UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE
DARCY RIBEIRO
Centro de Ciência e Tecnologia
Laboratório de Ciências Matemáticas

Larissa Ribeiro Sardinha

RECONHECIMENTO DE EMOÇÕES ATRAVÉS DA ANÁLISE SEQUENCIAL DE IMAGENS

Campos dos Goytacazes - RJ

2025

Larissa Ribeiro Sardinha

RECONHECIMENTO DE EMOÇÕES ATRAVÉS DA ANÁLISE SEQUENCIAL DE IMAGENS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Estadual do Norte Fluminense Darcy Ribeiro como requisito para a obtenção do título de Bacharel em Ciência da Computação, sob orientação de Prof. Luis Antonio Rivera Escriba.

Universidade Estadual do Norte Fluminense Darcy Ribeiro

Centro de Ciência e Tecnologia

Laboratório de Ciências Matemáticas

Ciência da Computação

Orientador: Prof. Dr. Luis Antonio Rivera Escriba

Campos dos Goytacazes - RJ

2025

Larissa Ribeiro Sardinha

RECONHECIMENTO DE EMOÇÕES ATRAVÉS DA ANÁLISE SEQUENCIAL DE IMAGENS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Estadual do Norte Fluminense Darcy Ribeiro como requisito para a obtenção do título de Bacharel em Ciência da Computação, sob orientação de Prof. Luis Antonio Rivera Escriba.

Prof. Dr. Luis Antonio Rivera Escriba
Orientador

Prof. Dr. João Luiz de Almeida Filho
Membro da Banca

Prof. Dr. Luis Alberto Rabanal Ramirez
Membro da Banca

Campos dos Goytacazes - RJ
2025

Este projeto é dedicado a Deus e à minha querida amiga Ediene Freitas Rodrigues da Costa (*in memoriam*).

Sem eles, este projeto não seria possível.

Agradecimentos

A Deus, pela minha vida e por me permitir superar todos os obstáculos encontrados ao longo da realização deste trabalho.

Aos meus pais, Uilis e Alailza, por sempre acreditarem em mim e por me apoiarem na conquista de espaços que eles próprios não tiveram a oportunidade de conquistar. Vocês são minha inspiração e minha base.

À minha irmã Rayssa, por compartilhar comigo as experiências ao longo dessa caminhada que chamamos de vida, e aos meus sobrinhos Sophia e Lucca, por serem o meu refúgio e o meu lar.

Aos amigos que esta universidade me proporcionou, em especial ao Jhonatan, Juliana, Nicole e Pablo, que dividiram comigo o dia a dia e a jornada acadêmica, sempre com muito café, sorrisos e ombros para os momentos de desânimo. Vocês tornaram essa caminhada mais leve e significativa.

Ao corpo docente, por compartilhar conhecimento e sabedoria, inspirando-me a buscar sempre mais. À universidade, por proporcionar a estrutura e o ambiente necessários para o meu crescimento acadêmico e pessoal, e ao seu corpo técnico, pelo apoio essencial em todas as etapas desta trajetória.

Por fim, mas não menos importante, à escola Leôncio Pereira Gomes e ao Instituto Federal Fluminense, que formaram a base do meu conhecimento.

"É curioso como não sei dizer quem sou. Quer dizer, sei-o bem, mas não posso dizer. Sobretudo tenho medo de dizer, porque no momento em que tento falar não só não exprimo o que sinto como o que sinto se transforma lentamente no que eu digo"

— Clarice Lispector, em Perto do Coração Selvagem.

Resumo

O reconhecimento de emoções por meio da análise sequencial de imagens tem se tornado uma ferramenta relevante em diversas áreas, como segurança, marketing e saúde, por auxiliar na compreensão dos estados emocionais de indivíduos. Nos últimos anos, observa-se um aumento no desenvolvimento de soluções baseadas em inteligência artificial para essa finalidade. Este trabalho propõe um sistema de reconhecimento automático de emoções, que utiliza técnicas de visão computacional e aprendizado de máquina. O processo é dividido em etapas, que incluem a detecção de faces em imagens, a extração de características visuais e a classificação das emoções. Para isso, são combinadas técnicas baseadas em descritores de imagens, como o Histograma de Gradientes Orientados (HOG), e modelos de redes neurais como as Redes Neurais Convolucionais(CNN) e as Memória de Curto Longo Prazo (LSTM), que permitem capturar tanto características espaciais quanto temporais das expressões faciais. Os resultados obtidos demonstram que o sistema é capaz de reconhecer emoções de forma eficiente, apresentando bom desempenho mesmo diante das variações naturais das expressões.

Palavras-chave: Reconhecimento de Emoções. Visão Computacional. Aprendizado de Máquina. HOG. CNN. LSTM.

Abstract

Emotion recognition through sequential image analysis has become a relevant tool in various fields, such as security, marketing, and healthcare, as it assists in understanding individuals' emotional states. In recent years, there has been a growing development of artificial intelligence-based solutions for this purpose. This work proposes an automatic emotion recognition system that employs computer vision and machine learning techniques. The process is divided into stages, which include face detection in images, visual feature extraction, and emotion classification. To achieve this, the system combines techniques based on image descriptors, such as the Histogram of Oriented Gradients (HOG), and neural network models, such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), which enable the capture of both spatial and temporal features of facial expressions. The results demonstrate that the system is capable of recognizing emotions efficiently, showing good performance even in the presence of natural variations in facial expressions.

Keywords: Emotion Recognition. Computer Vision. Machine Learning. HOG. CNN. LSTM.

Lista de ilustrações

Lista de trechos de código

Lista de abreviaturas e siglas

AU	Actions Unit - Unidade de Ação
BRRNs	Bidirectional Recurrent Neural Networks - Redes Neurais Recorrentes Bidirecionais
CNN	Convolutional Neural Network - Rede Neural Convolucional
DL	Deep Learning - Aprendizado Profundo
FACS	Facial Action Coding System - Sistema de Codificação de Ações Faciais
GANs	Generative Adversarial Networks - Redes Adversárias Generativas
GRU	Gated Recurrent Unit - Unidade Recorrente com Portas
HOG	Histogram of Oriented Gradients - Histograma de Gradientes Orientados
IA	Inteligência Artificial
KDEF-DYN	Karolinska Directed Emotional Faces - Dinâmico
LBP	Local Binary Pattern - Padrão Binário Local
LSTM	Long Short Term Memory - Memória de Curto Longo Prazo
ML	Machine Learning - Aprendizado de Máquina
MTCNN	Multi-Task Cascaded Convolutional Neural Network
RNAs	Redes Neurais Artificiais
RNN	Recurrent Neural Network - Rede Neural Recorrente
SVM	Support Vector Machine - Máquina de Vetores de Suporte

Sumário

1 Introdução

Vivemos em uma era altamente tecnológica, na qual tarefas cotidianas foram digitalizadas, visando facilitar a vida dos usuários, além de proporcionar melhor funcionamento, agilidade e precisão. Entre essas tarefas, destaca-se o reconhecimento de emoções, que busca interpretar expressões faciais humanas de forma automática, por meio da análise de imagens ou vídeos, emergindo assim como um campo de pesquisa fundamental e desafiador.

Entende-se como reconhecimento de emoções a capacidade de analisar visualmente a configuração e os movimentos de músculos faciais para identificar a emoção predominante em determinada expressão (WILHELM, 2014).

A detecção visual de movimentos musculares que definem uma emoção facial é possível através da análise de uma sequência de imagens que configure uma emoção conhecida. Assim, deve-se conhecer as variações das configurações que determinam uma emoção, o que constitui um processo complexo e muito difícil de ser realizada com exatidão sem o auxílio de ferramentas computacionais.

Nesse contexto, a visão computacional, aliada aos avanços em aprendizado profundo, oferece ferramentas poderosas para abordar essa complexidade. Técnicas de análise de imagem e vídeo permitem a detecção e classificação automatizada de expressões faciais, transformando dados visuais brutos em interpretações emocionais significativas (BHATT, 2020).

O reconhecimento de emoções faciais (*Facial Emotion Recognition – FER*) consolidou-se como uma subárea vital da visão computacional, focada na identificação de categorias emocionais a partir de dados faciais (CÎRNEANU; POPESCU; IORDACHE, 2023). Abordagens baseadas em aprendizado profundo, especialmente as que utilizam as redes neurais artificiais têm demonstrado grande sucesso.

Conhecemos como rede neural artificial um conjunto de neurônios artificiais que simulam a capacidade neural de raciocínio humano em máquinas. As redes neurais convolucionais (CNNs), um tipo de rede neural artificial, provaram ser particularmente eficazes na extração de características espaciais relevantes dos pixels da face, capturando padrões visuais complexos. Enquanto isso, as redes neurais recorrentes (RNNs), como LSTMs ou GRUs, são frequentemente empregadas para modelar a dinâmica temporal das expressões, crucial para a análise de vídeos (AWARI, 2023).

Contudo, a eficácia desses modelos depende da qualidade das características extraídas. A vasta quantidade de informação presente em cada quadro de vídeo, somada à variabilidade inerente às expressões, exige métodos robustos de caracterização para

garantir a precisão da identificação. Com o auxílio das técnicas de aprendizado profundo e do Histograma de Gradientes Orientados (HOG – Histogram of Oriented Gradients), é possível realizar a extração e classificação de características faciais sutis que correspondem a diferentes estados emocionais, proporcionando dessa forma uma base robusta para os sistemas de reconhecimento de emoções (ANIL; SURESH, 2023).

Apesar dos avanços, existem diversos fatores que dificultam o reconhecimento facial de emoções, incluindo variações de escala, localização e orientação, rotação dentro e fora do plano, expressões faciais, condições de iluminação, oclusões faciais, resolução da imagem e a quantidade de pessoas em uma mesma imagem (CÎRNEANU; POPESCU; IORDACHE, 2023). Além disso, questões como as diferenças culturais na expressão de emoções e a grande variedade de expressões faciais entre indivíduos dificultam a universalidade dos programas de identificação.

Dessa forma, torna-se imprescindível que os algoritmos empregados para realizar o reconhecimento facial de emoções sejam robustos e apresentem alta confiabilidade, especialmente em aplicações que demandam respostas precisas com base no estado emocional do usuário.

1.1 Formulação do Problema

O reconhecimento de emoções, por meio da análise facial, enfrenta diversos desafios, como a variabilidade das expressões faciais entre indivíduos, as diferenças culturais, assim como fatores externos, tais como iluminação, ângulo e oclusões, que comprometem a acurácia dos sistemas. Esses obstáculos impactam diretamente a capacidade dos algoritmos de identificar corretamente os estados emocionais. Diante da crescente demanda por sistemas mais precisos e confiáveis, em contextos como saúde, educação e marketing, torna-se necessário investigar estratégias computacionais que permitam aprimorar as métricas de desempenho no reconhecimento automático de emoções faciais.

1.2 Hipótese

A combinação de características extraídas pelo descritor HOG com as extraídas por uma rede neural convolucional (CNNs), quando utilizada para alimentar redes neurais recorrentes (RNNs) para classificação temporal, resulta em maior acurácia no reconhecimento de emoções faciais em vídeo. Acredita-se que essa abordagem captura as transições emocionais de forma mais eficaz do que métodos que dependem apenas de características de CNNs para a análise temporal.

1.3 Objetivos

Objetivo Geral: Desenvolver e avaliar um sistema de reconhecimento de emoções faciais em vídeo, empregando uma abordagem híbrida de extração de características HOG+CNN e uma classificação temporal utilizando uma RNN, visando assim aprimorar a precisão na identificação de expressões dinâmicas em comparação com métodos baseados apenas em CNN.

Objetivos Específicos:

1. Apresentar uma estrutura conceitual do reconhecimento facial de emoções.
2. Implementar uma etapa de pré-processamento para detecção facial e descritores HOG + uma rede neural convolucional com *transfer learning* para a extração de características faciais relevantes em cada frame.
3. Estabelecer um modelo eficiente de identificação das emoções mais comuns com base na análise sequencial de frames utilizando redes neurais recorrentes.
4. Avaliar quantitativamente o desempenho do sistema proposto (HOG+CNN+RNN) utilizando métricas padrão (acurácia, precisão, *F1-score* e matriz de confusão) em um *dataset* público.
5. Verificar o funcionamento do sistema proposto por meio de uma aplicação que permita observar a contagem de emoções em gráfico de barras, a evolução temporal das emoções e a distribuição de confiança das predições.

1.4 Justificação

A precisão e a confiabilidade em sistemas de reconhecimento de emoções possuem grande importância, visto que essa técnica traz melhorias para diversas aplicações, como a obtenção de informações sobre o comportamento e as preferências de clientes para o marketing, além de auxiliar profissionais de saúde a monitorar e diagnosticar problemas emocionais em pacientes.

Diante dessas limitações, a hipótese de que a combinação de descritores robustos, como HOG, com o poder de aprendizado das CNNs pode levar a uma representação de características mais completa e eficaz para a análise temporal com RNNs motiva este estudo, que busca contribuir para o avanço da área.

1.5 Método

A abordagem adotada neste trabalho baseia-se na combinação de redes neurais convolucionais (CNN) com redes neurais recorrentes (RNN), conforme proposta por (RANGULOV; FAHIM, 2021). Nesse método, a CNN é utilizada para a extração de características visuais relevantes em cada frame do vídeo, enquanto a RNN modela a sequência temporal dessas características para realizar a classificação das emoções.

Para o treinamento do sistema, será utilizado o banco de dados KDEF-DYN (KAROLINSKA INSTITUTET,), composto por vídeos de atores expressando diversas emoções faciais. Cada vídeo será segmentado em frames individuais, que passarão por um processo de pré-processamento e extração de características essenciais.

As características extraídas consistem na concatenação dos descritores obtidos pelo Histograma de Gradientes Orientados (HOG – Histogram of Oriented Gradients) e pelas camadas da CNN, formando um vetor único que proporciona uma representação mais robusta e completa das imagens faciais.

Este vetor resultante será então fornecido como entrada para uma rede LSTM (Long Short Term Memory), uma arquitetura de RNN especializada em modelar dependências temporais de longo prazo, que realizará a classificação final da emoção apresentada.

Por fim, o desempenho do sistema será avaliado quantitativamente com o uso de métricas padrão, tais como acurácia, precisão e F1-score, utilizando os dados do banco de referência para validação. Além disso, será verificado o funcionamento do sistema por meio de uma aplicação prática que permita observar o funcionamento do modelo.

1.6 Organização do Trabalho

A presente pesquisa é estruturada em 6 capítulos. O primeiro capítulo é introdutório, contextualizando o que é o reconhecimento facial de emoções e qual a sua importância. O segundo capítulo oferece um referencial teórico com revisão bibliográfica dos conceitos abordados. O terceiro capítulo apresenta o modelo proposto do sistema, com detalhes sobre sua estrutura e também sobre as tecnologias escolhidas para a implementação. O quarto capítulo apresenta o desenvolvimentodo modelo proposto. O quinto capítulo apresenta os resultados obtidos através da implementação. O sexto capítulo contém a conclusão da pesquisa e os trabalhos futuros. Ao final, é apresentado um Apêndice, que disponibiliza o link para o repositório do código-fonte desenvolvido, permitindo o acesso ao material utilizado e produzido durante a pesquisa, bem como informações adicionais para consulta.

