UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII
CENTRO CIÊNCIAS EXATAS E SOCIAIS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PATRICK DE ÂNGELIS AMÂNCIO MENESES SANTOS

DESENVOLVIMENTO DE UMA APLICAÇÃO DE RECONHECIMENTO DE EXPRESSÕES FACIAIS PARA AUXÍLIO À TESTES DE USABILIDADE DE SOFTWARE

PATOS
2022

PATRICK DE ÂNGELIS AMÂNCIO MENESES SANTOS

DESENVOLVIMENTO DE UMA APLICAÇÃO DE RECONHECIMENTO DE EXPRESSÕES FACIAIS PARA AUXÍLIO À TESTES DE USABILIDADE DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao Programa de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação

Área de concentração: Aprendizagem de Máquina

Orientador: Profa. Dra. Jannayna Domingues Barros Filgueira

PATOS
2022

S237d Santos, Patrick de Angelis Amâncio Meneses.

Desenvolvimento de uma aplicação de reconhecimento de expressões faciais para auxílio à testes de usabilidade de software [manuscrito] / Patrick de Angelis Amancio Meneses Santos. - 2022.

48 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas , 2022.

"Orientação : Profa. Dra. Jannayna Domingues Barros Filgueira , Coordenação do Curso de Ciências Exatas - CCEA."

1. Visão computacional. 2. Redes Neurais Convolucionais. 3. Reconhecimento de expressões. 4. Aprendizagem de máquina. I. Título

21. ed. CDD 005.3

Elaborada por Edson M. Almeida - CRB - CRB 15/743 BSC 7

PATRICK DE ÂNGELIS AMÂNCIO MENESES SANTOS

DESENVOLVIMENTO DE UMA APLICAÇÃO DE RECONHECIMENTO DE EXPRESSÕES FACIAIS PARA AUXÍLIO À TESTES DE USABILIDADE DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 24/03/2022

BANCA EXAMINADORA

Jannayna Domingues Barros Filgueira
_____
Prof. Dra. Jannayna Domingues Barros Filgueira
(Orientadora)

Ricardo Santos de Oliveira
_____
Prof. Dr. Ricardo Santos de Oliveira
(Examinador)

Fernando Medeiros Filho
_____
Prof. Dr. Fernando Medeiros Filho
(Examinador)

Dedico este trabalhos aos meus pais e a minhas avós.

AGRADECIMENTOS

Ao meu pai Arimateia e minha mãe Maria, por sempre serem o braço forte e a mão amiga durante toda minha existência.

Aos meus colegas de classe, em especial Luiz, Natan e Caio, que estiveram juntos nessa jornada e fizeram com que todo o trajeto fosse mais fácil, agradeço pelo apoio e amizade.

À professora Jannayna pela paciência e auxílio que vem se estendendo da iniciação científica até a orientação deste trabalho.

Aos meus colegas de trabalho, em especial Pagar.me, por me apoiarem sempre e compreenderem o momento que estava passando.

Aos professores do Curso de Ciência da Computação deste campus, em especial, Francisco Anderson, Fábio Júnior, Jannayna, Pablo Roberto, Pablo Suarez e Ingrid, que contribuíram com conhecimento e inspiração ao longo destes anos.

Aos funcionários da UEPB, pela presteza e atendimento sempre que necessário.

Aos amigos e familiares pelo apoio de sempre.

"Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que há muito a fazer."

Alan Turing

RESUMO

Produtos de software cada vez mais são produzidos para a sociedade, porém para que façam sentido para o usuário final é necessário que passem por um processo de validação e refatoração junto dos usuários finais. Neste contexto entra o teste de usabilidade, que visa justamente o feedback do usuário, porém muitas vezes o usuário não expressa tudo que sente ao usar o sistema, sendo este possível detectar através de um sistema que captura as emoções faciais durante a aplicação do teste. Este trabalho implementa um algoritmo de detecção de expressões faciais fazendo uso de técnicas de aprendizado de máquina - em especial o *transfer learning*, redes neurais convolucionais; e também um software capaz de gravar as sessões de teste. Para tornar possível foi treinado uma rede neural convolucional VGG16 utilizando a base de dados FER2013. O produto final é passível de utilização pois consegue reconhecer as emoções: felicidade, neutro, raiva e surpresa. Porém ainda sofre com problemas de instabilidade de classificação, não conseguindo detectar nojo.

Palavras-chave: Visão Computacional. Redes Neurais Convolucionais. Reconhecimento de expressões.

ABSTRACT

Software products are increasingly produced for society, but so that make sense to the end-user, they must go through the validation process and refactoring with end-users. In this context comes the usability test, which aims to the user's feedback, but many times precisely the user does not express everything that feels when using the system, which is possible to detect through a system that captures the facial emotions during the test application. This work implements an algorithm for the detection of facial expressions using machine learning techniques - especially transfer learning, convolutional neural networks; and also software capable of recording test sessions. To make it possible, the convolutional neural network was trained VGG16 using the FER2013 database. The final product is used because can recognize emotions: happiness, neutral, anger, and surprise. but still suffer having rating instability issues, failing to detect disgust

Key-words: Computer Vision. Convolutional Neural Networks. Expressions Recogntion.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

VC	Visão Computacional
CNN	Convolutional Neural Network
RNA	Rede Neural Artificial
SVM	Support Vector Machine
KNN	K-nearest Neighbors
CSV	Comma-separated Values
XML	eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

O processo de desenvolvimento de software é uma tarefa cada vez mais complexa. Várias camadas de abstração incluem o desenvolvimento dele, algo que vai muito além de código puro e atinge os domínios de produto, tecnologia, marketing e usabilidade. Dentre estes a usabilidade é um dos tópicos que mais pode determinar a adoção por parte dos usuários.

Segundo Hix e Hartson (1993), usabilidade é o conceito utilizado para descrever a qualidade da interação de uma interface diante de seus usuários. Sabendo disso, a usabilidade não se deve restringir apenas ao que se pensa que é melhor para o usuário, mas sim testar hipóteses diretamente com o mesmo, através de testes, e da coleta do feedback dos usuários.

As expressões faciais são a forma mais significativa e forte que ocorre na comunicação para que se possa conhecer o estado emocional de uma pessoa durante a comunicação (PARADA, 2017). Através delas acontece uma comunicação não verbal, onde através de movimentos da face, emoções são expostas.

Dessa forma para adicionar mais informações sobre qual a experiência do usuário durante uma sessão de teste de usabilidade de algum produto, o feedback não verbal que as expressões faciais podem trazer, pode ser mais uma ferramenta, para tentar simplificar a experiência com o produto e o torná-lo melhor na solução do problema do usuário.

A identificação das expressões faciais e sua classificação por computadores só é possível devido ao advento da visão computacional das redes neurais.

A Visão Computacional é uma ciência responsável pela forma como a máquina enxerga o ambiente a sua volta, por meio da extração de características de imagens originadas de dispositivos de captura(câmeras, scanners, etc), segundo Chiu e Raskar (2009). Todavia o processo de detecção de objetos, e em especial, faces e logo após sua classificação não é algo fácil para a máquina, que necessita realizar operações e utilizar técnicas de alta complexidade.

Para apoiar com o processo de visão computacional automatizado e classificação das expressões faciais, as Redes Neurais Convolucionais entram como uma das técnicas mais recentes. As Redes Neurais Convolucionais são um subconjunto das Redes Neurais Artificiais. As últimas se baseiam no funcionamento das redes neurais biológicas, para aprender novos conhecimentos, e as primeiras utilizam da mesma lógica, só que com inspiração no processo biológico de visão e detecção de características em imagens.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Desenvolver uma aplicação de apoio a testes de usabilidade, que captura a imagem da face de um usuário durante uma sessão de testes de usabilidade e ao fim, gera quais foram as emoções predominantes.

1.1.2 Objetivo específico

- Realizar estudo sobre visão computacional, redes neurais artificiais e redes neurais convolucionais;
- Desenvolver um algoritmo capaz de detectar emoções;
- Desenvolver um sistema que utilize o algoritmo e gere relatórios das sessões de testes, com as emoções predominantes.

1.2 PROBLEMÁTICA

A demanda por software só aumenta. O trabalho de forma ágil é fundamental para que hipóteses sejam formuladas e testadas em um curto espaço de tempo. Porém mecanismos para testes de usabilidade, ainda acabam se furtando apenas à comunicação direta, verbal com o usuário durante o teste e também através do preenchimento de questionários pós teste.

1.3 JUSTIFICATIVA

Cada vez mais software destinado à interação com os seres humanos está sendo desenvolvido. Com isso é necessário desenvolver técnicas mais avançadas para realizar testes com estes usuários, de modo a desenvolver um produto mais adequado e confortável ao usuário. O uso da linguagem, não verbal, busca atingir - dentro do teste de usabilidade - uma nova dimensão, que muitas vezes o usuário não consegue expressar com palavras. As emoções detectadas, em conjunto com o material já existente podem somar mais para avaliar possíveis dificuldades do usuário ao utilizar o produto, como também entender pontos fortes. Através dos testes de usabilidade, podemos ter feedbacks muito relevantes dos usuários, para aprimorar o sistema, porém muitas vezes não conseguimos expressar totalmente nossa opinião ao usá-lo. Este trabalho se propõe a desenvolver um sistema capaz de detectar expressões faciais durante uma sessão de teste de software, de modo a fornecer principalmente ao mercado ao pesquisador, mais feedbacks da sensação do usuário

sobre o software em teste. Para isto será feito o uso de técnicas de visão computacional e redes neurais artificiais.

