSIDERAÇÕES FINAIS

Este trabalho realiza um estudo comparativo entre técnicas para reconhecimento de expressões faciais em imagens 2D e 3D, por meio de abordagens de aprendizado de máquina, baseado em varreduras 3D geradas por um sensor de alta resolução, com o intuito de aplicar o modelo de conhecimento gerado para o domínio de dados gerados pelo sensor de baixa resolução Kinect. Além de comparar o desempenho de algoritmos de AM distintos, tais como SVM, KNN, RNA, CNN e Comitês de Máquinas, verificou-se como as diferentes metodologias para extração de características faciais influenciam o processo de classificação, buscando identificar que tipo de técnica, isoladamente, produz a melhor precisão de classificação.

O desenvolvimento deste trabalho foi dividido em duas partes, primeiramente a implementação de técnicas de extração de características faciais via textura, geometria e convolução, bem como a aplicação destas características em cinco algoritmos de AM. Para se ter uma base comparativa com estudos na literatura, além de possibilitar a identificação do fator de melhoria gerado pela inclusão dos dados 3D, verificou-se também a acurácia dos modelos quando treinados sem a utilização dos dados de varredura de profundidade.

Ambas as abordagens obtiveram bons resultados, sendo que, como esperado, a inserção da dimensão adicional de profundidade nos dados permitiu a obtenção de uma taxa de acerto 86,67% por meio da adaptação da CNN AlexNet. Entretanto, a adaptação deste modelo para o domínio de dados produzido pelo Kinect apresentou grande perda de desempenho, atingindo somente 72,62% de acerto, mesmo com os pré-processamentos realizados nos dados de captura do Kinect.

Com os resultados apresentados no Capítulo 6 pode-se afirmar que todos os objetivos propostos no início deste trabalho foram concluídos com sucesso, porém existem abordagens a serem avaliadas de forma a melhorar a acurácia do modelo de classificação, bem como procedimentos de otimização para permitir que o sistema de identificação utilizando o Kinect possa ser aplicado em tempo real sobre ambientes descontrolados.

Além disso, as seguintes conclusões também podem ser apontadas:

- As técnicas baseadas em textura da imagem da face (LBP e convolução) apresentam melhor desempenho e versatilidade que as técnicas baseadas em PFFs;

- Os algoritmos apresentaram melhor desempenho na classificação de expressões que apresentam características ou movimentações mais marcantes na face, tais como Felicidade e Surpresa;
- Mesmo sendo rudimentar, o protótipo de sistema de classificação desenvolvido prova a capacidade de aplicação do Kinect em sistemas de tempo real, se mostrando como uma alternativa viável aos sensores tradicionais, mesmo possuindo baixa resolução e altos níveis de ruído.

7.1 TRABALHOS FUTUROS

Em trabalhos futuros, de forma a buscar uma melhor taxa de acerto para o sistema, propõe-se a aplicação de novas abordagens de CNN mais complexas, tais como VGG26, GoogLeNet ou ResNet-152 para melhorar o desempenho principalmente nas classes onde os algoritmos “clássicos” tiveram maior dificuldade de diferenciação, como Medo, Desgosto e Surpresa.

Além disso, o desempenho do sistema pode ser aprimorado visando aplicações em ambientes descontrolados, por meio da complementação da base de dados de treinamento com novos dados contendo emoções faciais expressas sob diferentes ângulos e posições de cabeça, bem como sofrendo com oclusões de regiões da face.

A utilização de dados com dimensões temporais pode ter um impacto relevante no desempenho do sistema, visto que algumas expressões faciais apresentam uma forte correlação com o tempo de transição e apresentação. A expressão Surpresa, por exemplo, tem como suas principais características a forte intensidade de deformação facial e a extrema velocidade de transição para a expressão. Além disso, a aplicação de técnicas como a RNA Memória de Longo Prazo (do inglês, *Long Short-Term Memory*, ou LSTM) possui um potencial interessante por trabalhar especificamente com este tipo de informação.

REFERÊNCIAS

AHMED., H. A.; RASHID, T. A.; SADIQ, A. T. Face Behavior Recognition Through Support Vector Machines. **International Journal of Advanced Computer Science and Applications**, v. 7, n. 1, p. 101–108, 2016.