2 RECONHECIMENTO FACIAL DE EMOÇÕES

As emoções são parte de nós, parte do nosso dia a dia e de nossa interação com a sociedade; mas, nem sempre é fácil identificá-las, principalmente quando queremos decifrar qual emoção o outro está sentindo e o que, exatamente, essa emoção significa. Porém, há algo que nos entrega: nossa expressão facial. O famoso dito popular "está escrito na sua face" é verdadeiro, pois, mesmo sem o nosso consentimento, diversas vezes, as nossas expressões faciais revelam o nosso estado emocional.

A detecção visual de variações das expressões faciais que caracterizam uma emoção exige o conhecimento de diferentes configurações expressivas, o que torna o processo mais complexo. Com o avanço das técnicas de visão computacional, esse desafio tem sido enfrentado com maior eficiência. Em um mundo altamente visual, onde imagens e vídeos são constantemente produzidos, cresce a demanda por soluções capazes de interpretar essas informações. Essa evolução foi impulsionada pelos avanços em técnicas de aprendizado de máquina e, particularmente, pelo desenvolvimento de arquiteturas de aprendizado profundo capazes de modelar as complexas relações entre as configurações faciais e os estados emocionais subjacentes.

A identificação de emoções por meio do reconhecimento facial é uma ferramenta de grande importância, exigindo atenção especial em seu desenvolvimento por estar inserida em uma área interdisciplinar que combina conhecimentos da psicologia, visão computacional e inteligência artificial.

2.1 Emoções

As emoções são manifestações que envolvem reações intensas e breves do organismo em resposta a um evento inesperado ou, por vezes, muito aguardado e fantasiado (BOCK; FURTADO; TEIXEIRA, 2008); são processos complexos que envolvem tanto aspectos psicológicos quanto fisiológicos.

Segundo (DAMÁSIO, 1994), as emoções são respostas automáticas do organismo a estímulos ambientais, mediadas por estruturas cerebrais como o hipotálamo e o córtex pré-frontal. Essas estruturas trabalham em conjunto para gerar respostas emocionais que podem ser observadas em nossas reações.

Essas reações são importantes para a liberação de tensão, mas muitas das vezes saem do controle humano, tornando assim quase impossível não transparecer fisicamente

quando algo nos afeta, mesmo quando o nosso desejo é simular uma neutralidade; assim, de acordo com (BOCK; FURTADO; TEIXEIRA, 2008), as nossas feições acabam nos traindo e demonstrando a emoção que tanto lutamos para esconder.

Não se sabe ao certo a quantidade de emoções existentes, mas é possível observar na literatura que muitos autores determinam um conjunto de emoções básicas, sendo elas: alegria, medo, surpresa, tristeza, nojo e raiva (MIGUEL, 2015). A figura 1 ilustra algumas dessas emoções.

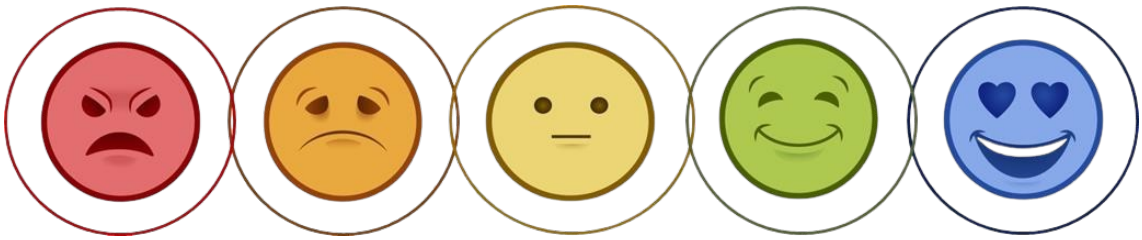

Figura 1 – Representação de Emoções (raiva, tristeza, calma, alegria, surpresa).

Como dito antes, embora o número exato de emoções ainda seja um tema em debate, a identificação e o reconhecimento dessas expressões faciais fundamentais têm sido um foco significativo em diversas áreas de estudo.

2.1.1 Expressões Faciais

Em 1976, surgiu uma forte teoria da identificação de emoções a partir da análise facial, com a publicação do artigo *Measuring Facial Movements* (em português, "Medindo o Movimento Facial") por (EKMAN; FRIESEN, 1976), em que os autores concluem que as expressões faciais são manifestações biológicas consistentes das emoções humanas.

Eles desenvolveram, então, um Sistema de Codificação de Ações Faciais (*Facial Action Coding System – FACS*), que classifica os movimentos musculares faciais possíveis em 46 Unidades de Ação (*Action Units – AUs*). Esse sistema é utilizado até hoje no treinamento e na avaliação de sistemas automáticos de reconhecimento de emoções.

A conclusão é a de que uma expressão emocional é resultado da junção de unidades de ações constituintes em um determinado grupo. Por tanto, uma expressão emocional é caracterizada através de uma série de movimentos.

As contrações musculares e movimentos de áreas do nosso rosto como olhos, pálpebras, testa, sobrancelhas e a parte inferior do rosto (ao redor da boca), contribuem principalmente para a manifestação dessas emoções, por tanto é possível detectar contrações através da análise de uma sequência de imagens (segmento de vídeo) relativo ao possível gesto (RANGULOV; FAHIM, 2021).

Podemos observar na Figura 2, uma representação de expressões faciais das seis emoções básicas (1. nojo, 2. medo, 3. alegria, 4. surpresa, 5. tristeza, 6. raiva).

Figura 2 – Representação de expressões faciais das seis emoções básicas SCHMIDT e COHN (2001).

A configuração facial de uma emoção é a união de várias movimentações musculares faciais subjacentes, que refletem nossos processos mentais e afetivos.

Para (CACIOPPO, 2000), as expressões faciais não surgem de forma abrupta, mas sim por meio de uma transição suave, na qual diferentes grupos musculares são ativados em sequência ou paralelamente; assim em alguns momentos a configuração muscular de uma emoção converge temporariamente com a de outra, criando expressões ambíguas ou mistas.

2.2 Visão Computacional e Reconhecimento Facial

A visão computacional é definida por (BASTOS; ESTEVES, 2021) como uma área da inteligência artificial que permite a interpretação automatizada de imagens e vídeos, possibilitando assim a identificação de objetos a partir de pixels de imagens brutas.

Conforme definido por (SHANMUGAMANI, 2018), este campo engloba técnicas e algoritmos que permitem a extração, análise e compreensão de informações significativas a partir de imagens e vídeos. Assim, o principal objetivo da visão computacional é "fazer com que as máquinas vejam o mundo da mesma forma que os humanos" (BHATT, 2020, p. 1-2). Essa técnica desempenha um importante papel no processo de reconhecimento facial, pois possibilita a detecção facial automática em imagens, identificando e realizando a extração de características faciais que possibilitam a análise da emoção.

Inicialmente, o reconhecimento facial por meio da visão computacional surgiu como uma técnica voltada à identificação de indivíduos com base em seus traços faciais, utilizando bancos de dados para autenticação de identidade (BASTOS; ESTEVES, 2021). A partir dessa aplicação, desenvolveu-se o reconhecimento automatizado de emoções faciais, baseado na análise algorítmica de gestos e expressões humanas sequenciais com o objetivo de identificar sinais emocionais (CÎRNEANU; POPESCU; IORDACHE, 2023).

O processo do reconhecimento de emoções a partir da análise facial com visão computacional possui um certo conjunto de etapas para garantir a sua eficácia, as principais são: captura dos dados, pré-processamento, extração de características e a classificação final, ilustradas na Figura 3.

Figura 3 – Etapas para o reconhecimento de uma emoção.

Para que o processo ocorra de maneira eficiente, o classificador deve ser treinado previamente com imagens já rotuladas, assim irá detectar padrões e aprender a reconhecer a emoção expressada. A Figura 4 ilustra o processo de reconhecimento de emoção facial automático da seguinte forma:

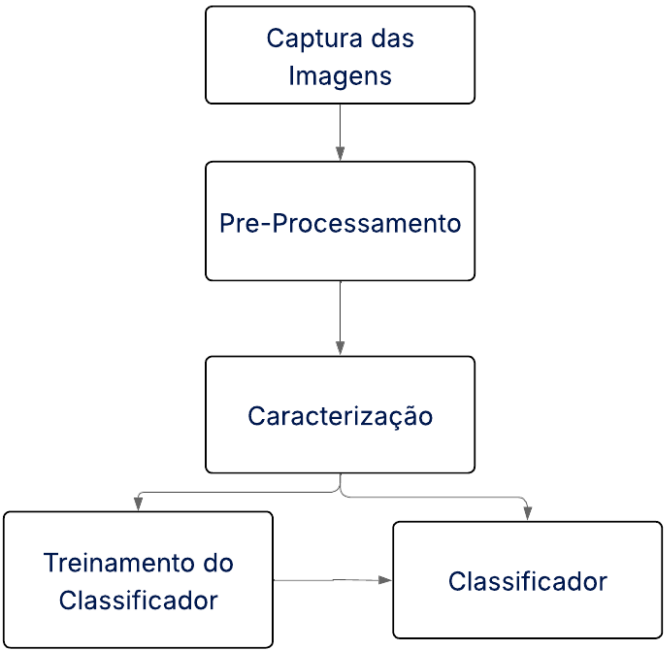

Figura 4 – Processo de Reconhecimento Facial Automático.

Dessa forma, para entender o processo completo de um reconhecedor facial automático, é necessária uma revisão de todas as suas etapas.

2.2.1 Captura das Imagens

O processo inicial de um sistema de visão computacional é a captura do dado, ou seja, a obtenção de imagens ou vídeos. Essa captação é realizada através de sensores como câmeras digitais, smartphones, webcam ou até mesmo através da busca em bancos de dados específicos para cada aplicação.

Para a detecção sequencial de emoções, o dado a ser capturado é um vídeo, que nada mais é do que uma série de imagens exibidas sequencialmente. Dessa forma, toda a série deve ser capturada e analisada de forma individual. Essas imagens individuais, também conhecidas como frames, contêm dados que devem ser processados quadro por quadro (Figura 5), permitindo assim a detecção e a interpretação de padrões visuais dinâmicos (GONZÁLEZ; WOODS, 2010).

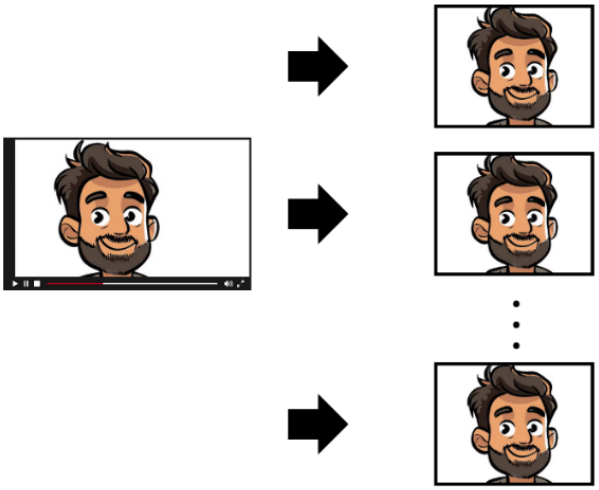

Figura 5 – Divisão de vídeo em quadros (imagens).

Quanto maior a quantidade de frames extraídos, maior a quantidade de informação. Além disso, fatores como a qualidade da imagem e nitidez são extremamente importantes pois influenciam diretamente as próximas etapas. Dessa forma, aspectos como resolução, taxa de quadros, iluminação, focalização e ruído devem ser considerados para garantir a integridade dos dados visuais coletados.

2.2.2 Pré-Processamento

A próxima etapa é o pré-processamento, onde os dados brutos são preparados para a análise. Para isso, aplica-se um conjunto de ações visando facilitar a interpretação das características, como a aplicação de filtros para a eliminação de ruídos, a detecção da área de interesse, o redimensionamento das imagens para um tamanho padrão, a rotação, entre outros. Além disso, os filtros podem ser aplicados com o objetivo de alterar o nível de intensidade de cores, a luminosidade ou até mesmo a quantidade de canais de cores,

visto que as informações necessárias na aplicação de reconhecimento facial de emoções independem das informações dadas pelas cores.

Outra ação é a detecção facial, ou seja, o processo de identificar e localizar rostos humanos em imagens ou vídeos, que é um dos pilares de todo esse processo. Os algoritmos criados para este fim buscam isolar a face para assim utiliza-la, dessa forma eles possuem um mesmo objetivo (Figura 6), mas os processos realizados são diferentes.

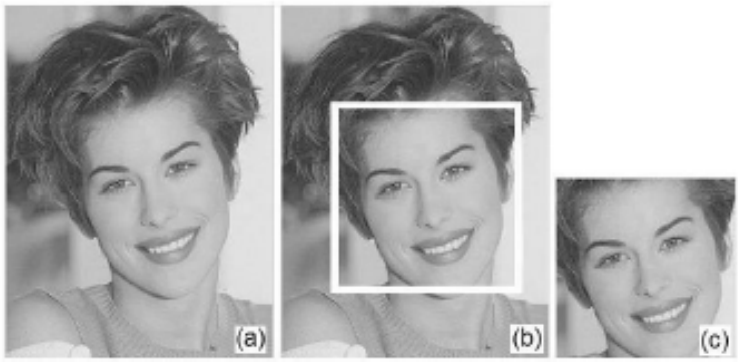

Figura 6 – Detecção da face. (a) Imagem de entrada; (b) face detectada; (c) face isolada (Rowley; Baluja; Kanade, 1998).

O método de detecção facial empregado por (RANGULOV; FAHIM, 2021) é o classificador Haar Cascade. Esse algoritmo utiliza características básicas que analisam a diferença de intensidade entre regiões adjacentes da imagem, somando os valores dos pixels em áreas retangulares contíguas, com base na luminosidade dos pixels, demonstrado visualmente na Figura 7.

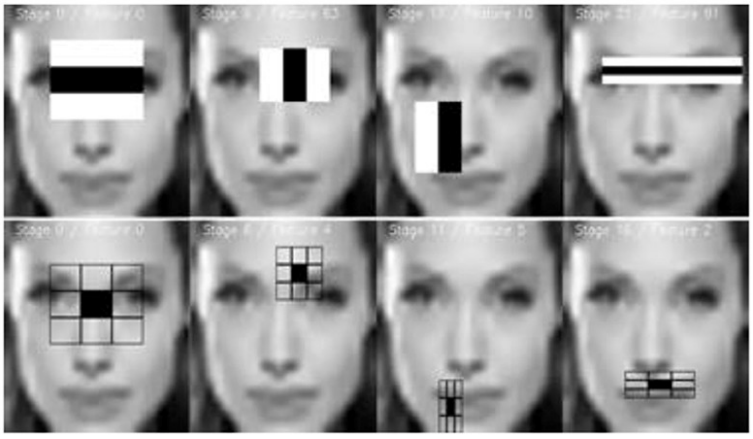

Figura 7 – Método Haar Cascade (RANGULOV; FAHIM, 2021).