1.4 ORGANIZAÇÃO DO TRABALHO

Esta pesquisa está organizada em 6 capítulos. O capítulo 2 é relativo à fundamentação teórica, aborda todos os temas e conceitos das tecnologias bases para a criação do sistema capaz de reconhecimento de expressões faciais: Visão Computacional e Redes Neurais Convolucionais. Além disso, há uma definição sobre teste de usabilidade. O capítulo 3 detalha como foi a implementação da solução. No capítulo 4 os resultados do trabalho são apresentados. No capítulo 5 está a conclusão do trabalho acompanhado de sugestões de trabalhos futuros. No fim as referências.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos que embasam o desenvolvimento deste trabalho.

2.1 VISÃO COMPUTACIONAL

Crowley e Christensen (2011), definem que Visão computacional é uma área da ciência dedicada ao estudo e desenvolvimento de métodos e teorias voltados à extração automática de informações úteis que estejam contidas em alguma imagem. Estas imagens por sua vez são capturadas através de dispositivos, como câmeras e scanners.

Devido a esse aspecto de trabalho com processamento de imagens digitais, para Barelli (2018) a Visão Computacional (VC) é uma ciência que engloba várias áreas como a matemática, física, dentre outras. Hoje existem diversas bibliotecas e linguagens de programação para realizar o desenvolvimento de softwares nessa área, todas elas em um alto nível de abstração, encapsulando conceitos matemáticos. Dessa forma conseguimos ter um acesso mais fácil ao desenvolvimento de novas ferramentas.

Para Backes e Junior (2018), o processo de Visão Computacional pode ser dividido em 5 fases descritas abaixo e ilustradas na figura 01:

- Aquisição: Realiza a captura da imagem por meio de dispositivos que tentam simular a função do olho. Exemplos são: Câmeras fotográficas, filmadoras, scanners, etc.
- Processamento de imagens: Para que possamos aplicar algum método de VC muitas vezes é necessário tratar a imagem, para que satisfaça as condições exigidas pelo método a ser aplicado. Esse processo pode se dar através da retirada de ruído, destaque de bordas, suavização da imagem, aumento de contraste, rotação da imagem, etc.
- Segmentação: Particiona a imagem em regiões de interesse. Como exemplo, uma imagem em que possui várias pessoas, poderia estar interessado em extrair os rostos das pessoas.
- Extração de características: Consiste na extração de um conjunto de características de objeto de interesse. Onde se busca algum atributo específico na imagem, que caracterize o objeto. Como uma espécie de “impressão digital” (analogia imperfeita) que permita identificá-lo.
- Reconhecimento de padrões: Esta etapa tem como papel classificar ou agrupar as imagens com base em seus conjuntos de características. Por exemplo, uma foto de

uma única laranja, saber-se que aquele objeto pertence à classe "laranja" com base em atributos como cor, rugosidade da casca, formato, tamanho etc. É importante salientar que o objeto visto não é igual às laranjas vistas no passado, mas apenas similar (na verdade, segundo a filosofia, a igualdade é um conceito teórico que não existe na natureza). No entanto, mesmo com essa limitação consegue-se classificá-lo corretamente na maioria dos casos.

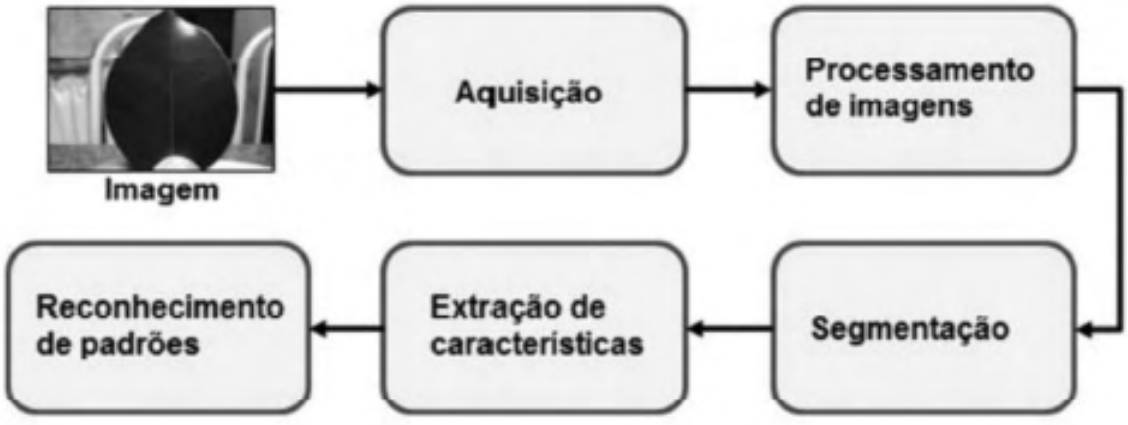

Figura 1 – Diagrama com etapas de processo de Visão Computacional

Fonte: Backes e Junior (2018)

2.1.1 Reconhecimento de padrões em imagem

O reconhecimento de padrões em imagens consiste na classificação a partir de características em imagens como texturas, cores, formas, entre outros. Através do reconhecimento de padrões, os sistemas de visão computacional conseguem identificar determinados objetos. Humanos e máquinas utilizam características particulares para diferenciar e reconhecer objetos (BARELLI, 2018).

Para os humanos é uma atividade tão comum que passa despercebida. Assim que vê algum objeto já conhecido, o ser humano identifica facilmente e o consegue diferenciar de outros, como quando diferenciamos um gato de um pássaro. Para que isso aconteça é realizada a captura e o processamento dessa informação, associando-a a alguma classe de dados.

Barelli (2018) define uma classe como um conjunto de padrões que possuem características em comuns. Por exemplo: para nós humanos, entender as características que definem um gato e um pássaro é algo fácil, todavia, para a máquina reconhecer e classificar esses padrões não é uma tarefa simples.

Para realizar a classificação de objetos podemos utilizar várias opções de algoritmos: algoritmos tradicionais de Machine Learning como Support Vector Machine(SVM), algoritmos de clusterização como K-nearest Neighbors (KNN) e mais modernamente as Convolutional Neural Networks(CNN).

2.1.2 Deteção da face

Devido ao aumento do número de computadores com maior capacidade de processamento e menor custo, o interesse no processamento digital de imagens aumentou, sendo incorporado em diversas aplicações como autenticação biométrica e vigilância. Todos tendo o reconhecimento de face como uma parte fundamental da aplicação (LI; JAIN, 2011).

Um sistema de detecção de facial busca imitar as capacidades naturais de um humano para reconhecer características faciais em diferentes ambientes e associá-las a informações já armazenadas na memória (FONSECA, 2016). Para Gouveia (2010), a detecção de faces consiste na utilização de técnicas computacionais para determinar se em uma imagem, há ou não faces, e existindo, deverá retornar a localização de cada face.

Existem dois métodos básicos para reconhecimento de rosto descritos em Agarwal et al. (2010). O primeiro baseia-se na extração de vetores de características de partes básicas de um rosto: olhos, nariz, boca e queixo. Essas características são coletadas e armazenadas em um vetor de recursos. O segundo método consiste na análise de componentes principais, onde as informações que melhor descrevem uma face são derivadas da imagem completa da face.

Diversas técnicas são utilizadas para detecção e extração de características da face em imagens, porém uma das mais utilizadas e conhecidas é a Haar Cascade.

2.1.3 Haar Cascade

Para que o Haar Cascade extraia as características do objeto desejado, ele é treinado através do algoritmo de Machine Learning AdaBoost. Para o treinamento, uma grande quantidade de imagens positivas e negativas são aplicadas ao algoritmo, gerando uma cascata dessas características, que são salvas em um arquivo XML.

Para extrair as características do objeto de interesse, o Haar Cascade utiliza máscaras chamadas de Haar Features, exemplificadas pela figura 2. Estas máscaras percorrem a imagem em que se busca o objeto segmentado, atuando como um filtro, efetuando as operações para extrair o objeto (BARELLI, 2018).

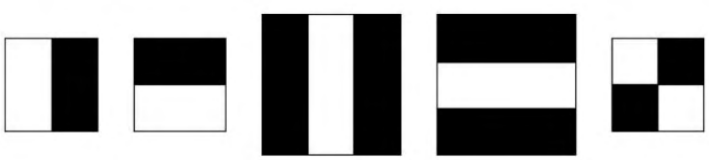

Figura 2 – Haar Features

Fonte: Barelli (2018)

A figura 3 mostra as máscaras Haar Features realizando seu processo de extração de características.