AHONEN, T.; HADID, A.; PIETIKÄINEN, M. Face description with local binary patterns: Application to face recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 28, n. 12, p. 2037–2041, 2006.

AIRES, G. V.; SANTOS, I. M. M.; BRANDÃO, P. S.; DE, F. L.; FAGUNDES, F. **Reconhecimento de expressões faciais através de análise facial computadorizada utilizando o Kinect v2**. XIX Encoinfo – Congresso de Computação e Tecnologias da Informação. Anais... Palmas. p.95–105, 2014.

AMARA, K.; RAMZAN, N.; ACHOUR, N.; et al. **Emotion Recognition via Facial Expressions**. Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA. Anais... , 2019.

BELLMAN, R. The Theory of Dynamic Programming. **Bulletin of the American Mathematical Society**, 1954.

BERRETTI, S.; AMOR, B. BEN; DAOUDI, M.; BIMBO, A. DEL. 3D facial expression recognition using SIFT descriptors of automatically detected keypoints. **Visual Computer**, v. 27, n. 11, p. 1021–1036, 2011.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. 1st ed. New York, USA: Springer-Verlag, 2006.

BOARETTO, M. A. R. **Machine Learning techniques applied in human activity recognition using RGB-D videos**. 100 f. Dissertação (Mestrado em Engenharia Elétrica) - Departamento de Engenharia Elétrica Universidade Federal do Paraná, Curitiba, 2017.

BOUZALMAT, A.; KHARROUBI, J.; ZARGHILI, A. Comparative study of PCA, ICA, LDA using SVM classifier. **Journal of Emerging Technologies in Web Intelligence**, v. 6, n. 1, p. 64–68, 2014.

BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, 1996.

BREUER, R.; KIMMEL, R. **A Deep Learning Perspective on the Origin of Facial Expressions**. Dissertação (Mestrado em Ciência da Computação) - Computer Science Department, Israel Institute of Technology, Haifa, Israel, 2017.

BROWNE, M. W. Cross-validation methods. **Journal of Mathematical Psychology**, v. 44, n. 1, p. 108–132, 2000.

BRUZZONE, L.; MARCONCINI, M. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 32, n. 5, p. 770–787, 2010.

BULWER, J. **Pathomyotamia, Or, A Dissection of the Signi cative Muscles of the A ections of the Minde: Being an Essay to a New Method of Observing the Most Important Movings of the Muscles of the Head, as They are the Neerest and Immediate Organs of the Voluntarie Or**. W.W. for Humphrey Moseley, 1649.

BURGES, C. J. C. Dimension Reduction: A Guided Tour. **Foundations and Trends® in Machine Learning** Hanover, USA: Now Publishers Inc., 2009. v. 2, p.275–364.

CARDIA NETO, J. B. **Reconhecimento de Faces 3D com Kinect**. 66 f. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Ciência da Computação, Universidade Estadual Paulsta “Júlia de Mesquita Filho“, São Paulo, 2014.

CHANTHAPHAN, N.; UCHIMURA, K.; SATONAKA, T.; MAKIOKA, T. **Facial Emotion Recognition Based on Facial Motion Stream Generated by Kinect.** Proceedings - 11th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2015. Anais... Bangkok, Thailand: IEEE. p.117–124, 2016.

COHEN, I.; SEBE, N.; GARG, A.; CHEN, L. S.; HUANG, T. S. Facial expression recognition from video sequences: Temporal and static modeling. **Computer Vision and Image Understanding**, v. 91, n. 1–2, p. 160–187, 2003.

CORTES, C.; VAPNIK, V. **Support-Vector Networks**. Norwell, USA: Kluwer Academic Publishers, 1995.

COSSETIN, M. J. **Reconhecimento De Expressões Faciais Utilizando Redução De Dimensionalidade Para Estratégia De Classificação Um-Contra-Um**. 137 f. Dissertação (Mestrado em Informática Aplicada) - Programa de Pós-Graduação em Informática Aplicada, Pontifíicia Universidade Católica do Paraná, Curitiba, 2015.

CYBERWARE INC. Cyberware laser scanner. Disponível em: <http://cyberware.com/products/scanners/px.html>. Acesso em: 20/10/2017.