Outro algoritmo de detecção utilizado por (BALLESTEROS *et al.*, 2024) é o MTCNN (*Multi-task Cascaded Convolutional Networks*), uma abordagem baseada em redes neurais convolucionais organizadas em cascata que realiza a detecção de rostos e pontos faciais com alta precisão. O MTCNN consiste em três etapas sequenciais: a *Proposal*

Network (P-Net), que gera regiões candidatas a conter rostos; a *Refine Networ*k (R-Net), que filtra falsos positivos e ajusta as caixas delimitadoras; e a *Output Networ*k (O-Net), que aprimora a detecção e identifica pontos faciais-chave como olhos, nariz e boca.

2.2.3 Caracterização

A caracterização é o processo de extrair informações relevantes ou atributos representativos de uma imagem para facilitar sua análise e reconhecimento.

Consiste basicamente em identificar e analisar detalhes específicos da face — como contornos, texturas, gradientes de intensidade e formas — que ajudam a diferenciar uma expressão facial ou uma emoção. Essas características extraídas funcionam como uma "impressão digital" da imagem, permitindo que algoritmos posteriores possam classificar ou reconhecer o que está representado na imagem com precisão. Técnicas comuns para caracterização incluem o Histograma de Gradiente Orientado (HOG), o LBP (Local Binary Patterns), entre outros.

a) Histograma de Gradientes Orientados

O histograma de gradientes orientados é calculado pela distribuição das intensidades e direções dos gradientes, ou seja, através das mudanças de intensidade da imagem, detectando assim padrões como bordas e contornos (ANIL; SURESH, 2023). As etapas do HOG podem ser vistas detalhadamente na Figura 8:

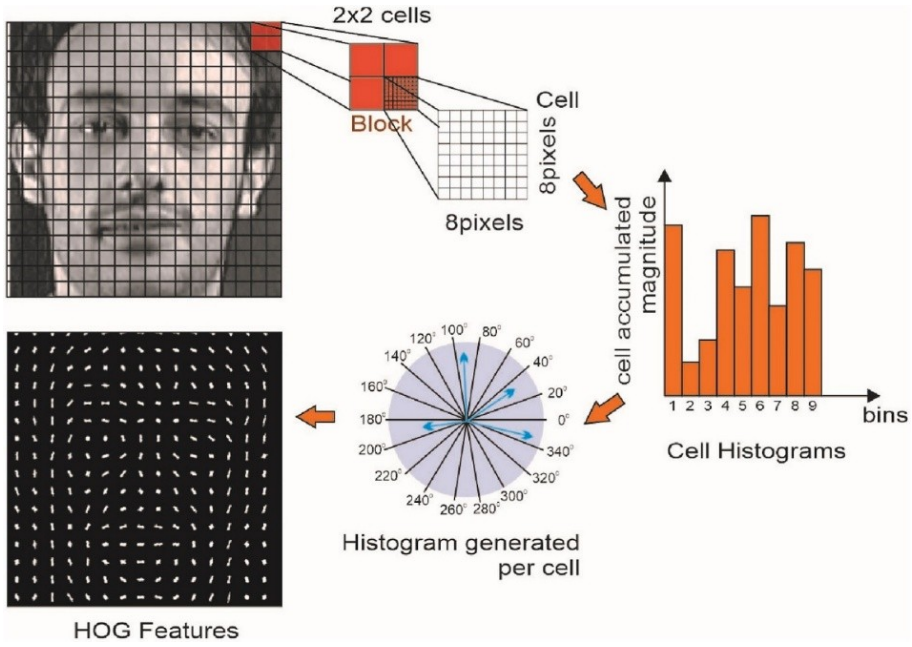

Figura 8 – Processo de um Histograma de Gradiente Orientado (ANIL; SURESH, 2023).

A imagem de entrada é redimensionada para um tamanho padrão e convertida para tons de cinza. Em seguida, a imagem é dividida em pequenas células, e para cada

pixel são calculadas a magnitude e a direção do gradiente, que geralmente é quantizada em ângulos discretos como 0°, 45°, 90° e 135°.

Após isso é calculado um histograma para cada célula, que representa a distribuição das orientações dos gradientes, evidenciando as bordas presentes. Para minimizar os efeitos de variações na iluminação, os histogramas são normalizados, dividindo-se cada histograma pela norma do conjunto de histogramas da imagem.

Por fim, os histogramas normalizados de todas as células são concatenados em um único vetor de características que representa o objeto contido na imagem.

b) Padrão Binário Local

O Padrão Binário Local (LBP) realiza a caracterização ao comparar cada pixel com seus vizinhos para converter as informações de textura em padrões binários.

O resultado forma números binários de 8 bits para cada pixel, que são transformados em um histograma representando as frequências dos padrões locais. Esse histograma funciona como um vetor de características, utilizado na classificação de imagens (SEDAGHATJOO; HOSSEINZADEH; BIGHAM, 2024). Na Figura 9 é possível observar como o algoritmo LBP funciona.

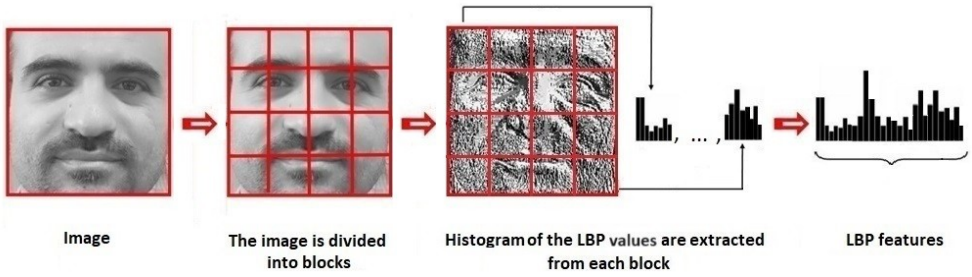

Figura 9 – Processo de Funcionamento do LBP (SEDAGHATJOO; HOSSEINZADEH; BIGHAM, 2024).

Dessa forma, o LBP se apresenta como uma técnica robusta e de baixo custo computacional para tarefas de classificação baseadas em textura.

2.2.4 Classificação

Após extrair as características é necessário identifica-las, ou seja, realizar a classificação; nesta etapa utiliza-se de algoritmos de aprendizado de máquina e redes neurais para categorizar as expressões faciais nas respectivas emoções.

Os métodos mais populares são as redes neurais artificiais e as máquinas de vetores de suporte.

a) Redes Neurais Artificiais

As redes neurais artificiais são capazes de aprender representações complexas dos dados através de um processo de treinamento supervisionado ou não supervisionado (Goodfellow; Bengio; Courville, 2016). Dentre os tipos mais conhecidos, destacam-se as redes neurais convolucionais (CNNs), utilizadas principalmente para reconhecimento de padrões espaciais em imagens, e as redes neurais recorrentes (RNNs), voltadas para o processamento de sequências temporais, como vídeos.

b) Máquina de Vetores de Suporte

As máquinas de vetores de suporte (SVM: *Support Vector Machine*) são algoritmos de aprendizado supervisionado muito eficazes para classificação e análise de padrões. Ele busca encontrar o hiperplano que separa de melhor forma as classes de dados, maximizando a margem entre elas (CORTES; VAPNIK, 1995).

2.2.5 Treinamento do Classificador

Nesta fase, o classificador é alimentado com um conjunto de dados rotulado, ou seja, cada amostra de entrada está associada à sua respectiva classe. O treinamento consiste em ajustar os parâmetros do modelo para minimizar o erro entre a predição realizada e o rótulo verdadeiro.

Durante o treinamento, o conjunto de dados é dividido em subconjuntos de treino e validação, o primeiro é usado para ajustar o modelo e o segundo para monitorar o desempenho e evitar que o modelo se adapte demais e pare de analisar. Após o treinamento, o classificador é avaliado com um conjunto de teste separado, verificando assim sua capacidade.

2.3 Aprendizado de Máquina e Aprendizado Profundo

O aprendizado de máquina (ML: *Machine Learning*) é a utilização de um extenso conjunto de dados para melhorar a capacidade de análise de dados, realizando dessa forma a identificação de padrões, para assim tomar uma decisão (Goodfellow; Bengio; Courville, 2016).

O reconhecimento de padrões por meio do aprendizado de máquina consolidou-se como uma das técnicas mais avançadas na atualidade, uma vez que, segundo (PRATEEK, 2017), essa abordagem permite a realização de previsões com base em dados que seriam desconhecidos e até mesmo imperceptíveis ao ser humano.

Com o aumento constante da quantidade de dados disponíveis, surgiu o aprendizado profundo (DL: *Deep Learning*), que possui a capacidade de processar grandes volumes de dados e que se beneficia do poder e da velocidade computacional das máquinas modernas (CÎRNEANU; POPESCU; IORDACHE, 2023), promovendo uma evolução exponencial

da inteligência artificial e potencializando todos os seus campos de aplicação. Dentre as técnicas mais importantes do ML e do DL estão as redes neurais artificiais (ANNs), que de acordo com (ALVES, 2020), baseiam-se na arquitetura dos neurônios humanos, buscando assim reproduzir o aprendizado através do desenvolvimento de sistemas que aprendem ao serem treinadas com uma base de dados de treino.

Em sua estrutura (Figura 10) é possível ver arranjos em nós, semelhantes a corpos celulares humanos, que se articulam com outros nós por meio de conexões programadas, ponderadas com base em sua capacidade de fornecer um resultado esperado (Goodfellow; Bengio; Courville, 2016).

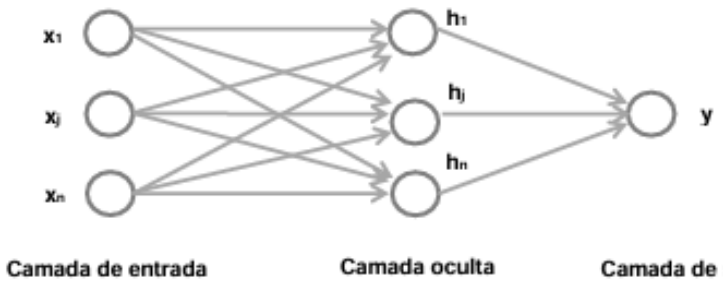

Figura 10 – Desenho de uma rede neural básica (SHANMUGAMANI, 2018).

Existem diversos tipos de redes neurais artificiais, com propostas e tamanhos variados, variando até mesmo o número de neurônios na camada de entrada e na camada oculta. Para esta pesquisa, é necessário analisarmos a estrutura de dois tipos de redes neurais: a rede neural convolucional e a rede neural recorrente.

2.3.1 Redes Neurais Convolucionais - CNNs

A rede neural convolucional (CNN) é uma arquitetura de rede neural artificial que incorpora operações de convoluções em diferentes camadas da rede. Esta rede é amplamente utilizada para tarefas de processamento de imagens e reconhecimento de padrões visuais (AWARI, 2023). As CNNs são projetadas para capturar padrões espaciais dentro de matrizes de entrada multidimensionais (CÎRNEANU; POPESCU; IORDACHE, 2023). Para isso, elas são treinadas através do uso de um conjunto de dados rotulados, onde cada entrada (por exemplo, uma imagem) está associada a uma saída desejada (por exemplo, a classe do objeto na imagem), sendo denominada como uma rede *Feedforward*, ou seja, a entrada considera apenas sua entrada atual, as entradas anteriores não são úteis durante o processo, perdendo assim a noção de ordem no tempo.

Uma das camadas mais importantes de uma CNN é a camada de convolução, onde após transformar a imagem de entrada em uma matriz de valores, são aplicados filtros (ou *kernels*) para detectar padrões como cantos, bordas ou texturas, resultando em mapas de característica. Esses mapas de características são passados por uma camada de ativação

ReLU para introduzir não linearidade e aprender padrões mais complexos, facilitando assim a extração de características.

A Figura 11 ilustra uma rede CNN com um número **n** de camadas de convolução para extrair características locais, seguidas de funções de ativação como ReLU para introduzir não linearidade; após isso passa para camada de *pooling* para reduzir a dimensão dos mapas de características, tornando o processamento mais eficiente.

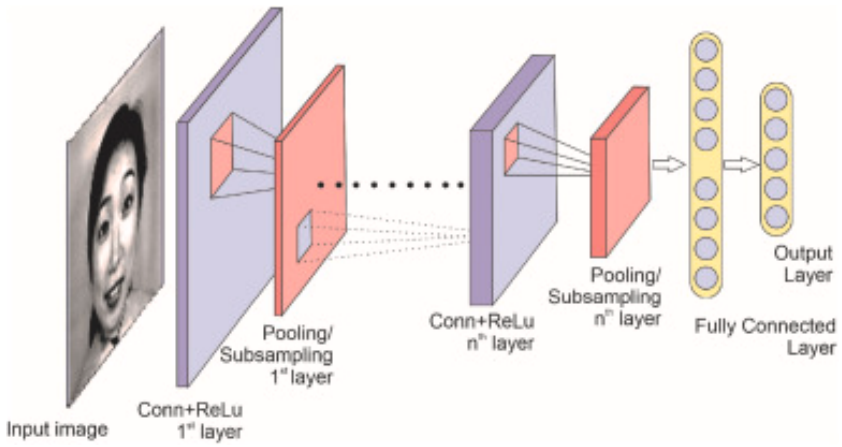

Figura 11 – Camadas de uma Rede Neural Convolucional (ANIL; SURESH, 2023).

Depois desse processo, as camadas totalmente conectadas geram pontuações para cada classe, que são convertidas em probabilidades, gerando assim o resultado da classificação como saída.

2.3.2 Redes Neurais Recorrentes - RNNs

De acordo com (HANAFI; BOUHORMA; LOFTI, 2021), uma rede neural recorrente (RNN) é um tipo de rede neural projetada para lidar com dados sequenciais, como séries temporais, textos, imagens e vídeos, em que a ordem dos elementos é importante. Isto é possível pois as RNNs possuem conexões recorrentes, ou seja, a saída de uma célula da rede pode ser reutilizada como entrada para a mesma célula nos próximos processamentos (Figura 12).

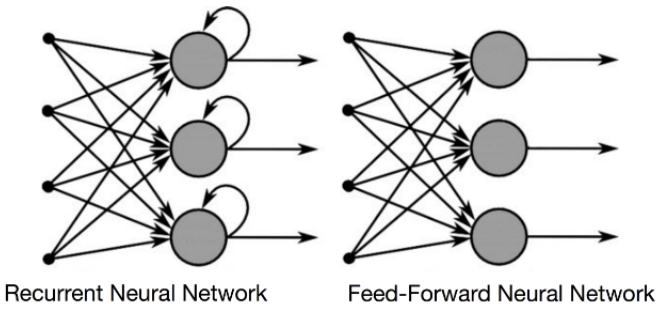

Figura 12 – Redes Neurais Recorrentes e Feed-Forward (HANAFI; BOUHORMA; LOFTI, 2021).

Um dos problemas das RNNs é que sua memória é rápida e curta, dessa forma ela lembra apenas o que aconteceu nas ultimas interações; assim surge a Memória de curto longo prazo, conhecida como LSTM (*Long Short Term Memory*) (RANGULOV; FAHIM, 2021).

A LSTM é um tipo de rede neural recorrente desenvolvida para solucionar o problema das dependências de longo prazo, visto que consegue "lembrar"de informações por longos períodos e processar até mesmo sequências de imagens (CÎRNEANU; POPESCU; IORDACHE, 2023).