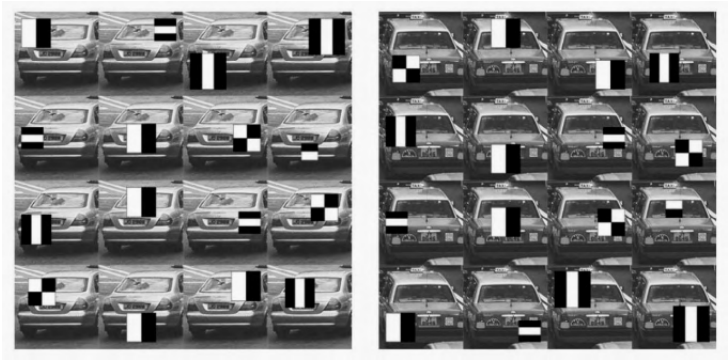

Figura 3 – Processo de extração de características

Fonte: Barelli (2018)

Para cada ponto da imagem que as máscaras percorrem, elas buscam sobrepor o objeto de interesse. O algoritmo realiza a subtração, da soma dos pixels sobrepostos pelo retângulo branco e da soma dos pixels sobrepostos pelo retângulo preto. A partir dessa operação é obtida uma característica do objeto, que acaba sendo a luminosidade e a intensidade do tom de cinza do pixel. Tendo este último fato em conta, implica que a variação da luminosidade pode alterar no resultado final da detecção. (BARELLI, 2018).

2.1.4 Reconhecimento de expressões e emoções faciais

Para Jain, Shamsolmoali e Sehdev (2019), o reconhecimento de emoções faciais é uma linha de pesquisa da área de visão computacional, computação afetiva, interação humano-computador e comportamento humano que lida com a previsão de emoções usando expressões faciais em imagens ou vídeos.

O processo de reconhecimento de expressões faciais consiste na detecção de emoções através de uma imagem onde exista uma face humana. Para cada emoção existe um conjunto de expressões faciais que podem representá-la. O sistema irá retornar para uma

expressão em um dado momento, qual sentimento com maior probabilidade, ele pode estar representando.

Para Zhang et al., apud Cosseti (2015), um sistema automático de reconhecimento de expressões deve resolver os seguintes problemas:

- Detecção e localização da face em uma cena;
- Extração de características da face;
- Redução de dimensionalidade;
- Classificação da expressão.

2.1.5 Emoções básicas em expresões faciais

Humanos diariamente expressam suas emoções, elas podem ser percebidas de diferentes formas e uma delas são as expressões faciais. De acordo com Libralon (2014) as expressões faciais são uma forma de comunicação não verbal, para Parada (2017) elas são a formas mais significativa e forte que ocorre na comunicação para conhecer o estado emocional de uma pessoa durante a comunicação.

Existem alguns sistemas de classificação de emoções, entre eles o de Paul Ekman é um dos mais conhecidos. Ekman (1976) definem as sete emoções básicas como: surpresa, medo, raiva, desgosto, tristeza, felicidade e neutro. Cada emoção expressa é descrita por três partes do rosto: sobrancelhas-testa, pálpebras-lábios e parte inferior do rosto, Ekman (1972). A figura 4 exemplifica como funciona um sistema de codificação de ação facial.

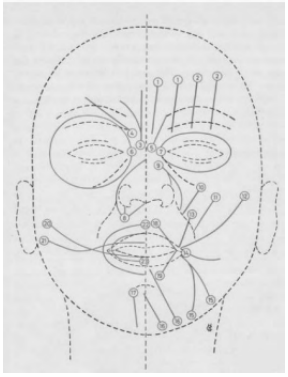

Figura 4 – Representação do sistema de codificação de ação facial

Fonte: AlMarri (2019)

2.2 TESTE DE USABILIDADE

O teste de usabilidade é um processo no qual participantes representativos avaliam o grau que um produto se encontra em relação a critérios específicos de usabilidade (RUBIN, 2018).

Para Ferreira (2002), o teste pode servir para diferentes propósitos que envolvem tipos de tarefas, medidas de performance e disposição de escalas, entrevistas ou inspeções a serem aplicadas, buscando encontrar problemas de usabilidade e fazer recomendações no sentido de eliminar os problemas e melhorar a usabilidade do produto, ou com a finalidade de se comparar dois ou mais produtos.

Os usuários finais são as pessoas que operam diariamente a interface digital e não tem conhecimento especialista sobre como ela funciona. Os usuários são a maioria das pessoas e que usam editores de texto, redes sociais ou caixas de banco. Usuário é aquele que conhece os caminhos a serem seguidos e tem comportamentos mecanizados durante sua navegação, aprendidos com a prática, o que permite a criação de um conhecimento tácito sobre a operação da interface (BERG, 2017).

Com a realização de testes de usabilidade, pode-se registrar os melhores resultados obtidos para futuras realizações levando à minimização do custo do serviço de suporte aos usuários, crescimento de vendas e prever o lançamento de produtos com menos problemas de usabilidade e mais competitivos (FERREIRA, 2002).

Testes de usabilidade são baseados na experimentação, no uso do objeto a ser pesquisado, nesse caso interfaces digitais. A partir da navegação de uma interface digital é possível aos pesquisadores coletar dados sobre o desempenho e satisfação do usuário. Para avaliar a eficiência e a eficácia de uma interface são usadas técnicas como a cronometragem dos tempos de execução de uma tarefa, a contagem dos erros de digitação, a retroalimentação em vídeo (vídeo feedback), gravação dos movimentos do mouse da tela, registro de arquivos e movimentos dos olhos (eye-tracking), entre outros. Essas ferramentas permitem avaliar a eficiência e a eficácia de um usuário e assim chegar ao desenvolvimento de um indicador que sirva de parâmetro para avaliar a acessibilidade de uma interface digital (BERG, 2017).

2.3 REDES NEURAIS ARTIFICIAIS

Guedes (2017) define Redes Neurais Artificiais como grupos de neurônios interligados que formam um sistema nervoso, tal como o cérebro humano. As redes neurais artificiais são abstrações do modelo biológico que permitem compor os neurônios como unidades simplistas de computação como ilustrado na figura 5. Nessa analogia os neurônios artificiais replicam o comportamento dos biológicos ao agrupar os impulsos provenientes de

entradas ou de axônios de outros neurônios até que um certo limite é atingido, esse limite é definido pela função de ativação. Quando esse limite é atingido, o neurônio dispara um impulso por seu axônio que por sua vez pode estimular outros neurônios ou ser uma saída.

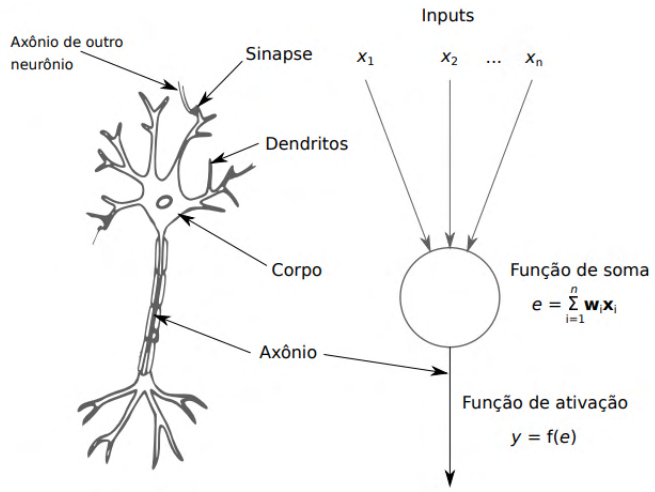

Figura 5 – Neurônio biológico e neurônio artificial

Fonte: Guedes (2017)

A visão moderna das redes neurais começou nos anos 1940 onde MCCULLOCH (1943), mostraram que as RNAs poderiam, em princípio, computar qualquer cálculo aritmético e lógico. Este trabalho frequentemente é reconhecido como a origem do campo de pesquisa das redes neurais artificiais (HAGAN HOWARD B. DEMUTH, 2014).

Um neurônio artificial, ilustrado na figura 5, tem como habilidade mais importante aprender de acordo com um conjunto de dados de entrada. O procedimento para realizar o processo de aprendizagem é chamado de algoritmo de aprendizagem, cuja função é modificar os pesos sinápticos da rede de uma forma ordenada para alcançar um objetivo de projeto desejado (HAYKIN, 2001).

Como ilustra a Figura 6 Haykin (2001), define três elementos básicos do neurônio artificial:

- Um conjunto de sinapses caracterizadas por um peso.
- Um somador para soma os sinais de entradas.
- Uma função de ativação para restringir a amplitude e saída de um neurônio.