DARWIN, C.; CUMMINGS, M. M., DUCHENNE, G.-B. **The expression of the emotions in man and animals**. London, 1872.

DENG, J.; PANG, G.; ZHANG, Z.; et al. CGAN Based Facial Expression Recognition for Human-Robot Interaction. **IEEE Access** IEEE, v. 7, n. c, p. 9848–9859, 2019.

DERKACH, D.; SUKNO, F. M. Automatic local shape spectrum analysis for 3D facial expression recognition. **Image and Vision Computing** Elsevier B.V, v. 79, p. 86–98, 2018.

DIETTERICH, T. G. Ensemble Methods in Machine Learning. **Multiple Classifier Systems**, v. 1857, p. 1–15, 2000.

DING, H. Combining 2D facial texture and 3D face morphology for estimating people’s soft biometrics and recognizing facial expressions. **Http://Www.Theses.Fr**, 2016.

DUCHENNE, G. B.; CUTHBERTSON, R. A. **The Mechanism of Human Facial Expression**. Cambridge, UK: Cambridge University Press, 1990.

EKMAN, P. **Emotion in the Human Face**. 2nd Editio ed. Cambridge, MA, USA: Cambridge University Press, 1982.

EKMAN, P. Strong Evidence for Universals in Facial Expressions: A Reply to Russell’s Mistaken Critique. **Psychological Bulletin**, 1994.

EKMAN, P.; FRIESEN, W. V.; HAGER, J. C. **Facial Action Coding System**. Salt Lake City, UT: Research Nexus, 2002.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. **Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina**. 1st ed. São Paulo: LTC Editora, 2011.

FLECK, L.; TAVARES, M. H. F.; EYNG, E.; HELMANN, A. Redes Neurais Artificiais: Princípios Básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, Curitiba, v. 7, n. 15. 2016.

FUKUSHIMA, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. **Neural Networks**, v. 1, n. 2, p. 119–130, 1988.

GARTY, H. hagaygarty/mdCNN. Disponível em: <https://github.com/hagaygarty/mdCNN>. Acesso em: 2/1/2019.

GEISSER, S. The predictive sample reuse method with applications. **Journal of the American Statistical Association**, v. 70, n. 350, p. 320–328, 1975.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. **Adaptive Computation and Machine Learning** Cambridge, MA, USA: MIT Press, 2017.

GÜNAY, A.; NABIYEV, V. V. **Automatic age classification with LBP.** 2008 23rd International Symposium on Computer and Information Sciences, ISCIS 2008. Anais... Istanbul, Turkey: IEEE. p.1–4, 2008.

HAAMER, R. E.; RUSADZE, E.; LÜSI, I.; et al. Review on Emotion Recognition Databases. **Human-Robot Interaction - Theory and Application**. p.39–64, 2018.

HANJALIC, A.; XU, L. Q. Affective video content representation and modeling. **IEEE Transactions on Multimedia**, v. 7, n. 1, p. 143–154, 2005.

HAPPY, S. L.; PATNAIK, P.; ROUTRAY, A.; GUHA, R. The Indian Spontaneous Expression Database for Emotion Recognition. **IEEE Transactions on Affective Computing**, v. 8, n. 1, p. 131–142, 2017.

HOLLANDA, A. G. **Detecção de Pontos Fiduciais Faciais usando Filtros de Correlação e Correspondências Espaciais**. 104 f. Dissertação (Mestrado em Engenharia Elétrica) - Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

HOSSAIN, M. S. Patient State Recognition System for Healthcare Using Speech and Facial Expressions. **Journal of Medical Systems**, v. 40, n. 12, 2016.

HUANG, Y.; WANG, Y.; TAN, T. **Combining Statistics of Geometrical and Correlative Features for 3D Face Recognition.** Procedings of the British Machine Vision Conference 2006. Anais... Edinburgh, Scotland: BMVA. p.879–888, 2006.

HUBEL, D. H.; WIESEL, T. N. Receptive fields of single neurones in the cat's striate cortex. **The Journal of Physiology**, v. 148, n. 3, p. 574–591, 1959.