Além da LSTM, existem outros tipos de redes neurais recorrentes como as redes neurais recorrentes bidirecionais (BRRNs), as unidades recorrentes fechadas (GNUs), as RNN com codificador-decodificador e etc (HANAFI; BOUHORMA; LOFTI, 2021).

2.3.3 Transferência de Aprendizado (*Transfer Learning*)

Uma das técnicas de aprendizado de máquina muito utilizada para lidar com uma baixa quantidade de dados é a transferência de aprendizado (CÎRNEANU; POPESCU; IORDACHE, 2023). Assim, se utiliza o conhecimento adquirido de uma rede já treinada com outros dados, aplicando-os em uma nova tarefa, melhorando assim a generalização; dessa forma as redes são treinadas não apenas para uma tarefa específica, mas de uma forma que possa ser aplicada em diferentes tarefas (OLIVAS *et al.*, 2009).

Na Figura 13 podemos observar a diferença entre métodos de aprendizado tradicionais, criados para tarefas específicas, e métodos de transferência de aprendizado, que podem ser aplicados a diferentes tarefas.

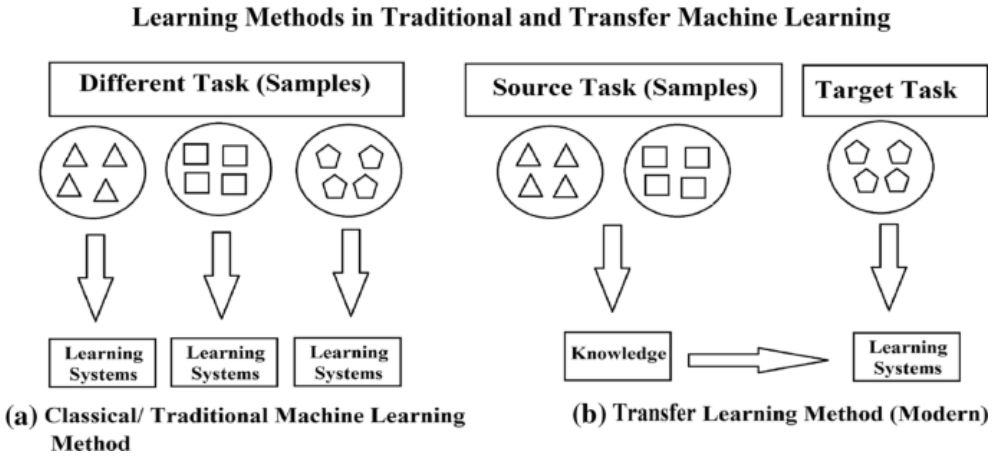

Figura 13 – Aprendizado de Máquina e Aprendizado por transferência (OLIVAS *et al.*, 2009).

Além de eliminar a necessidade de grandes conjuntos de dados de treinamento, a transferência de aprendizado acaba reduzindo o custo computacional ao reutilizar um modelo já existente para resolver um novo problema, aproveitando as características e os pesos já aprendidos pelps modelos, treinados geralmente com grandes datasets.

O uso de redes neurais convolucionais pré-treinadas têm sido muito explorados nos últimos anos, pois dessa forma as redes já aprenderam a identificar padrões gerais em imagens (CÎRNEANU; POPESCU; IORDACHE, 2023). Exemplos notórios de arquiteturas frequentemente utilizadas para a transferência de aprendizado incluem a VGG-16, a ConvNet e a MobileNetV2 (BALLESTEROS *et al.*, 2024).

2.4 Trabalhos Relacionados

O reconhecimento facial de emoções é um campo de estudo que tem recebido crescente atenção devido às suas amplas aplicações em áreas como interação humano-computador, segurança e bem-estar emocional.

No artigo (EKMAN; FRIESEN, 1976), os autores desenvolvem o Sistema de Codificação da Ação Facial (*Facial Action Coding System - FACS*), um método detalhado para descrever movimentos faciais específicos associados às expressões emocionais. A pesquisa de Ekman e Friesen foi fundamental para estabelecer uma base científica rigorosa para a interpretação das expressões faciais, ajudando a identificar padrões universais de expressão emocional que são reconhecidos globalmente.

A introdução das redes neurais convolucionais (CNNs) trouxe um avanço significativo na precisão do reconhecimento facial de emoções. (TANG, 2013), utilizou CNNs para reconhecimento de emoções em imagens faciais, e demonstrou a superioridade dessas redes em benchmarks como o FER-2013. Em (ANIL; SURESH, 2023) é desenvolvido uma metodologia híbrida para reconhecimento facial, que combina o HOG e a CNN para

extração e refinamento de características, seguido por classificação com um classificador não iterativo como o KELM. Essa técnica, denominada HOG-CKELM, alia o rápido tempo de treinamento do KELM à capacidade da CNN de extrair características profundas e invariantes, oferecendo alta precisão e rapidez quando comparada à abordagem HOG-CNN tradicional.

O artigo (CÎRNEANU; POPESCU; IORDACHE, 2023) aborda uma visão abrangente das tendências recentes no reconhecimento de emoções utilizando redes neurais e análise de imagens. O artigo destaca a importância crescente desse campo em diversas aplicações, como educação, saúde e segurança pública. Eles comparam CNNs com outras arquiteturas, como redes neurais recorrentes (RNNs) e redes adversárias generativas (GANs), destacando os elementos-chave, desempenho, vantagens e limitações de cada modelo, além de concluir que o uso de aprendizado por transferência e o desenvolvimento de arquiteturas mais eficientes possam melhorar acurácia. O artigo (BALLESTEROS *et al.*, 2024) tem como objetivo a criação de um software para reconhecer a emoção facial expressada, para isso utiliza técnicas de visão computacional e de inteligência artificial. O artigo utiliza um método de detecção facial para identificar a região de interesse e após isso, utiliza 2 redes neurais convolucionais, uma rede pré treinada, utilizada através do transfer learning, para a extração de características e uma nova rede neural convolucional criada para a classificação.

Em (HUANG *et al.*, 2023), técnicas de visão computacional foram utilizadas para identificar pontos de referência faciais importantes para a detecção da emoção facial. Com esse propósito, foram feitos experimentos de validação com duas redes neurais já criadas, fazendo o uso do tranfer learning, destacando assim a importância de usar a aprendizagem por transferência para melhorar o desempenho dos algoritmos. O estudo realizado por (SARVAKAR *et al.*, 2023) propõe o desenvolvimento e implementação de um sistema de reconhecimento facial com base em redes neurais convolucionais (CNN). A escolha da arquitetura correta permitiu obter uma boa acurácia na classificação de rostos. Os autores apontam que ajustes, como a redução da taxa de aprendizado e a simplificação da arquitetura, podem aumentar a acurácia e reduzir o custo computacional e o tempo de treinamento.

Em (RANGULOV; FAHIM, 2021) é proposta uma abordagem híbrida para reconhecimento de emoções em vídeos, combinando redes convolucionais (CNNs) como extratoras de características e redes neurais recorrentes (RNNs) para modelar a dinâmica temporal. O método foi testado em um grande conjunto de dados de vídeo, com resultados promissores, superando abordagens tradicionais que consideram apenas quadros estáticos. A combinação do extrator convolucional com a RNN permite capturar tanto o conteúdo visual quanto a evolução temporal, essencial para o reconhecimento robusto de emoções.

3 Sistema de Reconhecimento Facial de Emoções

O sistema de reconhecimento facial de emoções proposto busca analisar sequências de expressões faciais através de vídeos, visto que os vídeos na verdade são sequências de imagens(frames). Cada frame do vídeo deve ser analisado para detectar expressões faciais, ou seja, após a coleta, cada frame da sequência de expressão facial da emoção passa para as etapas de modo individual. Assim, é implementado um pipeline completo que analisa os vídeos, processa vinte frames selecionados e detecta a emoção presente em cada um deles. Este pipeline integra múltiplas técnicas de visão computacional e aprendizado profundo, organizadas em quatro etapas principais: pré-processamento, extração de características, classificação e análise de resultados.

A modelagem proposta na presente pesquisa é definida e detalhada na Figura 14, contendo assim as etapas necessárias e as técnicas utilizadas.

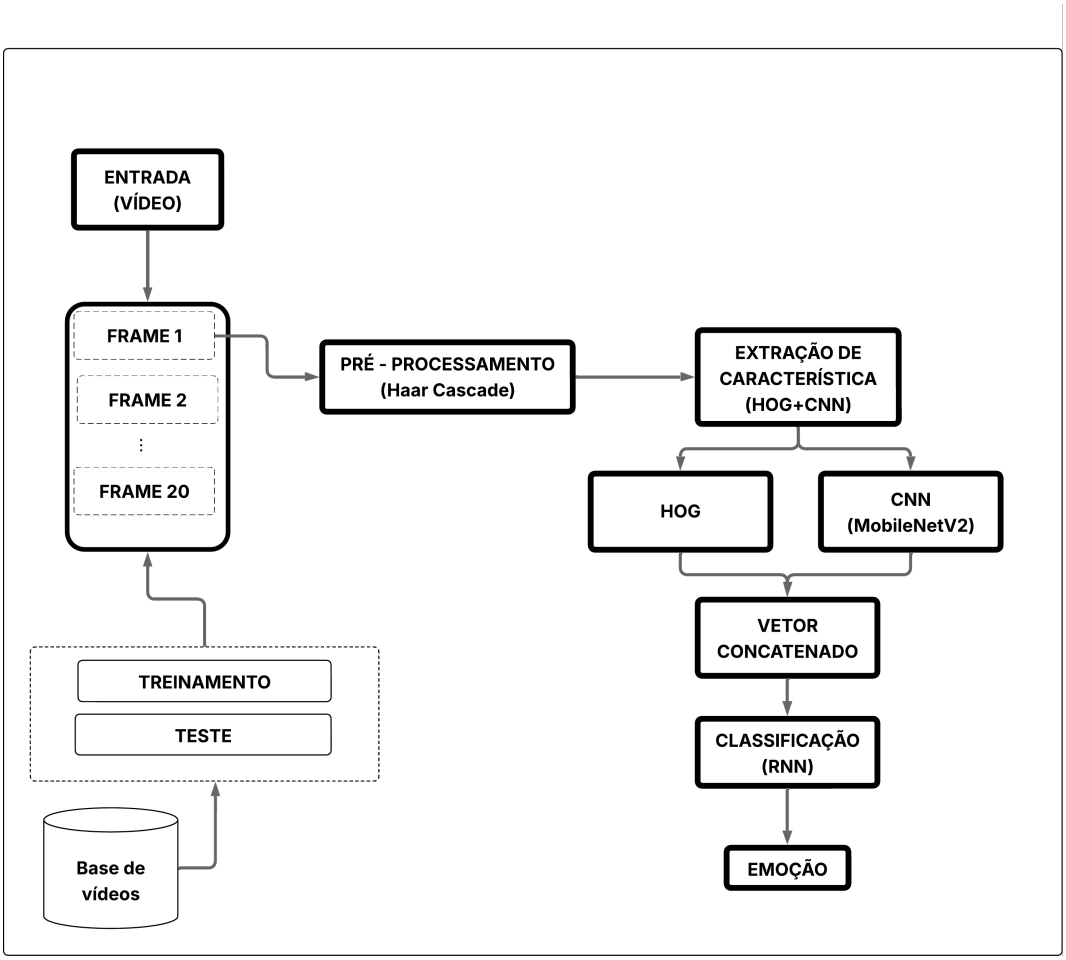

Figura 14 – Modelagem do Sistema Proposto.

Dessa forma, o sistema é organizado em uma sequência lógica de etapas, permitindo assim a análise eficiente das expressões faciais ao longo do tempo. Cada etapa será detalhada individualmente, destacando os métodos e algoritmos adotados

3.1 Captura da Imagem

As imagens/frames utilizadas serão divididos em três conjuntos nesse sistema: para treinamento, validação e teste.

Para isso, utiliza-se um banco de dados de vídeo faciais, o KDEF-dyn (*Karolinska Directed Emotional Faces – dynamic version*) (KAROLINSKA INSTITUTET,), composto por 240 vídeos que retratam atores expressando seis emoções (alegria, tristeza, raiva, medo, nojo e surpresa) de forma dinâmica, o que é fundamental para capturar as transições e nuances temporais das expressões faciais, demonstrando assim a evolução de um estado neutro para uma das emoções básica em vídeos curtos, com duração média de 1,033ms cada.

Cada um dos 240 vídeos será dividido em 20 frames, totalizando 4.800 frames que serão analisados em sequência, considerando os blocos de 20 frames por vez. Como dito antes, os dados foram separados em três conjuntos: 60% para treino, 20% para teste e 20% para validação. Essa divisão permite que o classificador aprenda a partir dos dados de treino e seja avaliado quanto ao desempenho com os dados de teste e validação.

Os vídeos serão processados respeitando a ordem temporal dos frames, garantindo que a sequência das expressões faciais seja mantida para análise. Após o treinamento, o modelo será testado em vídeos gravados especificamente para avaliar seu funcionamento, seguindo o mesmo procedimento de divisão em blocos sequenciais de 20 frames para classificação

3.2 Pré-Processamento das Imagens

No pré-processamento são aplicadas técnicas específicas para otimizar a qualidade e a eficiência do processamento dos frames de vídeo. Estas técnicas são fundamentais para garantir a precisão e o desempenho do sistema de reconhecimento facial.

Os frames são redimensionados para um formato padronizado de 48×48 pixels, garantindo consistência dimensional para o modelo e reduzindo significativamente a carga computacional.

A transformação de imagens coloridas em RGB para a escala de cinza reduz a dimensionalidade dos dados preservando as informações essenciais de iluminação que são críticas para a detecção de características faciais. Além disso, é necessário utilizar algoritmos

para identificar a região de interesse, ou seja, identificar a região facial, isolando-a de todo o resto. Um exemplo de algoritmo com esse objetivo é o Haar Cascade.

3.2.1 Haar Cascade

O Haar Cascade é um método que visa detectar objetos ou regiões de interesse, através do aprendizado de máquina (VIOLA; JONES, 2001). Assim, o algoritmo é treinado a partir de imagens que contém o objeto a ser identificado e também com imagens que não possuem o objeto.

O seu funcionamento consiste nas seguintes fases:

a) Características Haar

São realizados cálculos em regiões retangulares adjacentes em um local específico da imagem, visando buscar uma mudança de intensidade. Dessa forma, as características de haar são calculadas através da diferença da soma de pixel sob o retângulo branco e sob o retângulo preto (BALLESTEROS *et al.*, 2024), como podemos observar na Equação 3.1.

$$\text{Valor da característica} = \sum(\text{pixels na região branca}) - \sum(\text{pixels na região preta}) \tag{3.1}$$

Estas características são particularmente eficazes para detectar bordas, linhas e estruturas centro-circundantes que são comuns em faces humanas (Lienhart; Maydt, 2002). A Figura 15 ilustra os principais tipos de características Haar.