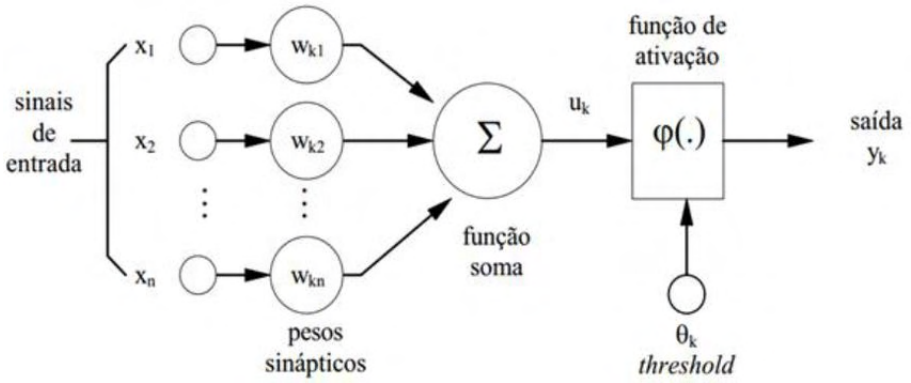

Figura 6 – Neurônio artificial

Fonte: Haykin (2001)

2.4 REDES NEURAIS CONVOLUCIONAIS

Redes Neurais Convolucionais(Convolutional Neural Network - CNN) são uma variação de uma Rede Neural Artificial comum, dedicada ao processamento de dados visuais. Karpathy (2017) apud Araújo (2017) elenca as seguintes vantagens das CNNs: "capacidade de extrair características relevantes através de aprendizado de transformações (kernels) e depender de menor número de parâmetros de ajustes do que redes totalmente conectadas com o mesmo número de camadas ocultas". Como cada unidade de uma camada não é conectada com todas as unidades da camada seguinte, há menos pesos para serem atualizados, facilitando assim o treinamento.

Um dos primeiros projetos de CNNs foi desenvolvido por Yann LeCun em 1988, tendo a mesma auxiliado a impulsionar o campo de Deep Learning. Inicialmente, a LeNet foi utilizada para reconhecimento de caracteres, tais como código postal e dígitos numéricos (ARAúJO, 2017). A arquitetura da rede está descrita na figura 7.

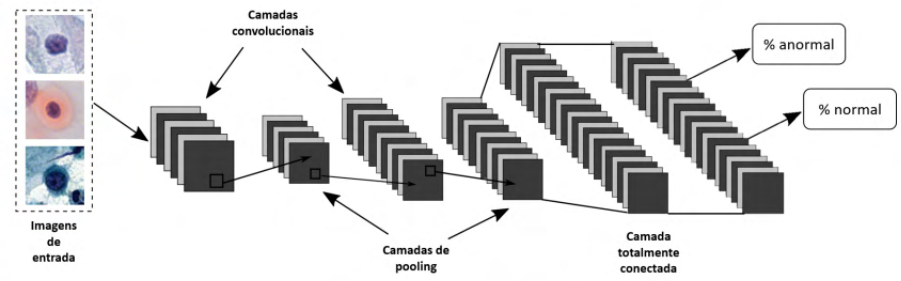

Figura 7 – Arquitetura de uma LeNet

Fonte: Araújo (2017)

A arquitetura de uma LeNet tem três principais camadas: convolucionais, de pooling e totalmente conectadas. As camadas convolucionais são responsáveis por extrair atributos dos volumes de entradas. As camadas de pooling são responsáveis por reduzir a dimensionalidade do volume resultante após as camadas convolucionais e ajudam a tornar a representação invariante a pequenas translações na entrada. As camadas totalmente conectadas são responsáveis pela propagação do sinal por meio da multiplicação ponto a ponto e o uso de uma função de ativação. A saída da CNN é a probabilidade da imagem de entrada pertencer a uma das classes para a qual a rede foi treinada. Na seção seguinte, detalhamos o que ocorre em cada uma dessas camadas (ARAúJO, 2017).

2.4.1 Camada de convolução

As camadas convolucionais consistem de um conjunto de filtros que recebem como entrada um arranjo 3D, também chamado de volume. Cada filtro possui dimensão reduzida, porém ele se estende por toda a profundidade do volume de entrada. Por exemplo, se a imagem for colorida, então ela possui 3 canais(Vermelho, Verde e Azul) e o filtro da primeira camada convolucional terá tamanho 5x5x3 (5 pixels de altura e largura, e profundidade igual a 3). Automaticamente, durante o processo de treinamento da rede, esses filtros são ajustados para que sejam ativados em presença de características relevantes identificadas no volume de entrada, como orientação de bordas ou manchas de cores. A relevância é avaliada de tal forma que os resultados sejam otimizados em função de um conjunto de amostras previamente classificadas (ARAúJO, 2017). A figura 8 exemplifica este processo.

Tanto o kernel(filtro) como a imagem, basicamente são matrizes. O processo de convolução consiste, que o kernel(que deve ser menor que a imagem) percorre os pixels da imagem, realizando uma multiplicação elemento a elemento entre o kernel e o subconjunto de pixels, que está selecionado pelo kernel, e a matriz resultante dessa multiplicação é somada e gera um novo valor, que ao fim da convolução vai gerar uma nova matriz.

2.4.2 Camada de pooling

Após uma camada convolucional, geralmente existe uma camada de pooling. O objetivo dessa camada é reduzir progressivamente a dimensão espacial do volume de entrada, consequentemente a redução diminui o custo computacional da rede e evitar o overfitting (ARAúJO, 2017).

Há duas maneiras de fazer o pooling: pool máximo e pool médio. Em ambos os casos, a entrada é dividida em espaços bidimensionais não sobrepostos. Por exemplo, na Figura 9, a camada 2 é a camada onde será aplicada o pooling. Para o pool médio, a média dos quatro valores na região são calculados. Para o pool máximo, o valor máximo dos quatro valores é selecionado (SAMER; RISHI; ROWER, 2015).

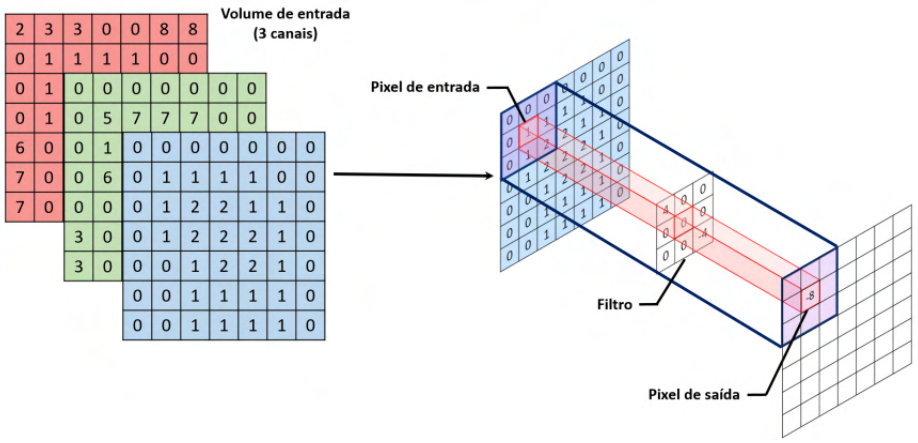

Figura 8 – Convolução de um filtro 3x3

Fonte: Araújo (2017)

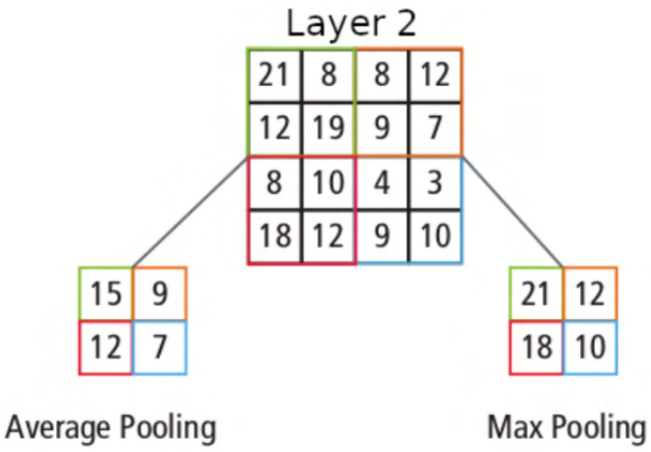

Figura 9 – Convolução de um filtro 3x3

Fonte: SAMER, RISHI e ROWER (2015)

2.4.3 Camada totalmente conectadas

A camada totalmente conectada tem esse nome devido aos neurônios anteriores estarem conectados com os próximos neurônios. É nessas camadas que acontece a propagação do sinal por meio da multiplicação ponto a ponto, como também se utiliza de uma função de ativação (ARAúJO, 2017). Esta camada é basicamente igual as camadas de uma rede neural comum. Enquanto nas camadas anteriores acontece a extração de características, aqui ocorre a classificação.

2.5 VGG16

Perumanoor (2021) define o VGG16 como uma arquitetura de rede neural convolucional, desenvolvida por Karen Simonyan e Andrew Zisserman, da universidade de Oxford no ano de 2014, divulgada através do artigo "Very Deep Convolutional Networks for Large-Scale Image Recognition". O nome VGG tem origem da abreviação do grupo de pesquisadores responsável pelo seu desenvolvimento, Visual Geometry Group, o "16" é devido ao fato da arquitetura possuir 16 camadas, conforme ilustra a figura 10. Ele se tornou amplamente conhecido por ser desenvolvido para o desafio ImageNet do Google, ficando no top-5, com acurácia de 92.7

Devido a este fato ele é amplamente utilizado para desenvolvimento de modelos e soluções de Deep Learning, principalmente ligadas a classificação de objetos, via treinamento base ou *transfer learning*.