HUYNH, T.; MIN, R.; DUGELAY, J. L. An efficient LBP-based descriptor for facial depth images applied to gender recognition using RGB-D face data. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 7728 LNCS, p. 133–145, 2013.

JAIMES, A.; SEBE, N. Multimodal human-computer interaction: A survey. **Computer Vision and Image Understanding**, v. 108, n. 1–2, p. 116–134, 2007.

JIE, Z.; MAHMOUD, M.; STAFFORD-FRASER, Q.; et al. **Analysis of yawning behaviour in spontaneous expressions of drowsy drivers.** Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018. Anais... Xi'an, China, 2018.

KANADE, T.; COHN, J. F.; TIAN, Y. **Comprehensive database for facial expression analysis.** Proceedings - 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2000. Anais... Grenoble, France: IEEE. p.46–53, 2000.

KANDASWAMY, C. **Contributions on Deep Transfer Learning**. 141 f. Universidade

do Porto, 2016.

KANG, H. B. **Various approaches for driver and driving behavior monitoring: A review.** Proceedings of the IEEE International Conference on Computer Vision. Anais... Sydney, NSW, Australia. p.616–623, 2013.

KEARNS, M. Boosting Theory Towards Practice : Recent Developments in Decision Tree Induction and the Weak Learning Framework. **Information and Computation**, p. 1991–1993, 1993.

KIM, S.; KIM, H. **Deep Explanation Model for Facial Expression Recognition Through Facial Action Coding Unit.** 2019 IEEE International Conference on Big Data and Smart Computing (BigComp). Anais... Kyoto, Japan. p.1–4, 2019.

KRIZHEVSKY, A.; HINTON, G. E. I**mageNet Classification with Deep Convolutional Neural Networks.** NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems. Anais... Lake Tahoe, Nevada, USA. p.1097–1106, 2012.

KUMAR, Y.; SHARMA, S. **A systematic survey of facial expression recognition techniques.** Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017. Anais... Erode, India. v. 2018–Janua, p.1074–1079, 2018.

KUMARI, J.; RAJESH, R.; POOJA, K. M. Facial Expression Recognition: A Survey. **Procedia Computer Science.** Elsevier B.V. v. 58, p.486–491, 2015.

KUSUMA, G. P.; CHUA, C. S. PCA-based image recombination for multimodal 2D + 3D face recognition. **Image and Vision Computing** Elsevier B.V., v. 29, n. 5, p. 306–316, 2011..

LAMB, T. D. Evolution of the eye. Scientists now have a clear vision of how our notoriously complex eye came to be. **Scientific American**, v. 305, n. 1, p. 64–69, 2011.

LI, B. Y. L.; MIAN, A. S.; LIU, W.; KRISHNA, A. **Using Kinect for face recognition under varying poses, expressions, illumination and disguise.** Proceedings of IEEE Workshop on Applications of Computer Vision. Anais... Tampa, FL, USA: IEEE. p.186–192, 2013.

LI, S. Z.; JAIN, A. K. **Handbook of Face Recognition**. London,UK: Handbook of Face Recognition, 2011.

LI, Y.; GAO, J.; LI, Q.; FAN, W. Ensemble learning. **Data Classification: Algorithms and Applications**, 2014.

LI, Y.; ZENG, J.; SHAN, S.; CHEN, X. Occlusion Aware Facial Expression Recognition Using CNN With Attention Mechanism. **IEEE Transactions on Image Processing** IEEE, v. 28, n. 5, p. 2439–2450, 2019.

LIN, D.; SUN, L.; TOH, K. A.; ZHANG, J. B.; LIN, Z. Biomedical image classification based on a cascade of an SVM with a reject option and subspace analysis. **Computers in Biology and Medicine**, v. 96, p. 128–140, 2018.

LIU, P.; HAN, S.; MENG, Z.; TONG, Y. **Facial expression recognition via a boosted deep belief network.** Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Anais... Columbus, OH, USA: IEEE. p.1805–1812, 2014.

LOPES, A. T. **Facial Expression recognition using Deep Learning - Convolutional Neural Network**. 89 f. Dissertação (Mestrado em Informática) - Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo, Vitória, 2016.

LU, J.; SIBAI, H.; FABRY, E. **Adversarial Examples that Fool Detectors**. 2017.