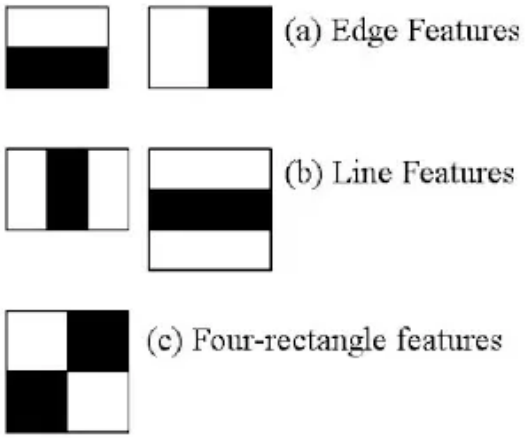

Figura 15 – Tipos de características de Haar (Implementation. . . ,).

Capturando assim características essenciais para a detecção da região de interesse, visto que a região dos olhos geralmente é mais escura que as bochechas, ou a ponte nasal é mais clara que os olhos adjacentes (BALLESTEROS *et al.*, 2024).

b) Imagem Integral

O processo de calcular cada retângulo pode ser longo, assim utiliza-se uma imagem intermediária para diminuir o tempo de processamento. Essa imagem intermediária é na verdade a imagem integral (BALLESTEROS *et al.*, 2024), dividida normalmente em quatro partes, como na Figura 16.

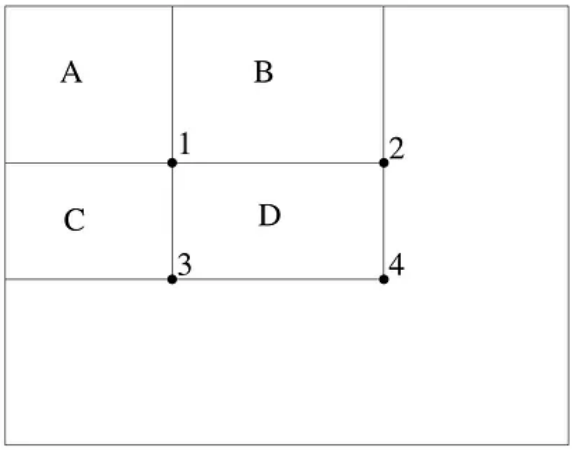

Figura 16 – Integral de uma imagem (VIOLA; JONES, 2001).

Dessa forma, a imagem integral na posição (x,y) é a soma dos pixels acima e à esquerda de (x,y), como podemos ver na Equação 3.2:

$$ii(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} i(x', y') \tag{3.2}$$

Onde:

ii(x, y)) representa a imagem integral na coordenada (x,y);

i(x', y') é imagem original.representa o valor da imagem original no ponto (x', y');

Reduzindo assim o grau de complexidade ao lidar com as imagens, calculando a soma dos pixels em qualquer região retangular usando apenas quatro referências na imagem integral.

c) Treinamento Adaboost

No treinamento Adaboost, apenas os melhores recursos obtidos são escolhidos e direcionados para diversos classificadores que são treinados separadamente, focando assim em corrigir os erros do classificador anterior (BALLESTEROS *et al.*, 2024).

A cada iteração, o algoritmo seleciona o classificador e as características que apresentam a menor taxa de erro. Quando um classificador identifica uma região na imagem, essa região é passada para o próximo classificador, que foca em outra característica.

Cada classificador se concentra em uma única característica, e o processo se repete a cada iteração, formando assim um classificador final mais robusto, como ilustrado na Figura 17.

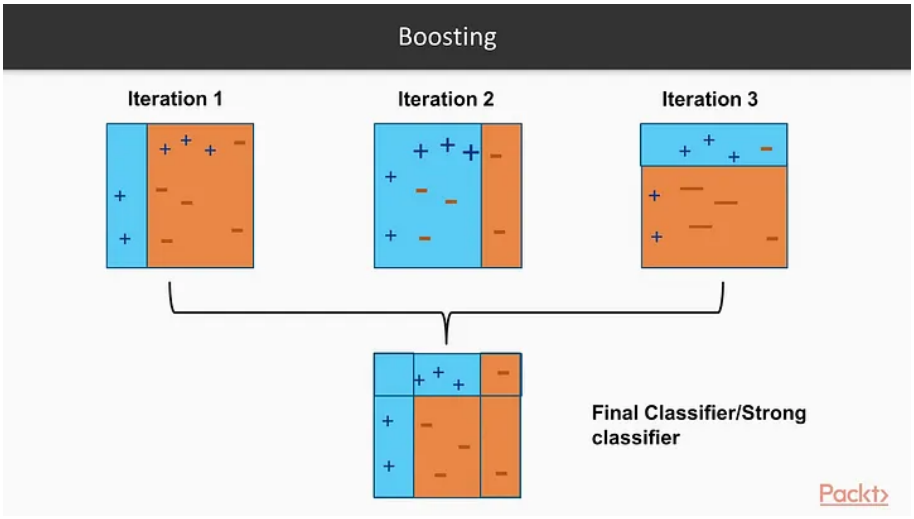

Figura 17 – Algoritmo Ada Boost (BALLESTEROS *et al.*, 2024).

Além disso, o método combina a aplicação dessas características com um classificador em cascata e o uso de uma janela deslizante para identificar rostos de forma eficiente (RANGULOV; FAHIM, 2021). A biblioteca OpenCV disponibiliza classificadores Haar Cascade pré-treinados para diversas aplicações, incluindo detecção facial, basta fornecer o caminho do arquivo e chamar a função correspondente para utilizá-los.

3.3 Extração de Características

A extração de características é a etapa onde apenas informações relevantes são extraídas das imagens faciais e tudo o que não é útil para a análise é descartado. No modelo proposto, utilizaremos a concatenação de dois recursos de extração: um extrator HOG e a CNN pré-treinada, a MobileNetV2.

3.3.1 Histograma de Gradiente Orientado (HOG)

Uma técnica de extração de características muito conhecida é o Histograma de Gradientes Orientados (HOG), que calcula os gradientes locais de intensidade da imagem e gera histogramas que descrevem as direções predominantes das bordas, criando assim uma representação robusta das estruturas faciais (ANIL; SURESH, 2023).

Segundo (SHU; DING; FANG, 2011), o processo de extração do HOG pode ser dividido nas seguintes etapas:

a) Cálculo do Gradiente

São calculados os gradientes horizontal e vertical de cada pixel da imagem através da aplicação de máscaras como o operador Sobel, muito utilizado para detecção de bordas em imagens ao identificar as variações na direção horizontal(Equação 3.3) e na direção vertical(Equação 3.4) de forma individual.

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \tag{3.3}$$

$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \tag{3.4}$$

Após obter os gradientes horizontal e vertical de cada pixel, podemos calcular a magnitude e a orientação do gradiente. A magnitude de cada gradiente é calculada através da raiz quadrada da soma dos quadrados do gradiente horizontal e do gradiente vertical através da Equação 3.5 abaixo:

$$G = \sqrt{g_x^2 + g_y^2} \tag{3.5}$$

A orientação do gradiente, ou seja, o seu ângulo, é calculada através do arco tangente do gradiente vertical ao gradiente horizontal, como podemos ver na Equação 3.6.

$$\theta = \arctan\left(\frac{g_y}{g_x}\right) \tag{3.6}$$

b) Divisão em Células

A imagem é dividida em pequenas regiões conectadas, denominadas "células". Para cada célula, cria-se um histograma das orientações dos gradientes. As orientações dos pixels dentro da célula calculadas anteriormente são quantizadas em um número pré-definido de "bins", ou seja, são definidos em categorias angulares. Cada pixel contribui para um bin do histograma com um peso proporcional à magnitude do seu gradiente. Dessa forma, a distribuição das orientações das bordas dentro da célula é obtida a partir do histograma.

c) Agrupamento em Blocos e Normalização

Para aumentar a robustez a variações locais de iluminação e contraste, as células são agrupadas espacialmente em blocos maiores e sobrepostos. Um vetor de características é formado pela concatenação dos histogramas de todas as células dentro de um bloco. Este vetor concatenado é então normalizado por exemplo, usando a norma L2, Equação 3.7.

$$\vec{v} = \frac{\vec{v}}{\|\vec{v}\|} \tag{3.7}$$

d) Formação do Vetor de Características

Finalmente, os vetores de características normalizados de todos os blocos da imagem são concatenados para formar o descritor HOG final, que representa a distribuição das orientações dos gradientes na região analisada. Este descritor HOG resultante pode então ser utilizado como entrada para algoritmos de aprendizado de máquina para tarefas de classificação e detecção de objetos.

3.3.2 MobileNetV2

A MobileNetV2 é uma rede neural Convolucional (CNN) desenvolvida por pesquisadores do Google como uma nova versão da MobileNet, criada para lidar com aplicações de visão computacional em dispositivos com poucos recursos de memória e processamento, além de sua agilidade (SHARMA, 2023).

Ela se destaca por seu equilíbrio entre eficiência computacional e acurácia na classificação de imagens. Sua estrutura possui as camadas ilustradas na Figura 18:

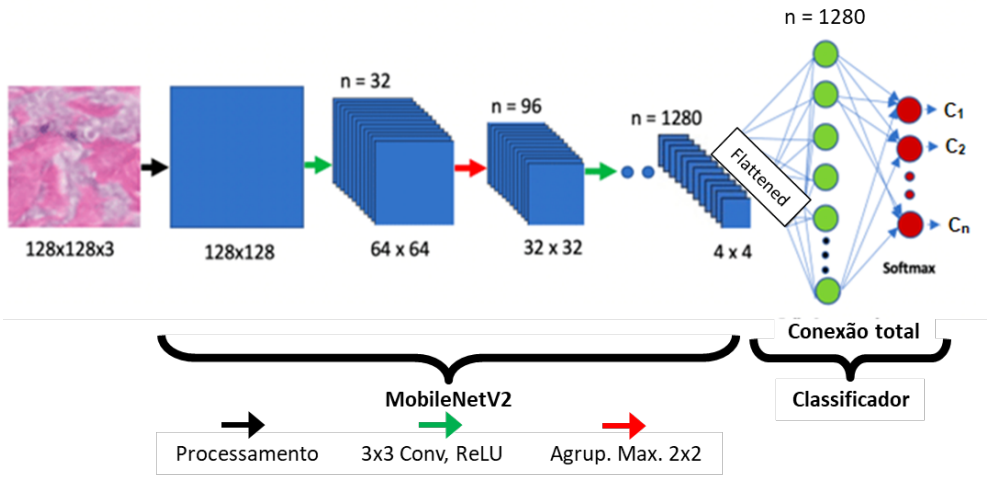

Figura 18 – Estrutura da MobileNetV2 (SHARMA, 2023).

Entre seus principais componentes estão as convoluções separáveis em profundidade, que dividem a operação padrão em duas etapas, a convolução em profundidade e a convolução pontual, diminuindo significativamente os cálculos. A estrutura também utiliza resíduos invertidos e projetos de gargalo, que expandem e comprimem os canais de forma estratégica, permitindo ao modelo extrair características mais complexas com menos parâmetros (SANDLER *et al.*, 2019).

Outro diferencial é o uso de gargalos lineares, que evitam a perda de informações ao empregar ativações lineares ao final dos blocos, e dos blocos SE (*Squeeze-and-Excitation*), que recalibram os canais para destacar os recursos mais relevantes. Esses elementos tornam o MobileNetV2 altamente eficaz para tarefas de visão computacional, mantendo

um equilíbrio entre desempenho e leveza (SANDLER *et al.*, 2019). Após essa camada Convolucional, ocorre uma camada de *Max Pooling* para agregar as características espaciais. Depois de passar por várias sequencias de Convolução + Max Pool, é gerado um vetor de característica que é então direcionado para a camada de classificação.

Esse modelo é muito utilizado em diversas aplicações através do aprendizado por transferência (*transfer learning*). Assim, os pesos pré-treinados com o conjunto de dados *ImageNet* são utilizados no novo modelo, aproveitando assim os conhecimentos obtidos durante o processamento do dataset original. A camada de classificação original é descartada, dessa forma será possível obter apenas o vetor de características gerado.

É realizado um ajuste para que o modelo se adeque a nova tarefa, adaptando as camadas finais ou novamente treinando de maneira progressiva mais camadas para otimizar o desempenho no dataset alvo (SHARMA, 2023).

3.4 Classificação

O vetor concatenado de características gerado pela extração do HOG e da MobileNetV2, será direcionado para a camada de entrada de uma rede neural recorrente, mais especificadamente uma LSTM.

A *Long Short-Term Memory(LSTM)*, em português Memória de curto longo prazo, é uma rede neural recorrente capaz de manter e atualizar informações por um período de tempo. Sua estrutura é composta por uma camada de entrada, diversas camadas ocultas e uma camada de saída, porém em suas camadas ocultas estão diversas unidades de memória, também conhecidas como células. Essas células são compostas por três portas principais: a porta de entrada(*Input Gate*), a porta de esquecimento(*Forget Gate*) e a porta de saída(*Output Gate*). São essas portas que controlam o fluxo de informações dentro da célula de memória (Figura 19).

Essa estrutura complexa de portões permite que a LSTM controle o fluxo de informações dentro da célula de memória, aprendendo quais informações são importantes para manter, quais devem ser esquecidas e quais devem ser usadas para fazer previsões ou classificações em cada ponto da sequência. Isso a torna particularmente eficaz em tarefas onde o contexto de longo prazo é essencial (Basiri *et al.*, 2021).

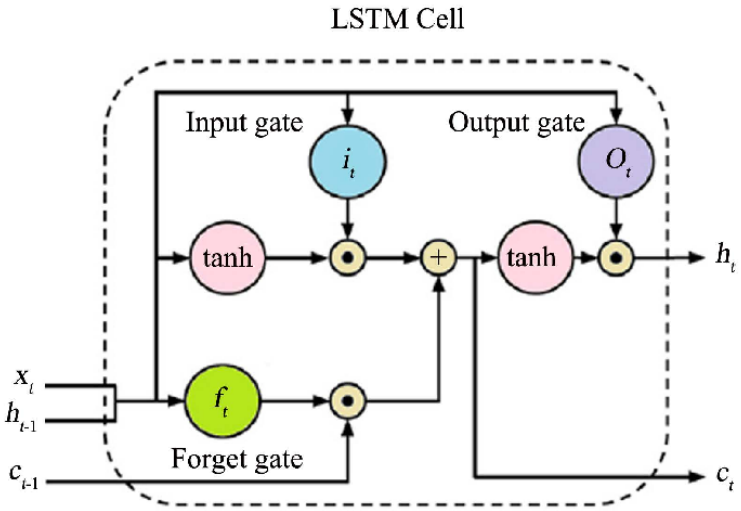

Figura 19 – Estrutura de uma LSTM (Basiri *et al.*, 2021).

Dessa forma a LSTM permite que o sistema identifique as características não apenas um único frame, mas também como eles evoluem ao longo do tempo, o que é crucial para a detecção de emoções em vídeos.

3.5 Ferramentas e Ambiente de Desenvolvimento

O sistema foi desenvolvido na linguagem Python, com ênfase nas bibliotecas TensorFlow e Keras para a implementação das redes neurais, incluindo a arquitetura MobileNetV2, além de Numpy e OpenCV para o pré-processamento das imagens e a detecção facial por meio do Haar Cascade. O treinamento do modelo foi executado em uma máquina com GPU, a fim de acelerar o processo.