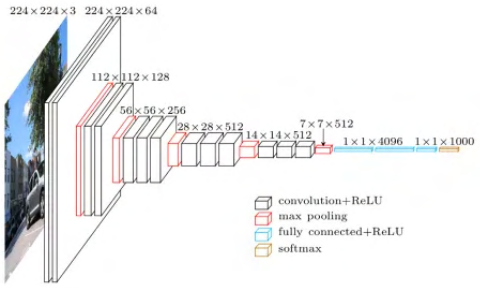

Figura 10 – Arquitetura VGG16

Fonte: Simonyan e Zisserman (2015)

2.6 VGGFACE

O modelo VGGFace foi desenvolvido pela universidade de Oxford e reinado com uma coleção de 2,6 milhões faces humanas. A ideia em utilizá-lo é por ser algo mais próximo do contexto em que se quer realizar a transferência de aprendizado, ele é mais específico em extrair características de rostos.

2.7 TRANSFER LEARNING

O *transfer learning* consiste em utilizar um classificador pré-treinado como ponto de partida para uma nova classificação Rosebrock (2019). A utilização consiste em utilizar os pesos do modelo já treinado em algum conjunto de dados. Rosebrock (2019) classifica os tipos de *transfer learning* em:

1. Extração de características: onde a rede pré-treinada é utilizada apenas para gerar como saída, as características que serão utilizadas por um outro algoritmo para realizar a classificação. A rede pré-treinada não sofre nenhuma alteração.

2. Fine-tunning: Nesse caso a rede original tem sua arquitetura modificada, removendo e/ou adicionando novas camadas e tendo seus parâmetros atualizados.

2.8 TRABALHOS RELACIONADOS

O trabalho de Mendonça (2018) desenvolve um modelo capaz de detectar 4 categorias de emoções ligadas às expressões: alegria, raiva, surpresa e desgosto. Para ser capaz de realizar essa detecção, foi utilizada a base de dados JAFFE, contendo 213 imagens, sendo que todas as imagens relativas às expressões de medo e tristeza foram retiradas, passando a 150 imagens. Para o pré-processamento houve a redução da dimensionalidade da imagem de 64x64 para 48x48 pixels. A arquitetura utilizada foi a LeNET, que teve os dados separados em 80% para treinamento e 20% para validação. A rede foi treinada na plataforma Google Colab, chegando a uma precisão de 62%. Para a detecção das faces foi utilizado o algoritmo de Haar Cascade.

O trabalho de Farias (2019) produz um modelo capaz de detectar 7 categorias de expressões faciais: felicidade, medo, raiva, tristeza, nojo, surpresa e neutra. Para ser capaz de realizar essa detecção, ela foi treinada com a base de dados FER2013 contendo 8989 imagens, sendo que todas as imagens relativas às expressões do tipo nojo, raiva, surpresa, tristeza, neutra, felicidade e medo. Estando todas com a resolução de 48x48 pixels. A arquitetura utilizada foi a Mini Xception pré-treinada fornecida pela biblioteca Keras. Para a detecção das faces foi utilizado o algoritmo de Haar Cascade. Para testes foram utilizadas as bases de dados: JAFFE, CK+ e FacesDB. Além disso, foi realizado um teste de vídeo em tempo real com 12 voluntários. A taxa média de acerto para detecções em base de dados foi, para cada emoção de: neutro(85,3%), felicidade (93%), tristeza (45,33%), raiva (abaixo de 35%), medo (60%), surpresa (80,33%), nojo (27%). A taxa média de acerto em tempo real foi para cada emoção de: neutro(20%), felicidade (57%), tristeza (35%), raiva 30%), medo (39%), surpresa (97%), nojo (66%).

O trabalho de Parada (2021) realiza o desenvolvimento de vários modelos destinados a detecção de expressões faciais, utilizando redes neurais convolucionais, para classificar as 7 expressões. Para construção foi utilizado o conjunto de dados FER2013 e algoritmos pré-treinados, como Resnet-50, Senet-50, VGG16 e FaceNet, aplicando a técnica de *transfer learning* para atingir uma acurácia final de 76,01%.

3 METODOLOGIA

O fluxo utilizado para desenvolvimento deste trabalho(figura 11), escolhida a base de dados, testou-se variações da técnica de *transfer learning*(fine tuning, alteração de parâmetros e mudança de pesos iniciais do modelo) para que se chegasse ao modelo com melhor acurácia e performance para o uso em tempo real, dentro do conhecimento adquirido durante o estudo e execução do trabalho.

O pré processamento da imagem consiste no carregamento das imagens em batches, seu redimensionamento para o tamanho aceito pela CNN(224x224) e também a alteração da escala dos seus pixels de 0 a 255 para 0 a 1, conhecido como normalização, que tem o intuito de tornar a imagem mais leve, facilitando seu processamento. A inicialização do modelo pré-treinado consiste em instanciar o modelo antigo com os pesos obtidos através de um treinamento anterior. A criação do novo classificador consiste em instanciar algum novo algoritmo(SVM, Logistic Regression, RNA, etc) para classificar as features extraídas pela CNN. O treinamento consiste em fazer com que o algoritmo final (CNN + classificador) carregue os dados progressivamente e vá otimizando até os limites configurados. A aferição da acurácia é o processo de verificar os resultados do treinamento(acurácia e perda). A aferição manual, consiste em executar o modelo em tempo real no módulo de captura para testes. Por fim, o modelo que obtiver o melhor resultado neste ciclo(baseado na acurácia e perda, quando disponível), será o escolhido como definitivo.

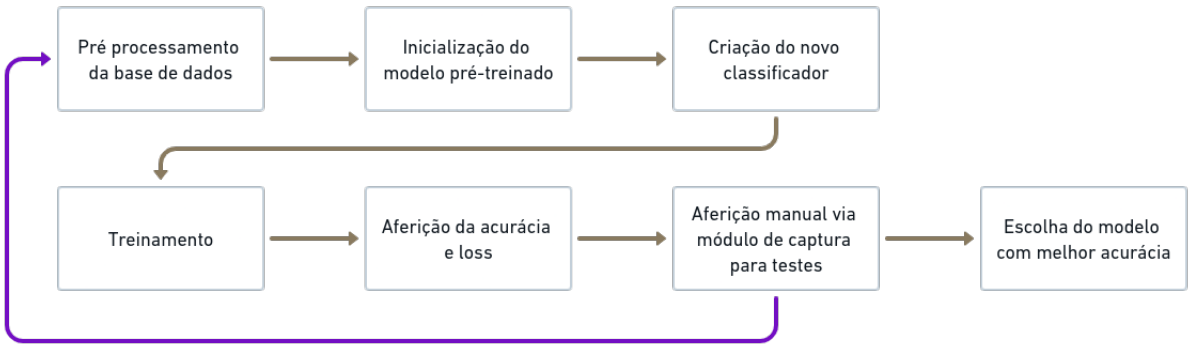

Figura 11 – Fluxo metodológico

Fonte: Autor (2022)

Os algoritmos utilizados neste trabalho foram: o algoritmo de Haar Cascade, que comumente é utilizado para a detecção de objetos, e neste caso foi utilizado para a detecção de faces. A implementação utilizada foi a padrão da biblioteca OpenCV para a linguagem python python, que já traz a detecção de faces. Durante a fase de desenvolvimento foram

utilizados três algoritmos para fazer uma análise exploratória(conforme descreve a seção 3.1), sendo elas o SVM, Logistic Regression e uma RNA. Os dois primeiros, vieram da biblioteca Sklearn, que foca em facilitar o desenvolvimento de sistemas de aprendizado de máquina. O processo ocorreu até chegar a versão mais acurada possível, dentro do escopo deste trabalho. Além do desenvolvimento do algoritmo de classificação foi produzido um módulo capaz de utilizar o produto final para que um usuário final seja capaz de utilizá-lo para os fins de teste de software.

Para todo o desenvolvimento do software, foi utilizada a linguagem de programação Python, devido às bibliotecas utilizadas para construção e uso do modelo. Foi utilizada a biblioteca OpenCV, por disponibilizar todas as ferramentas necessárias para realizar o tratamento digital de imagens e possui algoritmos de visão computacional, tal qual o Haar Cascade. Os algoritmos SVM e Logistic Regression foram providos pela biblioteca SKlearn, que tem todo o ferramental para desenvolvimento de aplicações de aprendizado de máquina. Para todo o trabalho relativo à redes neurais foi utilizada a biblioteca Keras, que também dispões algoritmos pré-treinados, como o VGG-16 e o ResNet50. Ambas facilitam o processo de desenvolvimento, pois implementam muito do ferramental a ser utilizado.

3.1 CLASSIFICADOR DE EMOÇÕES

3.1.1 Base de dados

A base de dados Kaggle (2021) é uma base de dados pública, que contém rostos de pessoas categorizados por diferentes expressões faciais. As imagens possuem dimensão de 48x48 pixels, sendo imagens de rostos centralizados, variadas em sexo, idade e obstruções(óculos). Também são divididas em 7 emoções: raiva, nojo, medo, felicidade, tristeza, surpresa e neutro. Por padrão o conjunto de treinamento e validação já vem separado, sendo para treinamento 28709 exemplos e para validação 3589 exemplos. Sendo eles divididos em 3995, raiva; 436, nojo; 4097, Medo; 7215, Felicidade; 4965 Neutro; 4830 Tristeza e 3171, surpresa. A figura 12 possui um gráfico que apresenta a distribuição dos exemplos.