LYONS, M. J. Automatic classification of single facial images. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 21, n. 12, p. 1357–1362, 1999.

MA, X.-H.; TAN, Y.-Q.; ZHENG, G.-M. A fast classification scheme and its application to face recognition. **Journal of Zhejiang University: Science C**, v. 14, n. 7, p. 561–572, 2013.

MA, Y.; GUO, G. **Support Vector Machines Applications**. Springer International Publishing, 2014.

MANISHA, DR JAGJIT SINGH, D. N. R. P. Facial Expression Recognition using Salient Feature and Neural Network. **International Journal of Computer Science and Information Technologies**, v. 6, n. 3, p. 3249–3251, 2015.

MAO, Q. Using Kinect for real-time emotion recognition via facial expressions. **Frontiers of Information Technology & Electronic Engineering**, v. 16, n. 4, p. 272–282, 2015.

MARR, D.; HILDRETH, E. Theory of edge detection. **Proceedings of the Royal Society of London - Biological Sciences**, v. 207, n. 1167, p. 187–217, 1980.

MASE, K.; PENTLAND, A. Recognition of Facial Expression from Optical Flow. **IEICE TRANSACTIONS on Information and Systems**, 1991.

MATSUDA, Y.-T.; FUJIMURA, T.; KATAHIRA, K.; et al. The implicit processing of categorical and dimensional strategies: an fMRI study of facial emotion perception. **Frontiers in Human Neuroscience**, v. 7, 2013.

MAYYA, V.; PAI, R. M.; MANOHARA PAI, M. M. Automatic Facial Expression Recognition Using DCNN. **Procedia Computer Science** The Author(s), v. 93, n. September, p. 453–461, 2016.

MEHRABIAN, A. Communication Without Words. **Psychology Today, Vol 2** New York, USA: Ziff-Davis Publishing Co., p. 53–56, 1968.

MIAN, A. S.; BENNAMOUN, M.; OWENS, R. Keypoint detection and local feature matching for textured 3D face recognition. **International Journal of Computer Vision**, v. 79, n. 1, p. 1–12, 2008.

MICROSOFT CORPORATION. Kinect. Disponível em: <http://www.microsoft.com/en-us/kinectforwindows/>. Acesso em: 20/10/2017.

MIN, R.; KOSE, N.; DUGELAY, J. L. KinectfaceDB: A kinect database for face recognition. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 44, n. 11, p. 1534–1548, 2014.

MINOLTA CO. LTD. Minolta 3D Scanner. Disponível em: <https://sensing.konicaminolta.us/services/discontinued-services/3d-support/>. Acesso em: 20/10/2017.

MOHAMMADI, M. R.; FATEMIZADEH, E.; MAHOOR, M. H. PCA-based dictionary building for accurate facial expression recognition via sparse representation. **Journal Of Visual Communication and Image Representation** Elsevier Inc., v. 25, n. 5, p.

1082–1092, 2014.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre Aprendizado de Máquina. **Sistemas inteligentes: fundamentos e aplicações**. p.525, 2003.

NASCIMENTO JR, C. L.; YONEYAMA, T. **Inteligência Artificial em Controle e Automação**. 1ª Edição ed. São Paulo: Editora Edgard Blüche LTDA, FAPESP, 2004.

NEOCLEOUS, C.; SCHIZAS, C. Artificial Neural Network Learning: A Comparative Review. **Methods and Applications of Artificial Intelligence** Berlin, Heidelberg: Springer, v. 2308, p. 300–313, 2002.

O'BRIEN, T. Microsoft's new Kinect is official: larger field of view, HD camera, wake with voice. Disponível em: <https://www.engadget.com/2013/05/21/microsofts-new-kinect-is-official/>. .

OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 7, p. 971–987, 2002.

OLIVER, N. M.; PENTLAND, A. P.; BÉRARD, F. LAFTER: A real-time face and lips tracker with facial expression recognition. **Pattern Recognition**, v. 33, n. 8, p. 1369–1382, 2000.

PAN, J. **Feature-based Transfer Learning with Real-world Applications**. 128 f. The Hong Kong University of Science and Technology, 2010.