4 Implementação

Este capítulo descreve em detalhes a implementação do sistema de reconhecimento de emoções faciais através da análise de imagem proposto neste trabalho, assim como os resultados experimentais obtidos. As etapas incluem a aquisição e preparação dos dados, o pré-processamento das imagens e a extração de características, o processo de treinamento e, por fim, a avaliação e análise dos resultados alcançados.

4.1 Captura dos Vídeos

A etapa inicial da implementação consistiu na obtenção e preparação do conjunto de dados para treinar e avaliar o modelo de reconhecimento de emoções. Para este propósito, utilizou-se o banco de dados KDEF-dyn (*Karolinska Directed Emotional Faces – dynamic version*) (KAROLINSKA INSTITUTET,) como fonte. Este *dataset* é composto por 240 vídeos que retratam atores expressando seis emoções de forma dinâmica, o que é fundamental para capturar as transições e nuances temporais das expressões faciais, demonstrando assim a evolução de um estado neutro para uma das emoções básicas.

Cada um desses vídeos foi processado (Código 4.1) para a extração de 20 frames espaçados, transformando os 240 vídeos em 240 sequências, cada uma composta por 20 frames, totalizando 4.800 frames para análise

```
#Captura dos frames:
            frame_count = 0
            capturados_temp = {}
            while cap.isOpened() and frame_count < total_frames_video:
                ret, frame = cap.read()
                if not ret:
                    break
                if frame_count in indices_frames:
                    capturados_temp[frame_count] = frame
                frame_count += 1
            cap.release()
```

Lista de código 4.1 – Sequencia de captura dos frames de um vídeo.

O conjunto de dados é dividido (Código 4.2) em 60% para o treinamento(2880 imagens, 144 sequências), 20% para a validação (960 imagens, 48 sequências) e 20% para o teste (960 imagens, 48 sequências), garantindo uma separação adequada entre as fases de aprendizado e validação do modelo, buscando evitar o *overfitting* - quando o modelo se adapta demais aos dados que viu e não consegue lidar com novas informações que não estavam no conjunto de treinamento.

```
    # Mapeamento de rótulos para garantir estratificação correta
```

```
    map_rotulos_unicos = {label: i for i, label in enumerate(np.unique(y_seq))}
    y_seq_mapped = np.array([map_rotulos_unicos[label] for label in y_seq])

    try:
        val_test_size = VALIDATION_SIZE + TEST_SIZE # Tamanho combinado para Val
+Teste
        train_size = 1.0 - val_test_size

        # Divide em Treino e (Val + Teste)
        X_treino_seq, X_val_teste_seq, y_treino_seq, y_val_teste_seq =
train_test_split(
            X_seq, y_seq_mapped, train_size=train_size, random_state=42,
stratify=y_seq_mapped
        )

        # Calcula a proporção de teste relativa ao conjunto (Val + Teste)
        test_size_relative = TEST_SIZE / val_test_size

        # Divide (Val + Teste) em Validação e Teste
        X_val_seq, X_teste_seq, y_val_seq, y_teste_seq = train_test_split(
            X_val_teste_seq, y_val_teste_seq, test_size=test_size_relative,
random_state=42, stratify=y_val_teste_seq
        )
```

Lista de código 4.2 – Divisão de sequências de frames para treino

O parâmetro *stratify=rotulos_videos* garante que 20% das sequências de cada uma das emoções irá para o conjunto de teste e de validação, evitando assim que uma emoção fique desbalanceada no conjunto de teste.

4.2 Pré-Processamento

Como dito antes, para cada vídeo do dataset, são extraídos uma sequência fixa de 20 frames por vídeo. Desta forma, é necessário que cada frame extraído passe por um pré-processamento básico essencial: conversão para escala de cinza, eliminando informações de cor e focando nas variações de intensidade, e o redimensionado para um tamanho padrão de 48x48 pixels. Esse processo é descrito no Código 4.3.

Além disso, é necessário utilizar o *Haar Cascade*, para realizar a detecção facial determinando a região de interesse.

```
            frames_selecionados_para_video = []
            for idx_target in indices_frames:
                if idx_target in capturados_temp:
                    frame_to_use = capturados_temp[idx_target]
                    frame_cinza = cv2.cvtColor(frame_to_use, cv2.COLOR_BGR2GRAY)
                    faces = face_cascade.detectMultiScale(frame_cinza,
scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
                    if len(faces) > 0:
                        (x, y, w, h) = faces[0]
                        face_roi = frame_cinza[y:y+h, x:x+w]
```

Lista de código 4.3 – Implementação do pré-processamento dos frames.

Essa padronização é necessária para garantir que todas as entradas para os extratores de características tenham dimensões consistentes.

4.3 Extração de Características

A etapa seguinte é a extração de características. O sistema implementado emprega uma abordagem híbrida, combinando dois métodos para capturar diferentes níveis de informação da imagem: o Histograma de Gradientes Orientados (HOG) e uma rede neural convolucional (CNN) pré-treinada (MobileNetV2).

4.3.1 Histograma de Gradientes Orientados

A extração de características HOG é realizada para cada frame já pré-processado, além disso é aplicado um filtro Gaussian Blur, com o objetivo de suavizar a imagem e reduzir o ruído. O processo(Código 4.4) se inicia com o cálculo dos gradientes de intensidade nas direções horizontal e vertical utilizando o operador Sobel e a partir destes, a magnitude e a orientação do gradiente são computadas para cada pixel.

```
    def calcularGradiente(imagem):
        imagem = cv2.GaussianBlur(imagem.astype(np.float32), (3, 3), 0)
        gradienteX = cv2.Sobel(imagem, cv2.CV_32F, 1, 0, ksize=1)
        gradienteY = cv2.Sobel(imagem, cv2.CV_32F, 0, 1, ksize=1)
        magnitude = np.sqrt(gradienteX**2 + gradienteY**2)
        angulo = np.arctan2(gradienteY, gradienteX) * (180 / np.pi)
        angulo[angulo < 0] += 180
        return magnitude, angulo
```

Lista de código 4.4 – Processo de extração de características por HOG.

A imagem é então dividida em células de 6x6 pixels. Dentro de cada célula, um histograma das orientações dos gradientes (distribuídas em 12 bins) é construído, ponderado pelas magnitudes dos gradientes, processo descrito no Código 4.5.

```
    def calcularHistogramaCelula(imagens, tamanho_imagem=(48,48), tamanho_bloco
    =2, tamanho_celula=6, bins=12):
        hog_vetores = []
        for img in tqdm(imagens, desc="Custom HOG Features"):
            if img.shape != tamanho_imagem:
                img = cv2.resize(img, tamanho_imagem, interpolation=cv2.
    INTER_AREA)
            img = img.astype(np.float32)

            magnitude, angulo = HOGFeatureExtractor.calcularGradiente(img)

            altura, largura = img.shape
            celula_x = largura // tamanho_celula
            celula_y = altura // tamanho_celula

            histograma_orientacoes = np.zeros((celula_y, celula_x, bins))

```

```
            for i in range(celula_y):
                for j in range(celula_x):
                    mag_celula = magnitude[i*tamanho_celula:(i+1)*tamanho_celula
    , j*tamanho_celula:(j+1)*tamanho_celula]
                    ang_celula = angulo[i*tamanho_celula:(i+1)*tamanho_celula, j
    *tamanho_celula:(j+1)*tamanho_celula]

                    hist = np.zeros(bins)
                    for y_cell in range(tamanho_celula):
                        for x_cell in range(tamanho_celula):
                            bin_idx = int(ang_celula[y_cell, x_cell] / (180.0 /
    bins)) % bins if bins > 0 else 0
                            hist[bin_idx] += mag_celula[y_cell, x_cell]
                    histograma_orientacoes[i, j] = hist
```

Lista de código 4.5 – Implementação do cálculo do histograma por célula.

Em seguida (Código 4.6), os blocos de células (2x2) são formados, deslizando sobre a imagem com sobreposição. Os histogramas das células dentro de cada bloco são concatenados e normalizados utilizando a técnica L2-Hys (normalização L2, seguida de clipping a 0.2 e re-normalização L2) para conferir robustez a variações de iluminação.

```
            blocos_y = celula_y - tamanho_bloco + 1
            blocos_x = celula_x - tamanho_bloco + 1
            blocos_normalizados = []

            for y_block in range(blocos_y):
                for x_block in range(blocos_x):
                    bloco = histograma_orientacoes[y_block:y_block+tamanho_bloco
    , x_block:x_block+tamanho_bloco].flatten()
                    norm = np.linalg.norm(bloco) + 1e-6
                    bloco_normalizado = bloco / norm
                    bloco_normalizado = np.minimum(bloco_normalizado, 0.2) #
    Clipping L2-Hys
                    norm = np.linalg.norm(bloco_normalizado) + 1e-6
                    bloco_normalizado = bloco_normalizado / norm
                    blocos_normalizados.append(bloco_normalizado)

            vetor_hog = np.concatenate(blocos_normalizados) if
    blocos_normalizados else np.array([])
            hog_vetores.append(vetor_hog)
```

Lista de código 4.6 – Sobreposição de frames para histograma.

Finalmente, os vetores normalizados de todos os blocos são concatenados para formar o descritor HOG final para aquele frame. Este descritor captura informações sobre as formas e texturas locais presentes na expressão facial.

4.3.2 Características via CNN (MobileNetV2)

Paralelamente à extração HOG, as características de mais alto nível foram extraídas utilizando a arquitetura MobileNetV2, pré-treinada na vasta base de dados *ImageNet* (Código 4.7).

```
# ==================== Extrator CNN (MobileNetV2) ====================
mobilenet = None
def inicializar_mobilenet(tamanho_imagem):
    global mobilenet
    if mobilenet is None:
        input_shape_tf = (tamanho_imagem[1], tamanho_imagem[0], 3)
        mobilenet = MobileNetV2(weights='imagenet', include_top=False, pooling='avg',
                                input_shape=input_shape_tf)
        mobilenet.trainable = False
        print(f"MobileNetV2 inicializado com input_shape={input_shape_tf}")
```

Lista de código 4.7 – Implementação da Inicialização do Extrator CNN (MobileNetV2).

A função inicializar_mobilenet é responsável por carregar o modelo *MobileNetV2* através da biblioteca Keras, configurando-o para atuar como um extrator de características: a camada de classificação final é removida (include_top=False), uma camada de *Global Average Pooling* é adicionada (*pooling*='avg') para obter um vetor de características de tamanho fixo, e os pesos das camadas convolucionais são congelados (mobilenet.trainable = False), caracterizando assim que o modelo será utilizado apenas através do *transfer learning*.

A função extrair_cnn (Código 4.8) processa os frames em escala de cinza, convertendo-os para um formato RGB ao replicar o canal de cinza três vezes e aplicando o pré-processamento específico da *MobileNetV2*(*preprocess_input*).

```
def extrair_cnn(imagens_gray):
    global mobilenet

    if imagens_gray.ndim == 2:
        imagens_gray = np.expand_dims(imagens_gray, axis=0)
    if imagens_gray.ndim == 3:
        imagens_gray = np.expand_dims(imagens_gray, axis=-1)

    if imagens_gray.shape[-1] == 1:
        imgs_rgb = np.repeat(imagens_gray, 3, axis=-1)
    elif imagens_gray.shape[-1] == 3:
        imgs_rgb = imagens_gray
    else:
        raise ValueError(f"Formato de imagem inesperado para CNN: {imagens_gray.shape}")

    imgs_rgb = imgs_rgb.astype(np.float32)
    imgs_rgb = preprocess_input(imgs_rgb)
)
    extracao = mobilenet.predict(imgs_rgb, batch_size=32, verbose=0)
```

Lista de código 4.8 – Implementação do Extrator CNN.

Os frames processados são enviados ao modelo MobileNetV2, e os vetores de características resultantes da camada de *pooling* são coletados.

4.3.3 Combinação de Características

A etapa final da extração consiste em combinar as informações obtidas pelos dois métodos (Código 4.9). Para cada frame individual, o vetor de características HOG e o vetor de características CNN são concatenados.

```
    # --- Combinação de Features ---
    print(f"Concatenando HOG ({features_hog.shape}) e CNN ({features_cnn.shape})
     features...")
    try:
        features_combinadas = np.concatenate([features_hog, features_cnn], axis
    =1)
    except ValueError as e:
        print(f"\nErro LSTM ao concatenar features: {e}. Shapes: HOG={
    features_hog.shape}, CNN={features_cnn.shape}")
        return None, None, None
    print(f"Shape features combinadas (antes de reshape): {features_combinadas.
    shape}")
```

Lista de código 4.9 – Implementação da criação do vetor de característica combinado.

O resultado é um único vetor de características combinado por frame, que integra tanto as informações de textura e forma local do HOG quanto as representações da CNN, fornecendo uma descrição mais rica e robusta da expressão facial naquele instante para a modelagem temporal subsequente.

4.4 Classificador - LSTM

Para analisar as sequências temporais das características combinadas (HOG+CNN) extraídas dos frames e classificar a emoção expressa no vídeo, é implementada uma RNN do tipo LSTM. A rede é construída (Código 4.10) como uma pilha de camadas. A entrada para o modelo consiste nas sequências de características combinadas, ou seja, é a dimensão do vetor resultante da concatenação das características HOG e CNN.

```
# ==================== Construir Modelo LSTM ====================
def construir_modelo_LSTM(input_shape, num_classes):
    model = Sequential([
        LSTM(128, return_sequences=True, input_shape=input_shape),
        Dropout(0.5),
        LSTM(64),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dense(num_classes, activation='softmax')
    ])
```

Lista de código 4.10 – Implementação da definição do modelo LSTM.

A primeira camada é uma LSTM com 128 unidades, configurada com *return_sequences=True*, significando que esta camada processa a sequência de entrada e produz uma sequência de saída de mesmo comprimento, permitindo que a informação temporal seja passada

para a camada seguinte. Segue-se com a camada *Dropout* com taxa de 0.5, uma técnica de regularização que desativa aleatoriamente metade das conexões durante o treinamento para mitigar o risco de *overfitting*.

A segunda camada LSTM possui 64 unidades e é configurada com *return_sequences=False*, fazendo com que a camada processe a sequência de entrada vinda da camada anterior, mas retorne apenas o vetor de saída correspondente ao último passo temporal. Este vetor final encapsula a informação relevante de toda a sequência processada. Outra camada Dropout com taxa de 0.5 é adicionada após a segunda LSTM.

Após as camadas recorrentes, uma camada Densa (totalmente conectada) com 64 neurônios é incluída para processar a representação final da sequência.

Finalmente, a camada de saída é uma camada Densa com um número de neurônios igual ao número de classes de emoção identificadas no dataset, ou seja, 6 neurônios.

A função de ativação *softmax* é utilizada nesta última camada para produzir uma distribuição de probabilidade sobre as classes de emoção, onde a classe com a maior probabilidade é selecionada como a predição do modelo para a sequência de vídeo.

4.4.1 Treino

O processo de treinamento do modelo LSTM é configurado e executado utilizando as funcionalidades da biblioteca Keras, como mostra o segmento código 4.11. Antes do treinamento, o modelo é compilado especificando-se os componentes essenciais para o aprendizado.

```
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])
    print("Modelo LSTM construído:")
    model.summary()
    return model
```

Lista de código 4.11 – Implementação da construção da LSTM.