Um detalhe para este conjunto é que não possui muitos dados, o que dificulta um treinamento com maior qualidade. Além disso, as classes estão distribuídas de forma desbalanceada, fazendo com que o resultado final acabe sendo enviesado para as classes com maior peso.

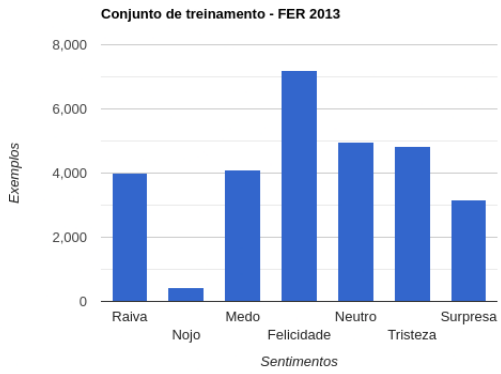

Figura 12 – Distribuição dos exemplos do FER2013

Fonte: Autor (2022)

3.1.2 Métodos e experimentos

Para que a detecção e classificação de emoções faciais seja possível, as arquiteturas VGG-16 e ResNet 50 são utilizadas para o treinamento de modelos de aprendizagem profunda. Entretanto, para que o treinamento seja mais eficiente, muitos dados são necessários, na casa dos milhões. O que não é o caso para este trabalho que utiliza para treinamento um conjunto de dados abaixo de 30 mil imagens. Diante do fato a utilização da técnica de *transfer learning* é uma alternativa.

Para a construção da CNN foi inicialmente necessário utilizar as bases de imagens para treinar a CNN. Foi realizado um benchmark, treinando os algoritmos de VGG16 (SIMONYAN; ZISSERMAN, 2015) com vários padrões de *transfer learning* diferentes, de modo que o que obtivesse melhor desempenho fosse selecionado como classificador para a aplicação. Neste caso foi utilizada a base de dados FER2013, buscando classificar as 7 emoções do conjunto de dados para treinamento: raiva, nojo, medo, felicidade, tristeza, surpresa e neutro.

Para realizar o treinamento foi utilizada a plataforma Google Collab, na sua versão pro. Através dela é possível ter acesso a mais memória e GPUs com mais poder de processamento, o que faz com que seja possível processar mais dados em uma velocidade maior.

Neste trabalho foi feita uma análise exploratória, passando da técnica mais simples até a mais avançada(fine-tuning) para entender melhor o resultado e a performance de cada.

3.1.3 VGG16 com pesos ImageNet e classificadores SVM e Logistic Regression

Essa primeira análise buscou implementar a forma de *transfer learning* mais inicial. Neste caso, os pesos utilizados na rede foram os da ImageNet, como o próprio Rosebrock (2019), sugere iniciar com feature extraction, e caso não obtenha um resultado adequado caminhar para fine-tuning e suas várias possibilidades.

O conjunto de dados utilizado originalmente não possui um tamanho compatível com com a entrada de dados do VGG16(224x224 pixels), dessa forma ao fazer o carregamento de dados, foi utilizado a classe ImageDataGenerator e seu método flow_from_directory, que permite que as imagens sejam carregadas ao poucos na memória em certas quantidades(no caso foi selecionadas 128) para não estourar todo o uso da mesma, também permite com que ao serem carregadas elas possam ser redimensionadas para o tamanho correto, além de gerar imagens adicionais, utilizando as originais e gerando cópias com rotações.

A primeira etapa para a classificação é a extração de características nos conjuntos de dados de treinamento e validação. O modelo VGG16, processou cada um dos exemplos, e os persistiu em um arquivo CSV para cada (treino e teste). A primeira coluna do arquivo CSV guarda qual a classificação correta para aquele exemplo, e a segunda coluna traz a lista de características gerada pelo VGG16.

Para que o VGG16 gere apenas as características ao invés da classificação, para o modelo original(imagenet) deve ser feita uma alteração na arquitetura do modelo(conforme ilustra figura 13), onde é removida a camada totalmente conectada da rede convolucional, que é onde acontece a classificação de acordo com as características geradas pelas camadas anteriores da rede neural. Como não há interesse na classificação original, devido estarmos transferindo o conhecimento obtido para outro domínio de classificação, apenas nos interessa as características que os filtros da CNN geraram, conforme demonstra a figura a seguir.

Com os arquivos CSVs já populados com as features extraídas pelo VGG16, é possível utilizá-los como insumo para treinar os algoritmos escolhidos. Nesta primeira etapa dois algoritmos de machine learning comuns foram utilizados, o SVM e o Logistic Regression. Um detalhe é que o fluxo não é ligado diretamente, o fluxo de extração de características passa por uma "pausa" que é a geração do CSV, que em um segundo momento é consumido pelo novo classificador.

3.1.4 VGG16 com pesos ImageNet e fine-tuning

Como segundo experimento, foi usada a técnica de fine-tuning, que se assemelha à anterior, em que o classificador original é removido, e nesse caso o substituiremos por uma rede neural. Neste caso a rede neural nova não precisará gerar o arquivo com as features intermediário, visto que ela funcionará de forma semelhante a rede original, já retornará

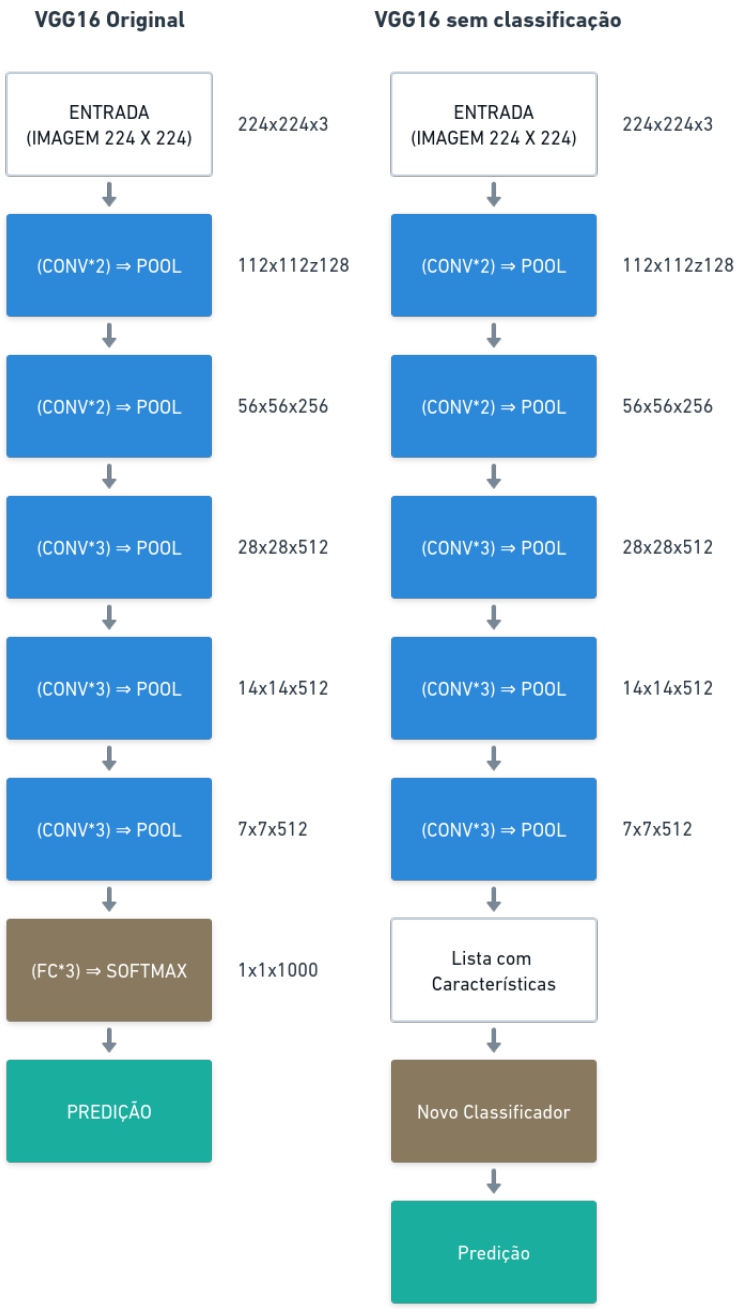

Figura 13 – VGG16 original a esquerda e modificado para extração de características a direita

Fonte: Rosebrock (2019) Adaptada (2022)

a previsão de classe. Outra modificação importante é que as duas últimas camadas de convolução da rede original foram "descongeladas", o que significa que elas agora serão re-treinadas, terão seus pesos alterados, durante o novo treinamento.