PANTIC, M.; SEBE, N.; COHN, J. F.; HUANG, T. **Affective multimodal human-computer interaction.** Proceedings of the 13th annual ACM international conference on Multimedia - MULTIMEDIA '05. Anais... Hilton, Singapore: ACM. p.669–676, 2005.

PEARSON, K. LIII. *On lines and planes of closest fit to systems of points in space*. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, 1901.

PEREIRA, H. A.; SOUZA, A. F. D.; MENEZES, C. S. D. **Obtaining evidence of learning in digital games through a deep learning neural network to classify facial expressions of the players.** Proceedings - Frontiers in Education Conference, FIE. Anais... , San Jose, USA, 2019.

PERVEEN, N.; GUPTA, S.; VERMA, K. **Facial expression recognition using facial characteristic points and Gini index.** 2012 Students Conference on Engineering and Systems, SCES 2012. Anais... Allahabad, Uttar Pradesh, India: IEE. p.1–6, 2012.

PHILLIPS, P. J.; FLYNN, P. J.; SCRUGGS, T.; et al. **Overview of the face recognition grand challenge.** Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005. Anais...**,** San Diego, CA, USA: IEEE. v. I, p.947–954, 2005.

PICARD, R. W. **Affective Computing**. Cambridge, MA, USA, 1995.

PORIA, S.; CAMBRIA, E.; BAJPAI, R.; HUSSAIN, A. A review of affective computing: From unimodal analysis to multimodal fusion. **Information Fusion** Elsevier, v. 37, n. September 2017, p. 98–125, 2017.

PUGLIESI, J. B.; SINOARA, R. A.; REZENDE, S. O. Combinação de Regressores Homogêneos e Heterogêneos: Precisão e Compreensibilidade. , 2003.

ROCHA, R. H. S. **Reconhecimento de Objetos por Redes Neurais Convolutivas**.

49 f. Dissertação - Programa de Pós-Graduação em Otimização e Raciocínio Automático, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2015.

RODRIGUEZ, P.; CUCURULL, G.; GONALEZ, J.; et al. Deep Pain: Exploiting Long Short-Term Memory Networks for Facial Expression Classification. **IEEE Transactions on Cybernetics**, 2017.

RUPP, K. CPU, GPU and MIC Hardware Characteristics over Time. Disponível em: <https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/>. Acesso em: 27/7/2019.

SAHLA, N. E. **A Deep Learning Prediction Model for Object Classification**. 58 f. Delft University of Technology, 2018.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **Annual Review in Automatic Programming**, 1969.

SANDBACH, G.; ZAFEIRIOU, S.; PANTIC, M. Local normal binary patterns for 3D facial action unit detection. **Proceedings - International Conference on Image Processing, ICIP**, v. 203143, p. 1813–1816, 2012.

SANTOS, E. M. **Teoria e Aplicação de Support Vector Machines à e Reconhecimento deAprendizagem Objetos Baseado na Aparência Eulanda Miranda dos Santos**. 121 f. Dissertação (Mestrado em Informática) - Pós-Graduação em Informática, Universidade Federal da Paraíba, 2002.

SAVRAN, A.; ALYÜZ, N.; DIBEKLIOĞLU, H.; et al. Bosphorus database for 3D face analysis. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 5372 LNCS, p. 47–56, 2008.

SAVRAN, A.; SANKUR, B. Non-rigid registration based model-free 3D facial expression recognition. **Computer Vision and Image Understanding**, v. 162, p. 146–165, 2017.

SCHMIDHUBER, J. Deep Learning in neural networks: An overview. **Neural Networks**, v. 61, p. 85–117, 2015.

SHAN, C.; GONG, S.; MCOWAN, P. W. Facial expression recognition based on Local Binary Patterns: A comprehensive study. **Image and Vision Computing**, v. 27, n. 6, p. 803–816, 2009.

SIDDIQUE, M. N. H.; TOKHI, M. O. **Training Neural Networks: Backpropagation vs. Genetic Algorithms.** Proceedings of the International Joint Conference on Neural Networks. Anais... Washington, DC, USA,: IEEE. p.2673–2678, 2001.

SILVA, L. P. E **Rastreamento Facial e Refinamento de Pontos Fiduciais 3D Baseado na Região do Nariz em Ambientes Não Controlados**. 80 f. Dissertação (Mestrado em Informática) - Programa de Pós-Graduação em Informática, Universidade Federal do Paraná, Curitiba, 2017.