É usado o algoritmo de otimização estocástica *adam*, baseado em gradiente amplamente utilizado e eficaz para problemas de aprendizado profundo.

O treinamento é realizado através da chamada ao método *model.fit* (Código 4.12). Este método recebe como entrada os dados de treinamento (*X_treino, y_treino*), que consistem nas sequências de características combinadas e seus respectivos rótulos numéricos de emoção.

```
    callbacks_list = [
        callbacks.EarlyStopping(monitor='val_accuracy', patience=25,
    restore_best_weights=True, verbose=1),
        callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=10,
     min_lr=1e-7, verbose=1),
    ]
```

```
5
6     print("\nIniciando treinamento do modelo LSTM...")
7     historico = modelo.fit(X_treino, y_treino,
8                            epochs=100,
9                            batch_size=16,
10                           validation_data=(X_val, y_val),
11                           callbacks=callbacks_list,
12                           verbose=1)
13
14    print("\nTreinamento concluído. Avaliando no conjunto de teste...")
```

Lista de código 4.12 – Implementação do Treino da LSTM .

O número máximo de épocas é definido como 100 e o tamanho do lote é estabelecido em 16, indicando que os pesos do modelo seriam atualizados após o processamento de 16 sequências de vídeo.

Para monitorar o desempenho e controlar o processo de treinamento, são utilizados *callbacks* do Keras. O *EarlyStopping* é empregado para interromper o treinamento caso a perda de validação não apresentasse melhora por um número consecutivo de épocas e carregasse os pesos do melhor modelo salvo pelo *ModelCheckpoint* ao final. Além disso, o *ReduceLROnPlateau* é utilizado para reduzir a taxa de aprendizado do otimizador sempre que a perda de validação estagnasse por um determinado número de épocas, auxiliando na convergência para um mínimo melhor.

O conjunto de validação separado inicialmente (20% dos dados) é fornecido ao modelo treinado, permitindo que o modelo fosse avaliado ao final de cada época, fornecendo informações sobre a generalização do modelo e servindo como base para a atuação dos *call-backs*.

4.4.2 Teste e Classificação

Após a conclusão do treinamento ao atingir o *EarlyStopping* na época 47, o modelo treinado restaura os pesos para o da melhor época, assim ele utiliza os pesos da época 22 para realizar predições no conjunto de teste, vistos no segmento de código 4.13.

```
1     loss, accuracy = modelo.evaluate(X_teste, y_teste, verbose=0)
2     print(f"Acurácia: {accuracy:.4f}")
3
4     y_pred_prob = modelo.predict(X_teste, batch_size=16, verbose=0)
5     y_pred = np.argmax(y_pred_prob, axis=1)
```

Lista de código 4.13 – Implementação da predição no conjunto teste.

Para obter a classificação final para cada sequência, aplica-se a função *np.argmax* sobre as probabilidades preditas ao longo do eixo das classes. Isso seleciona o índice (correspondente à classe de emoção) com a maior probabilidade como a predição do modelo para aquela sequência.

Com os rótulos preditos e os rótulos verdadeiros para o conjunto de teste, é possível calcular diversas métricas de desempenho utilizando funções da biblioteca *Scikit-learn.*

5 Resultados

Para avaliar o resultado do sistema de reconhecimento de emoções através da análise sequencial de imagens, utilizamos um conjunto de teste e uma aplicação com dados espontâneos, avaliando assim a aplicabilidade do sistema no mundo real.

5.1 Conjunto de Treino

Como dito antes, o conjunto de teste é utilizado para quantificar o bom funcionamento do sistema, sendo composto por 20% dos dados totais do banco de dados. Esses dados não foram utilizados durante o treinamento, fornecendo assim diversas informações sobre o desempenho do modelo proposto ao analisar novos dados.

As métricas quantitativas calculadas resumem a capacidade do classificador LSTM, alimentado pelas características HOG e CNN combinadas, em generalizar para dados não vistos. A acurácia média alcançada no conjunto de teste, ou seja, a porcentagem geral de acerto do modelo foi de 83.3%. A precisão, que mede a exatidão de previsões verdadeiras foi de 85.3% e o F1-score, uma média ponderada entre Precisão e *Recall* (capacidade de encontrar todos os positivos) atingiu 82.8%.

A análise da matriz de confusão (Figura 20) permitiu uma compreensão mais detalhada do desempenho por classe. Nela podemos observar que a emoção Felicidade (*Happiness*) e a emoção Nojo (*Disgust*) foram classificadas com 100% de precisão.

Por outro lado, o modelo demonstrou maior confusão entre outras emoções. A Raiva (*Anger*) foi corretamente identificada em 75% dos casos, mas foi confundida com Surpresa em 25% das vezes. A emoção Medo (*Fear*) apresentou a maior dificuldade de distinção, com apenas 50% de acertos, sendo confundida com Surpresa (*Surprise*) (25%), Tristeza (*Sadness*) (12%) e Nojo (*Disgust*) (12%). As emoções Tristeza (*Sadness*) e Surpresa (*Surprise*) foram classificadas corretamente em 88% das vezes, com a Tristeza sendo confundida com Nojo (12%) e a Surpresa com Medo (12%).

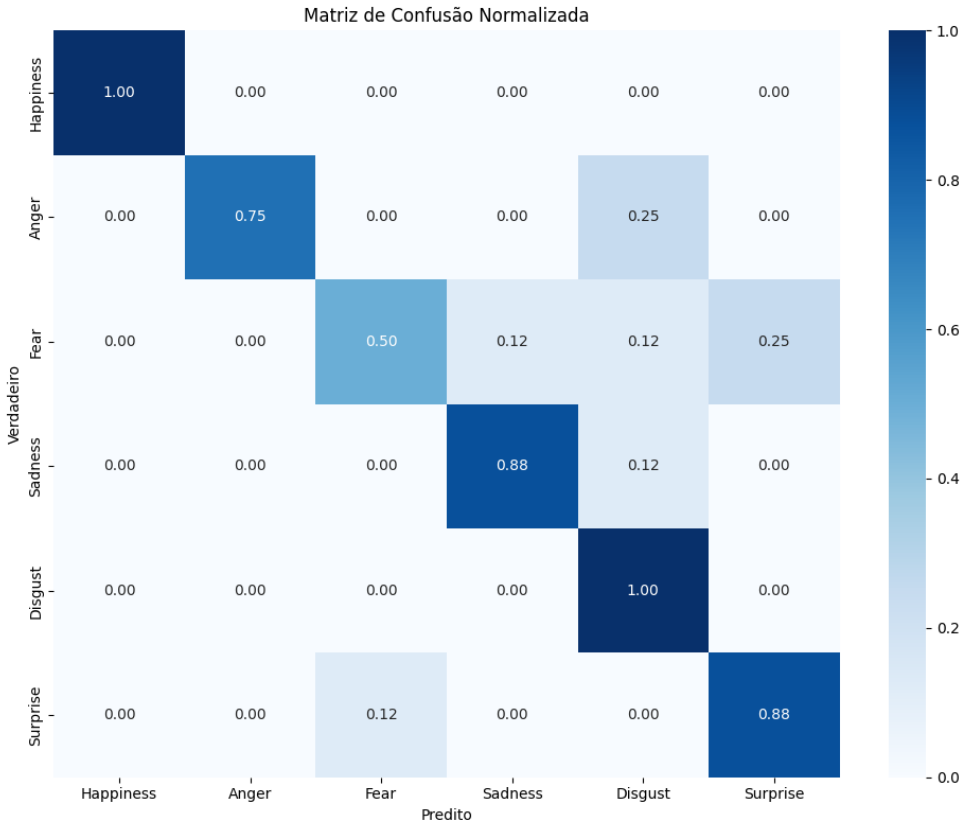

Figura 20 – Matriz de Confusão gerada.

Esses resultados indicam uma dificuldade do modelo em distinguir certas emoções com base nas características aprendidas, especialmente Medo, Raiva e Surpresa.

Nas Figuras 21 e 22 podemos observar as curvas de aprendizado, onde são plotados a acurácia e a perda nos conjuntos de treinamento e validação ao longo das épocas, forneceram informações sobre o processo de treinamento.

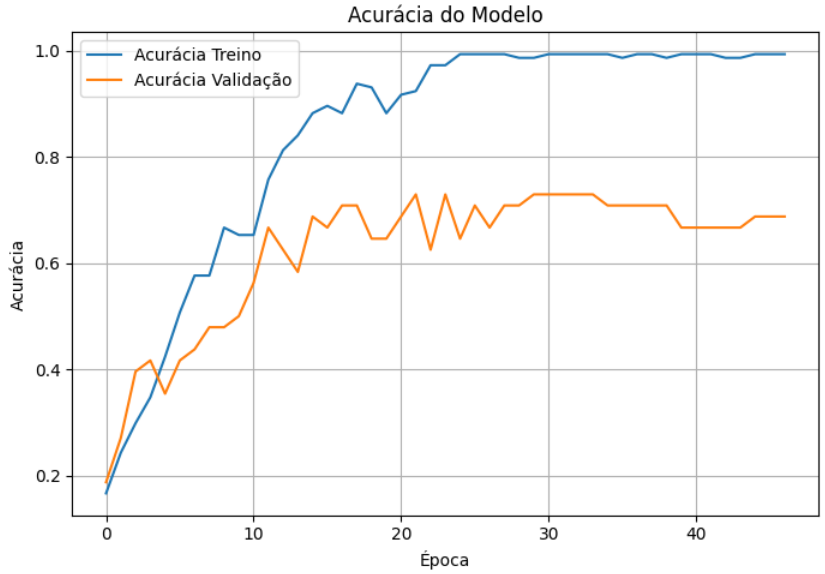

Figura 21 – Curvas de aprendizado - Acurácia.

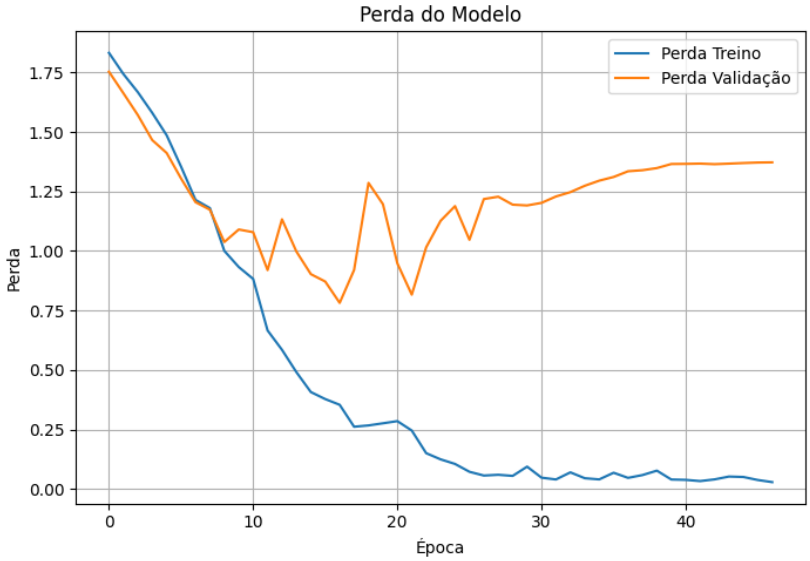

Figura 22 – Curvas de aprendizado - Perdas.

A diferença entre as curvas de treino e validação sugere um grau considerável de *overfitting*, indicando que o modelo se especializou nos dados de treinamento, mas não generaliza tão bem para dados novos.

A análise visual das amostras (Figura 23) confirmou a capacidade do modelo em classificar corretamente diversas expressões. No entanto, também são observados erros de classificação que confirmam os dados da matriz de confusão, como a classificação incorreta da expressão de Medo como Surpresa.

Figura 23 – Análise visual de amostras.

5.2 Aplicação

Com o objetivo de validar ainda mais o sistema desenvolvido, foi criada uma aplicação capaz de utilizar a rede neural previamente treinada para análise de vídeos espontâneos.

Essa aplicação emprega o mesmo pipeline adotado na fase de desenvolvimento do modelo, incluindo o pré-processamento dos frames, a detecção facial por meio do classificador Haar Cascade, a extração de característicasa MobileNetV2, e, por fim, a classificação da sequência de emoções com a rede LSTM já treinada. Essa etapa tem como objetivo avaliar a aplicabilidade prática do modelo em situações reais, fora do ambiente controlado de treinamento, permitindo uma análise mais robusta de seu desempenho em contextos naturais.

Os resultados obtidos (Figuras 24 e 25) nessa validação demonstram que o modelo apresenta boa capacidade de generalização, sendo capaz de reconhecer emoções em cenários do mundo real, apesar de cometer alguns erros.

Figura 24 – Parte 1 da amostra da aplicação em funcionamento.

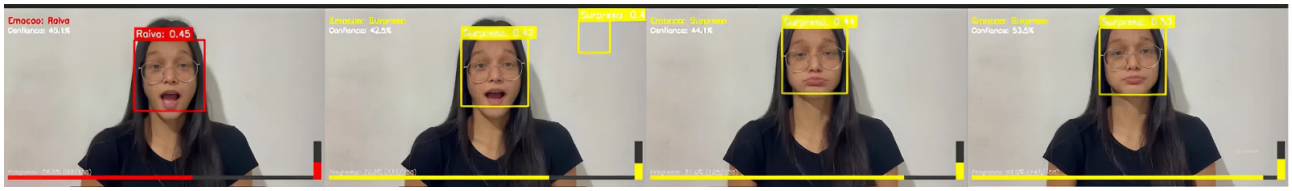

Figura 25 – Parte 2 da amostra da aplicação em funcionamento.

Abaixo podemos observar o gráfico da contagem de emoções na amostra (Figura 26), que mostra que durante a análise do vídeo foram realizadas 8 detecções para Feliz, 7 para Surpresa e 5 para Raiva.

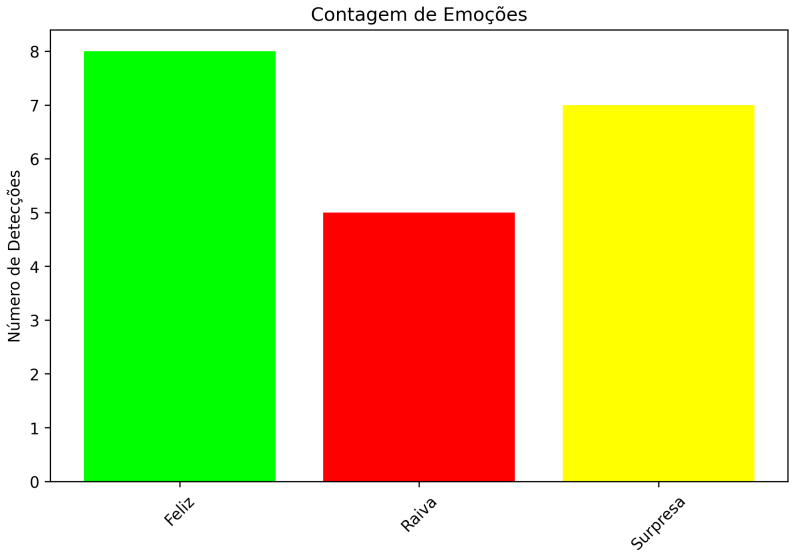

Figura 26 – Gráfico de Contagem de Emoções.