A rede final consiste em uma camada de planificação do vetor de features, seguida de 2 blocos de camadas totalmente conectadas, sendo elas compostas por uma camada

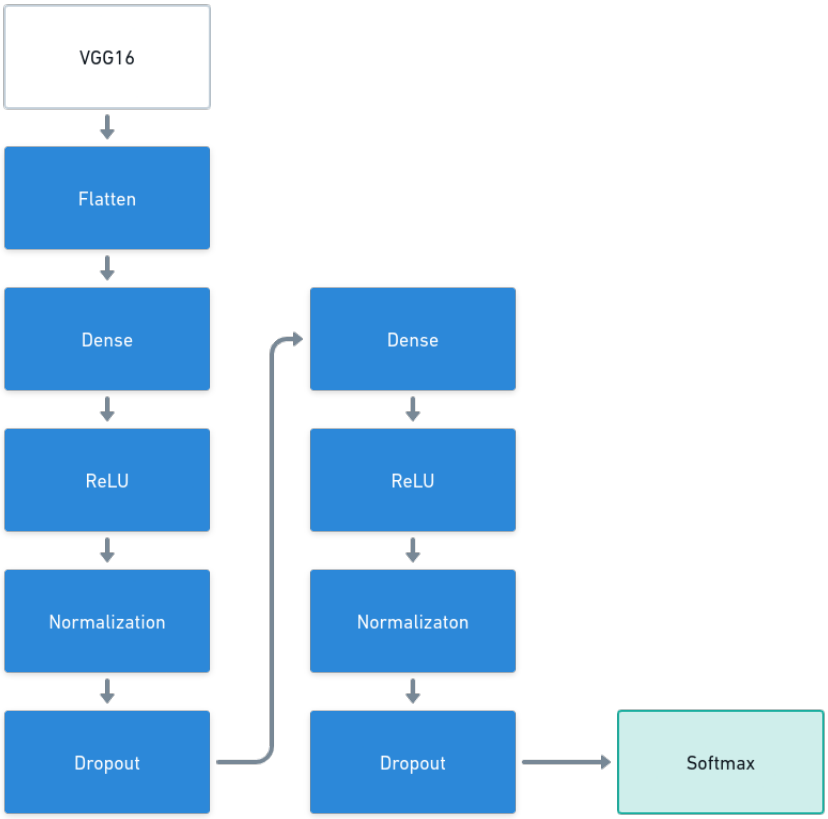

Figura 14 – Arquitetura da RNA para classificação

Fonte: Rosebrock (2019) Adaptada (2022)

densa, seguida por uma camada de ativação (ReLu), uma camada de normalização e um dropout. Ao fim vem uma camada de ativação softmax. Conforme identificado na figura 14.

3.1.5 VGG16 com pesos VGGFace e fine-tuning

Este experimento seguiu o mesmo padrão do anterior, com a diferença dos pesos utilizados na rede pré-treinada, foram utilizados os pesos do modelo Keras VGGFace (MALLI, 2021). Também foi utilizado o VGGFace através da biblioteca keras-vggface que provém a arquitetura VGG16, dentre outras, pré-treinados na base de dados. Foi encontrado o uso deste modelo pré-treinado no projeto de mestrado de Parada (2021). No trabalho original o melhor desempenho foi com o modelo ResNet50, porém tive problemas ao treinar o algoritmo, principalmente relacionados a estouro de memória ram ao carregar o conjunto de dados.

3.2 MÓDULO DE GRAVAÇÃO DE SESSÕES

Para o uso do classificador pelo usuário final, foi necessário desenvolver um software capaz de gravar as sessões de teste e ao mesmo tempo realizar a classificação em tempo real. Para o desenvolvimento foi utilizada a linguagem Python, devido já ser usada para o desenvolvimento dos modelos de aprendizado de máquina que realiza a classificação das emoções faciais. A biblioteca utilizada para desenvolver a interface gráfica foi o Tkinter, por ser nativa da linguagem de programação, além disso as bibliotecas OpenCV e o Keras foram utilizados, para realizar a captura e pré-processamento da imagem da webcam e o carregamento do modelo treinado e sua execução, respectivamente.

O hardware utilizado nos testes foi um desktop com processador Intel® CoreTM i5 de décima geração de 3.90GHz, memória RAM(Random Access Memory) de 16GB e sistema operacional Manjaro Linux. Para captura das imagens foi utilizada uma webcam Logitech® C920s com resolução Full HD.

Inicialmente o desenvolvimento do módulo seria um sistema web, devido a maior flexibilidade e capacidade de acesso aos relatórios de qualquer máquina, porém devido ao tempo limitado junto ao escopo mais amplo do projeto(desenvolver o classificador e o módulo), junto a complexidade maior de se desenvolver um sistema web, que tenha que acessar e fazer streaming da imagem, optou-se pelo desenvolvimento de um cliente desktop, pois seria mais fácil acessar os recursos da máquina e se evitaria questões relativas à alocação de infraestrutura de servidores e autenticação.

Pensando nos pontos já citados a idéia do cliente desktop é ter os recursos mínimos necessários, sendo mais próximo de uma prova de conceito, definidos na Tabela 1 e os requisitos não funcionais na Tabela 2.

Tabela 1 – Requisitos funcionais

ID	**Requisitos funcionais**
RF1	Executar a classificação de emoções faciais em tempo real.
RF2	Executar a classificação de emoções faciais em tempo real e realizar a gravação do vídeo.
RF3	Para as sessões gravadas, guardar a emoção predominante ao fim de cada sessão.
RF4	Ser possível acessar sessões gravadas(assistir vídeo e ver emoção predominante).

Fonte: Autor (2022)

Tabela 2 – Requisitos não funcionais

ID	**Requisitos não funcionais**	**Descrição**
RNF1	Simplicidade	O Sistema deverá ter uma interface simples e clara.
RNF2	Compatibilidade	O sistema deverá ser capaz de executar a classificação apenas com a CPU, para que seja possível executar em configurações menos robustas.

Fonte: Autor (2022)

4 RESULTADOS

Este capítulo apresenta os resultados alcançados, por este trabalho, apresentando os resultados das atividades realizadas e os artefatos gerados.

Os algoritmos de SVM e Logistic Regression, que utilizaram as *features* extraidas através do VGG16(com os pesos ImageNet), tendo resultado inferior as próximas iterações. A acurácia do SVM foi de 49% e o Logistic Regression de 50%, para ambos não há *loss*.

Inicialmente VGG 16 com as duas últimas camadas desbloqueadas para fine-tuning no primeiro treino com o middleware de *early stopping* com paciência igual a 5. O parâmetro de *early stoping*, faz com que caso o valor de um determinado campo(neste caso o *loss* de validação) não evolua durante um número de épocas determinado(paciência) o algoritmo irá encerrar, antes do total de épocas total. Isso ajuda a prevenir o overfitting. A figura 15 mostra o resultado.

Os paramêtros para a avaliação dos resultados dos modelos, são a acurácia - o quão os valores obtidos pela previsão do modelo é verdadeiro em relação aos dados de treinamento e validação - e o *loss* - que mede o grau de erro nas previsões. Para o primeiro, valores próximos de 100% são os desejados e para o segundo valores mais próximos de zero são desejados.

O parâmetro de *early stopping* foi teve de a paciência(espera) de 5 para 10 e obteve o seguinte resultado, apresentado na figura 16.

Em relação à acurácia e o *loss*, percebe-se que através dos gráficos na figura 15, em relação a figura 16, que houve uma piora, devido a pouca alteração na acurácia, mas um significativo aumento no *loss*. Após essas mudanças, notou-se que o middleware de *early stopping* estava fazendo com que o treinamento encerrasse muito cedo, devido o aumento seguido do *loss* de validação. Ele foi removido para testar a hipótese de que o *loss* iria diminuir com o passar das épocas de treinamento. A remoção não surtiu o efeito esperado e acabou gerando apenas o overfitting.

O projeto de mestrado de Parada (2021) faz o comparativo de várias redes fazendo o benchmarking entre elas. Pelas razões explicadas anteriormente, foi escolhido o VGG16 com os pesos do VGGFace. No conjunto de validação o modelo obteve uma acurácia aproximada de 63% e um *loss* de 1.0339.

Durante os testes, o classificador executando no módulo de gravação de sessões, a emoção “nojo” dificilmente foi detectada, provavelmente devido ao fato de ser a com menos amostras no conjunto de treino e ao mesmo tempo ter algumas amostras que são bastante semelhantes a da emoção “raiva”, como demonstra a figura 18.