SOLTANPOUR, S.; BOUFAMA, B.; JONATHAN WU, Q. M. A survey of local feature methods for 3D face recognition. **Pattern Recognition**, 2017.

STANSBURY, D. Model Selection: Underfitting, Overfitting, and the Bias-Variance Tradeoff. Disponível em: <https://theclevermachine.wordpress.com/2013/04/21/model-selection-underfitting-overfitting-and-the-bias-variance-tradeoff/>. .

STENROOS, O. **Object detection from images using convolutional neural networks**. 75 f. Dissertação (Mestrado em Ciência da Computação) - Master's Programme in Computer, Communication and Information Sciences, Aalto University, Espoo, Finlândia, 2017.

TANG, J.; ALELYANI, S.; LIU, H. Feature Selection for Classification: A Review. In: C. C. Aggarwal (Ed.); **Data Classification: Algorithms and Applications** Boca Raton, USA: CRC Press. v. 1, p.37, 2014.

TARNOWSKI, P.; KOŁODZIEJ, M.; MAJKOWSKI, A.; RAK, R. J. Emotion recognition using facial expressions. **Procedia Computer Science** Elsevier B.V., v. 108, p. 1175–1184, 2017.

TATIBANA, C. Y.; KAETSU, D. Y. Uma Introdução às Redes Neurais. Disponível em: <http://www.din.uem.br/ia/neurais/>. Acesso em: 22/10/2017.

TELI, M. N. **Dimensionality Reduction and Classification of Time Embedded Eeg Signals**. 63 f. Dissertação (Mestrado em Ciência da Computação) - Department of Computer Science, Colorado State University, Fort Collins, Estados Unidos, 2007.

TERVEN, J. R.; CÓRDOVA-ESPARZA, D. M. Kin2. A Kinect 2 toolbox for MATLAB. **Science of Computer Programming**, 2016.

THE TFEID PROJECT. The Taiwanese Facial Expression Image Database. Disponível em: <http://bml.ym.edu.tw/tfeid/>. Acesso em: 2/9/2018.

THIRUMURUGANATHAN, S. A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm. Disponível em: <https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/>. Acesso em: 19/5/2019.

TORRALBA, A.; EFROS, A. **Unbiased look at dataset bias.** Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Anais... Colorado Springs, CO, USA: IEEE. p.1521–1528, 2011.

TSALAKANIDOU, F.; MALASSIOTIS, S.; STRINTZIS, M. G. Face localization and authentication using color and depth images. **IEEE Transactions on Image Processing**, v. 14, n. 2, p. 152–168, 2005.

VIERIU, R. L.; TULYAKOV, S.; SEMENIUTA, S.; SANGINETO, E.; SEBE, N. **Facial expression recognition under a wide range of head poses.** 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015. Anais...**,** Ljubljana, Slovenia, 2015.

VILLANUEVA, W. J. P. **Comitê de Máquinas em Predição de Séries**. 178 f. Dissertação (Mestrado em Ciência da Computação) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2006.

VINCIARELLI, A.; PANTIC, M.; HEYLEN, D.; et al. Bridging the gap between social animal and unsocial machine: A survey of social signal processing. **IEEE Transactions on Affective Computing**, v. 3, n. 1, p. 69–87, 2012.

VRETOS, N.; NIKOLAIDIS, N.; PITAS, I. **3D facial expression recognition using Zernike moments on depth images.** Proceedings - International Conference on Image Processing, ICIP. Anais... Brussels, Belgium, 2011.

VUKADINOVIC, D.; PANTIC, M. **Fully Automatic Facial Feature Point Detection Using Gabor Feature Based Boosted Classifiers.** 2005 IEEE International

Conference on Systems, Man and Cybernetics. Anais... Waikoloa, HI, USA: IEEE. v. 2, p.1692–1698, 2005.

WEI, W.; JIA, Q.; CHEN, G. Real-time Facial Expression Recognition for Affective Computing Based on Kinect. , p. 161–165, 2016.

WU, J. **Introduction to convolutional neural networks**. 2017.