A Figura 27 mostra a evolução temporal das emoções revelando assim uma progressão sequencial das emoções ao longo dos quadros analisados. Inicialmente, a emoção Feliz foi detectada nos primeiros quadros. Em seguida, houve uma transição para Raiva, permanecendo estável por um período. Por fim, a emoção mudou para Surpresa, mantendo-se constante até o final do vídeo.

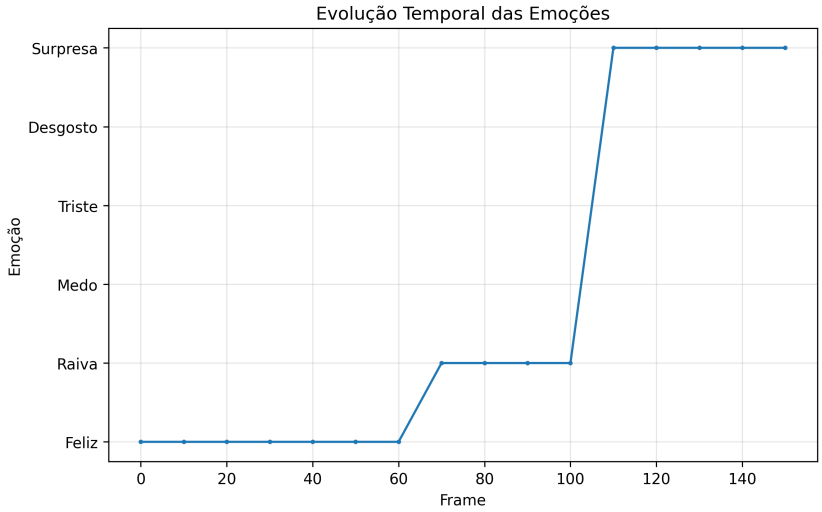

Figura 27 – Evolução Temporal das Emoções.

A Figura 28 apresenta um histograma das pontuações de confiança das detecções. A média de confiança foi de 0.508, indicada pela linha tracejada. A distribuição mostra uma concentração significativa de detecções com confiança em torno de 0.45 a 0.55, com um pico notável em 0.5. Há também uma menor frequência de detecções com confiança mais alta, como em 0.6 e um ponto isolado em 0.85, sugerindo que a maioria das detecções ocorreu com um nível de confiança moderado.

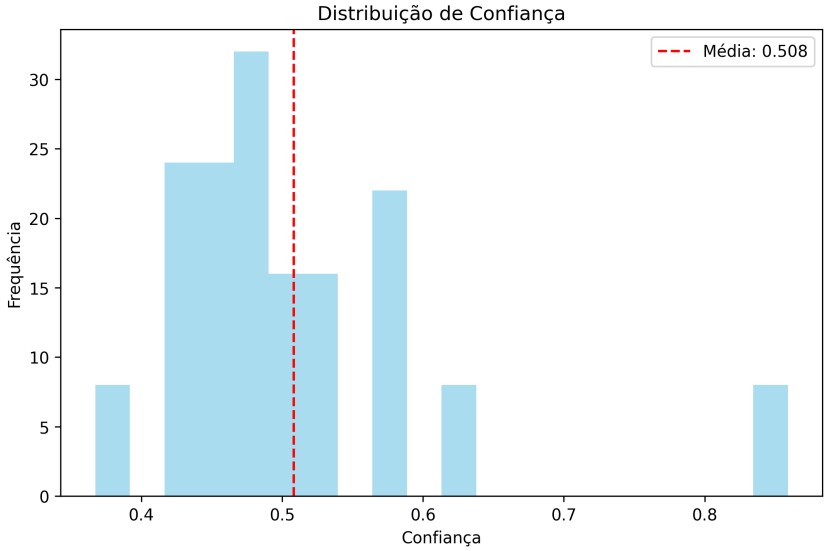

Figura 28 – Distribuição de Confiança das Detecções.

De forma geral, os resultados demonstram a viabilidade do modelo para o reconhecimento de emoções através da análise sequencial de imagens, alcançando um desempenho notável. No entanto, também evidenciam desafios significativos, especialmente na distinção entre emoções cujas expressões faciais são semelhantes.

6 Conclusão

O presente trabalho teve como principal objetivo a implementação de um sistema de reconhecimento automático de emoções por meio da análise sequencial de imagens, utilizando técnicas de visão computacional e aprendizado de máquina.

Para isso foram aplicadas técnicas para facilitar a análise, incluindo a detecção facial com classificadores em cascata baseados em características Haar (*HAAR cascade*), a combinação de características extraídas por Histograma de Gradiente Orientado (HOG) com as extraídas por uma rede neural convolucional(CNN); e uma *Long Short-Term Memory* (LSTM), com o objetivo de classificar uma sequência de frames faciais em uma emoção.

A avaliação do sistema, utilizando a base de dados *KDEF-dyn* e dados reais, apresentou um bom desempenho geral. Destacou-se a excelente precisão para as emoções Felicidade e Nojo, também apresentou taxas de acerto entre consideráveis para outras emoções.

Entretanto, foram identificadas limitações significativas, visto que o sistema apresentou dificuldade em distinguir algumas emoções. Além disso, a análise do treinamento revelou a ocorrência de *overfitting* no modelo, comprometendo sua capacidade de generalização para novos dados. Para trabalhos futuros, recomenda-se a aplicação de técnicas de regularização mais eficazes. Também é crucial ampliar e diversificar os conjuntos de dados utilizados no treinamento e na validação, de modo a incluir maior representatividade de etnias, faixas etárias, condições de iluminação e espontaneidade das expressões.

Como sugestão adicional, propõe-se a adaptação do sistema para operação em tempo real, com captura dos vídeos via webcam. Além disso, a integração multimodal, combinando a análise visual com informações de áudio, poderá contribuir significativamente para a robustez do sistema.

Conclui-se que a arquitetura proposta é tecnicamente viável e promissora. As limitações identificadas e as sugestões para trabalhos futuros abrem caminhos para novas pesquisas que busquem superar os desafios atuais e avançar no desenvolvimento de sistemas de reconhecimento automático de emoções mais precisos e eficazes.

Referências

ALVES, P. M. **Inteligência Artificial e Redes Neurais**. Brasília: IPEA, 2020. Acesso em: 18 jun. 2024. Disponível em: https://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/106-inteligencia-artificial-e-redes-neurais. Citado na página 26.

ANIL, J.; SURESH, L. P. A novel fast hybrid face recognition approach using convolutional kernel extreme learning machine with hog feature extractor. **Sensors**, v. 30, 2023. Citado 6 vezes nas páginas 8, 14, 23, 27, 29 e 35.

AWARI. **Redes neurais convolucionais: como elas funcionam**. 2023. Acesso em: 18 jun. 2024. Disponível em: https://awari.com.br/redes-neurais-convolucionais/. Citado 2 vezes nas páginas 13 e 26.

BALLESTEROS, J. *et al.* Facial emotion recognition through artificial intelligence. **Frontiers in Computer Science**, 2024. Citado 7 vezes nas páginas 8, 22, 29, 30, 33, 34 e 35.

BASIRI, M. *et al.* Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. **Future Generation Computer Systems**, v. 115, p. 279–294, 2021. Citado 3 vezes nas páginas 8, 38 e 39.

BASTOS, E. A. V.; ESTEVES, V. B. Tecnologias de reconhecimento facial: um estudo a partir do contexto de vigilância digital e sutil. **Direitos Democráticos & Estado Moderno**, v. 3, n. 1, p. 91–107, jan/jun 2021. Acesso em: 15 abr. 2025. Disponível em: https://revistas.pucsp.br/index.php/DDEM/article/view/53875. Citado na página 19.

BHATT, D. e. a. Computer vision and deep learning for emotion recognition in images: a survey. **Journal of Artificial Intelligence Research**, v. 1, p. 1–25, 2020. Citado 2 vezes nas páginas 13 e 19.

BOCK, A. M. B.; FURTADO, O.; TEIXEIRA, M. d. L. T. **Psicologias: uma introdução ao estudo de psicologia**. 14. ed. São Paulo: Saraiva, 2008. Citado 2 vezes nas páginas 17 e 18.

CACIOPPO, J. T. e. a. The psychophysiology of emotion. *In*: LEWIS, M.; HAVILAND-JONES, J. M. (ed.). **Handbook of emotions**. 2. ed. New York: Guilford Press, 2000. p. 173–191. Citado na página 19.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, 1995. Citado na página 25.

CÎRNEANU, A.-L.; POPESCU, D.; IORDACHE, D. Novas tendências no reconhecimento de emoções usando análise de imagens por redes neurais, uma revisão sistemática. **Sensors**, v. 23, n. 16, p. 1–20, 2023. Citado 8 vezes nas páginas 13, 14, 19, 25, 26, 28, 29 e 30.

DAMÁSIO, A. R. **O erro de Descartes: emoção, razão e o cérebro humano**. São Paulo: Companhia das Letras, 1994. Citado na página 17.

EKMAN, P.; FRIESEN, W. V. Measuring facial movement. **Environmental Psychology and Nonverbal Behavior**, v. 1, n. 1, p. 56–75, 1976. Citado 2 vezes nas páginas 18 e 29.

GONZÁLEZ, R. C.; WOODS, R. E. **Processamento digital de imagens**. 3. ed. São Paulo: Pearson Prentice Hall, 2010. Citado na página 21.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. Disponível em: https://www.deeplearningbook.org/. Citado 2 vezes nas páginas 25 e 26.

HANAFI, A.; BOUHORMA, M.; LOFTI, E. Building a deep learning model to generate human readable text using recurrent neural networks and lstm. 2021. Citado 3 vezes nas páginas 8, 27 e 28.

HUANG, Z. *et al.* A study on computer vision for facial emotion recognition. **Scientific Reports**, v. 13, p. 8425, 2023. Citado na página 30.

IMPLEMENTATION of Haar Cascade Classifier and Eye Aspect Ratio for Driver Drowsiness Detection Using Raspberry Pi - Scientific Figure on ResearchGate. Available at: https://www.researchgate.net/figure/a-Edge-feature-b-Line-feature-and-c-Four-Triangle-feature_fig5_339448897. Citado 2 vezes nas páginas 8 e 33.

KAROLINSKA INSTITUTET. **Karolinska Directed Emotional Faces – dynamic version (KDEF-dyn) [banco de dados]**. Acesso em: 20 maio 2025. Disponível em: https://kdef.se/. Citado 3 vezes nas páginas 16, 32 e 40.

LIENHART, R.; MAYDT, J. An extended set of haar-like features for rapid object detection. *In*: IEEE. **Proceedings of the International Conference on Image Processing**. [*S.l.: s.n.*], 2002. v. 1, p. I–I. Citado na página 33.

MIGUEL, F. K. Psicologia das emoções: uma proposta integrativa para compreender a expressão emocional. **Psico-USF**, v. 20, n. 1, p. 153–162, jan/abr 2015. Citado na página 18.

OLIVAS, E. S. *et al.* **Manual de pesquisa sobre aplicações e tendências de aprendizado de máquina: algoritmos, métodos e técnicas**. [*S.l.: s.n.*]: Referência em Ciência da Informação, 2009. Citado 3 vezes nas páginas 8, 28 e 29.

PRATEEK, J. **Artificial Intelligence with Python**. Mumbai: Packt, 2017. Citado na página 25.

RANGULOV, D.; FAHIM, M. Emotion recognition on large video dataset based on convolutional feature extractor and recurrent neural network. *In*: **Proceedings of the 4th IEEE International Conference on Image Processing, Applications and Systems (IPAS 2020)**. Genova, Itália: IEEE, 2021. p. 14–20. Citado 7 vezes nas páginas 8, 16, 18, 22, 28, 30 e 35.

ROWLEY, H. A.; BALUJA, S.; KANADE, T. Rotation invariant neural network-based face detection. *In*: **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. Santa Barbara, CA, USA: IEEE Computer Society, 1998. p. 38–44. Citado 2 vezes nas páginas 8 e 22.

SANDLER, M. *et al.* Mobilenetv2: Inverted residuals and linear bottlenecks. Google Inc., 2019. Disponível em: https://arxiv.org/abs/1801.04381. Citado 2 vezes nas páginas 37 e 38.

SARVAKAR, K. *et al.* Facial emotion recognition using convolutional neural networks. **Materials Today: Proceedings**, v. 80, p. 3560–3564, 2023. Citado na página 30.

SCHMIDT, K. L.; COHN, J. F. Human facial expressions as adaptations: evolutionary questions in facial expression. **American Journal of Physical Anthropology**, v. 44, n. S33, p. 3–24, 2001. Citado 2 vezes nas páginas 8 e 19.

SEDAGHATJOO, Z.; HOSSEINZADEH, H.; BIGHAM, B. S. Local binary pattern (lbp) optimization for feature extraction. **arXiv preprint arXiv:2407.18665v1**, jul 2024. Acesso em: 24 maio 2025. Disponível em: https://arxiv.org/abs/2407.18665. Citado 2 vezes nas páginas 8 e 24.

SHANMUGAMANI, R. **Deep Learning for Computer Vision**. Mumbai: Packt, 2018. Citado 3 vezes nas páginas 8, 19 e 26.

SHARMA, N. **What is MobileNetV2? Features, Architecture, Application and More**. 2023. Acesso em: 3 jun. 2025. Disponível em: https://www.analyticsvidhya.com/blog/2023/12/what-is-mobilenetv2-features-architecture-application-and-more/. Citado 3 vezes nas páginas 8, 37 e 38.

SHU, C.; DING, X.; FANG, C. Histogram of the oriented gradient for face recognition. **Tsinghua Science and Technology**, v. 16, n. 2, p. 216–224, apr 2011. ISSN 1007-0214. Citado na página 35.

TANG, Y. **Deep learning using linear support vector machines**. Toronto, 2013. Acesso em: 12 jun. 2025. Disponível em: https://doi.org/10.48550/arXiv.1306.0239. Citado na página 29.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. *In*: IEEE. **Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)**. Kauai, HI, 2001. p. I–511–I–518. Citado 3 vezes nas páginas 8, 33 e 34.

WILHELM, O. e. a. Test battery for measuring the perception and recognition of facial expressions of emotion. **Frontiers in Psychology**, v. 5, p. 1–23, 2014. Citado na página 13.

Apêndices

APÊNDICE A – Material Completo

Este apêndice disponibiliza o link para acesso ao repositório do projeto no GitHub, que contém os códigos desenvolvidos durante a elaboração deste trabalho, incluindo os scripts de pré-processamento, treinamento, teste e validação do modelo de reconhecimento de emoções, bem como arquivos auxiliares e instruções para reprodução dos experimentos.

Para acessar o repositório com a solução desenvolvida neste trabalho, acesse o seguinte link: https://github.com/larisard/Emotion-Recognitions

Caso haja qualquer dificuldade no acesso ou na utilização dos materiais,entre em contato por e-mail: larissasardinha@pq.uenf.br.

Agradecemos pela atenção e pelo interesse na pesquisa apresentada. Esperamos que o material disponibilizado seja útil e contribua para o desenvolvimento de futuros estudos e projetos na área.