WU, N.; GERAS, K. J.; SHEN, Y.; et al. Breast density classification with deep convolutional neural networks. **NIPS**, 2017.

XU, C.; WANG, Y.; TAN, T.; QUAN, L. **Automatic 3D face recognition combining global geometric features with local shape variation information.** IEEE International Conference on Automatic Face and Gesture Recognition, FG. Anais... Seoul, South Korea: IEEE. p.308–313, 2004.

YAN, J.; ZHENG, W.; CUI, Z.; et al. **Multi-cue fusion for emotion recognition in the wild.** Proceedings of the 18th ACM International Conference on Multimodal Interaction. Anais... Tokyo, Japan. p.458–463, 2016.

YANG, D.; ALSADOON, A.; PRASAD, P. W. C.; SINGH, A. K.; ELCHOUEMI, A. An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment. **Procedia Computer Science** Elsevier B.V., v. 125, n. 2009, p. 2–10, 2018.

YANG, L.; ZHANG, L.; DONG, H.; ALELAIWI, A.; SADDIK, A. EL. Evaluating and improving the depth accuracy of Kinect for Windows v2. **IEEE Sensors Journal**, v. 15, n. 8, p. 4275–4285, 2015.

YIN, L.; WEI, X.; SUN, Y.; WANG, J.; ROSATO, M. J. A **3D facial expression database for facial behavior research.** FGR 2006: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition. Anais... Southampton, UK: IEEE. p.211–216, 2006.

ZAPLETAL, O. **Image Recognition by Convolutional Neural Networks - Basic Concepts**. 75 f. Dissertação (Mestrado em Engenharia Elétrica) - Department of Control and Instrumentation, Brno University of Technology, Brno, República Checa, 2017.

ZENG, Z.; PANTIC, M.; ROISMAN, G. I.; HUANG, T. S. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 31, n. 1, p. 39–58, 2009.

ZHANG, L.; TJONDRONEGORO, D.; CHANDRAN, V. Representation of facial expression categories in continuous arousal-valence space: Feature and correlation. **Image and Vision Computing**, v. 32, n. 12, p. 1067–1079, 2014.

ZHANG, S.; ZHAO, X.; LEI, B. Facial expression recognition based on local binary patterns and local fisher discriminant analysis. **WSEAS Transactions on Signal Processing**, v. 8, n. 1, p. 21–31, 2012.

ZHANG, Y.; ZHANG, L.; HOSSAIN, M. A. Adaptive 3D facial action intensity estimation and emotion recognition. **Expert Systems with Applications** Elsevier Ltd, v. 42, n. 3, p. 1446–1464, 2015.

ZHANG, Z.; CUI, L.; LIU, X.; ZHU, T. **Emotion detection using Kinect 3D facial points.** 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI). Anais..., Omaha, USA. p.407–410, 2016.

ZHAO, X.; DELLANDRÉA, E.; ZOU, J.; CHEN, L. A unified probabilistic framework for automatic 3D facial expression analysis based on a Bayesian belief inference and statistical feature models. **Image and Vision Computing** Elsevier B.V., v. 31, n. 3, p. 231–245, 2013.

ZHAO, X.; ZOU, J.; LI, H.; et al. Automatic 2.5-D Facial Landmarking and Emotion Annotation for Social Interaction Assistance. **IEEE Transactions on Cybernetics**, v. 46, n. 9, p. 2042–2055, 2016.

ZOHRA, F. T.; GAVRILOVA, M. KINECT Face Recognition Using Occluded Area Localization Method. **Trans. on Comput. Sci. XXX**, v. LNCS 10560, p. 12–28, 2017a.

ZOHRA, F. T.; GAVRILOVA, M. KINECT Face Recognition Using Occluded Area Localization Method. **Transactions on Computational Science XXX: Special Issue on Cyberworlds and Cybersecurity**, v. LNCS 10560, p. 12–28, 2017b.

ZOPPIS, I.; MAURI, G.; DONDI, R. Kernel Machines: Introduction. **Reference Module in Life Sciences**, , n. 2014, 2018.

ZUCKER, S. Differential geometry from the frenet point of view: Boundary detection, stereo, texture and color. **Handbook of Mathematical Models in Computer Vision** Boston, MA, USA: Springer, 2006.