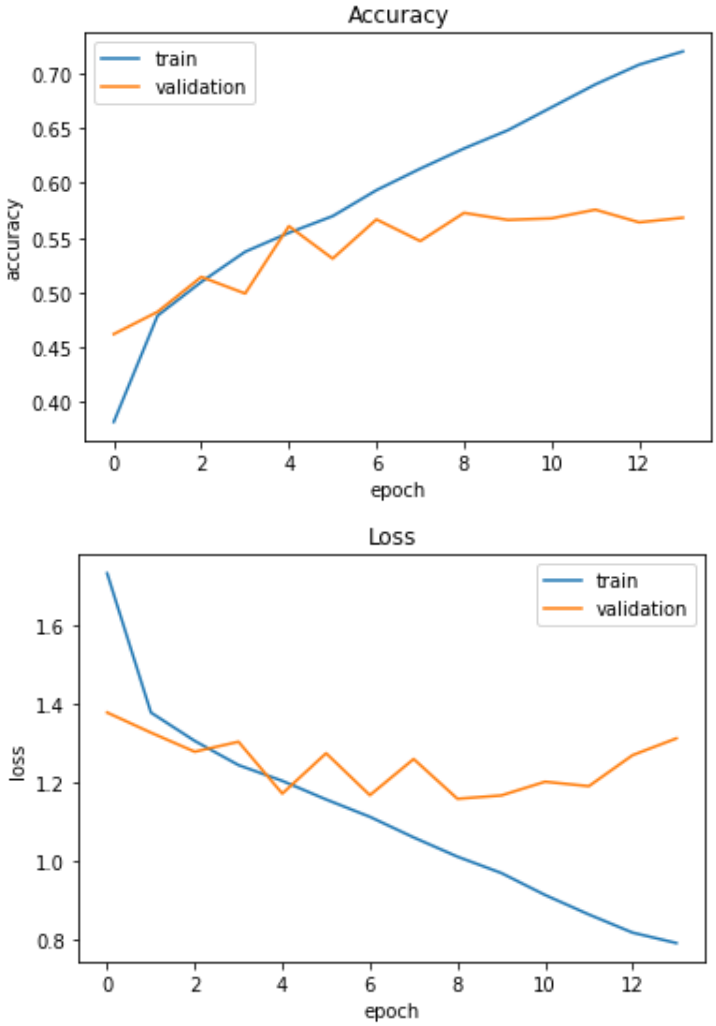

Figura 15 – Acurácia(acima) e *Loss*(abaixo) dos modelo de VGG16 com fine tuning - teste 1

Fonte: Autor (2022)

Nos testes manuais a emoção “medo” também dificilmente é detectada, normalmente só é detectada quando há alguma inclinação da cabeça.

As emoções felicidade, neutro, medo, surpresa e raiva conseguiram ser detectadas, conforme apresenta a figura 19

Outro problema encontrado na detecção de emoções é a oclusão, fazendo testes manuais foi percebido que o algoritmo teve mais oscilações quando o usuário tinha alguma oclusão na face, conforme demonstra a figura 20.

A aplicação conseguiu cumprir com os requisitos funcionais propostos, foi criada a tela para a execução do modelo em tempo real(sem gravação), assim como a gravação e o acesso a mesma, tanto via o próprio software, que executa o vídeo e traz os dados da

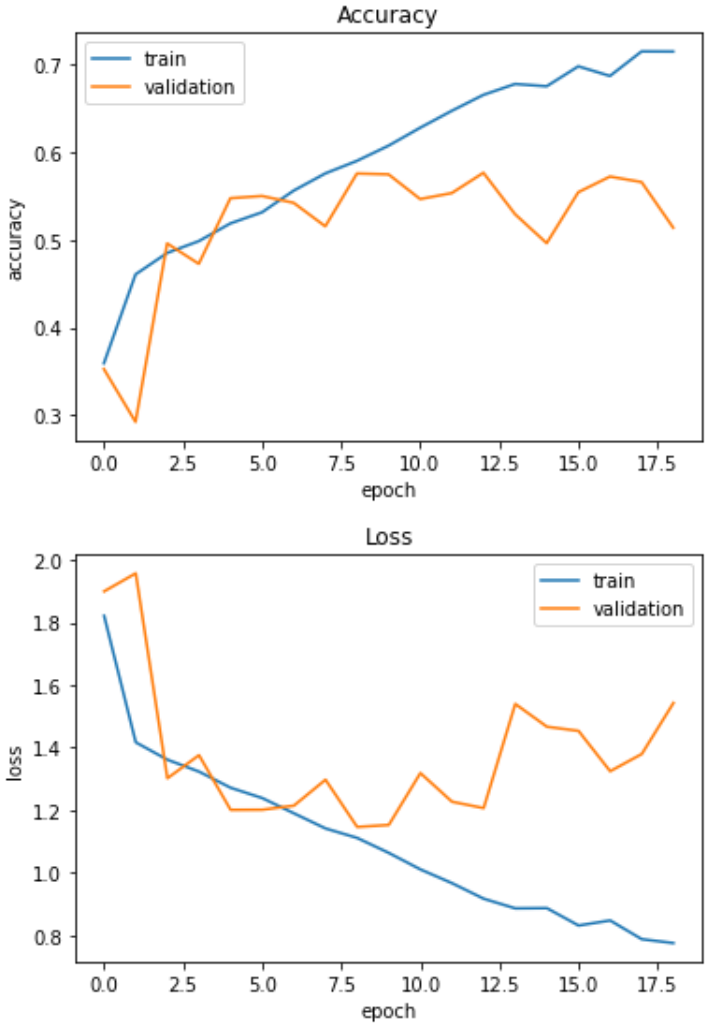

Figura 16 – Acurácia(acima) e *Loss*(abaixo) dos modelo de VGG16 com fine tuning - teste 2

Fonte: Autor (2022)

emoção que mais apareceu, assim como a distribuição percentual das expressões durante a sessão de testes gravada. As figuras 21 a 25 mostram todas as telas do sistema.

Todas as gravações das sessões ficam armazenadas em uma pasta subpasta dentro do diretório do programa, a pasta “sessions"essa por sua vez armazena uma pasta para cada sessão de testes, que contém o vídeo da gravação dos testes e um arquivo CSV com as emoções preditas, organizadas por ordem de detecção, para que a partir delas seja possível executar análises estatísticas.

Um problema enfrentado foi a gravação do vídeo, em que o vídeo fica mais rápido durante sua execução do que a imagem em tempo real durante o teste, provavelmente por algum desalinhamento do parâmetro de FPS(frames por segundo) na gravação, houve várias alterações nos parâmetros de gravação, porém sem ganhos significativos. O código

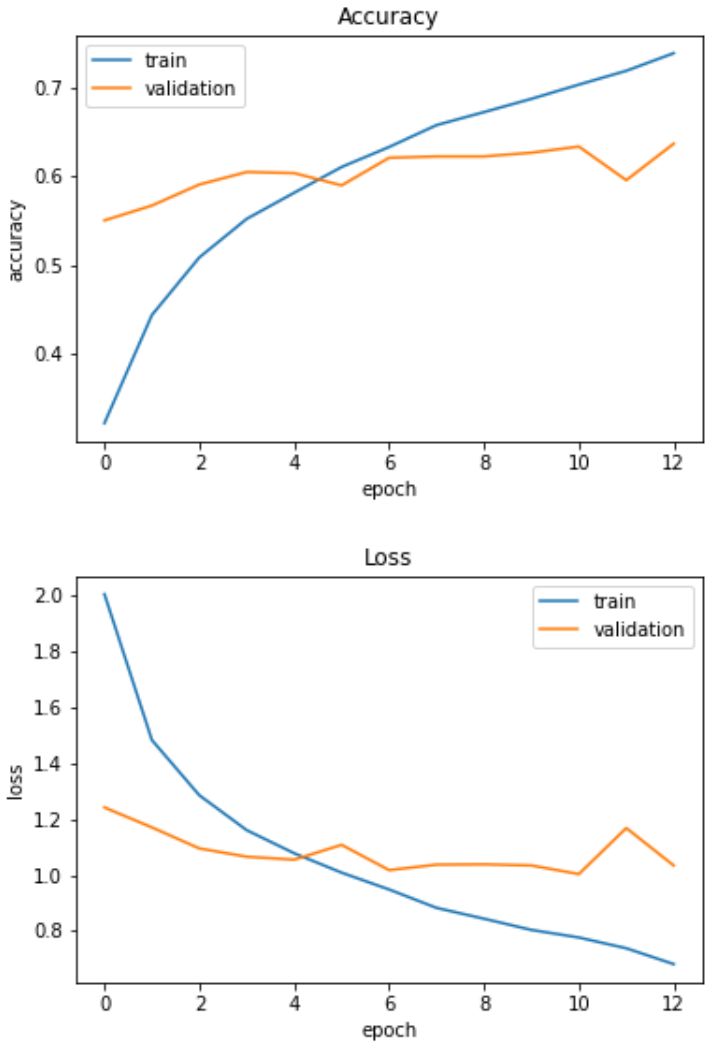

Figura 17 – Acurácia(acima) e *Loss*(abaixo) dos modelos treinados no VGG16 com pesos do VGGFace

Fonte: Autor (2022)

do sistema pode ser acessado em https://github.com/patrickdeangelis/tcc

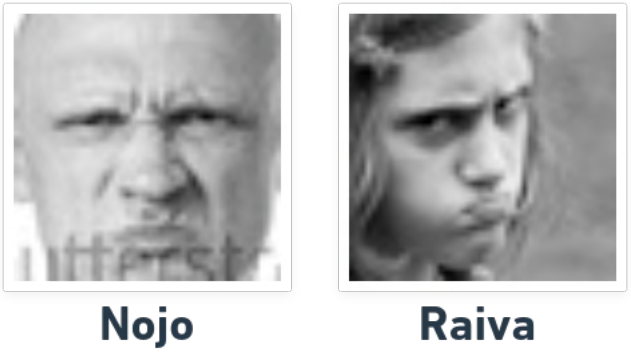

Figura 18 – Exemplos das emoções previstas corretamente pela aplicação

Fonte: Autor (2022)

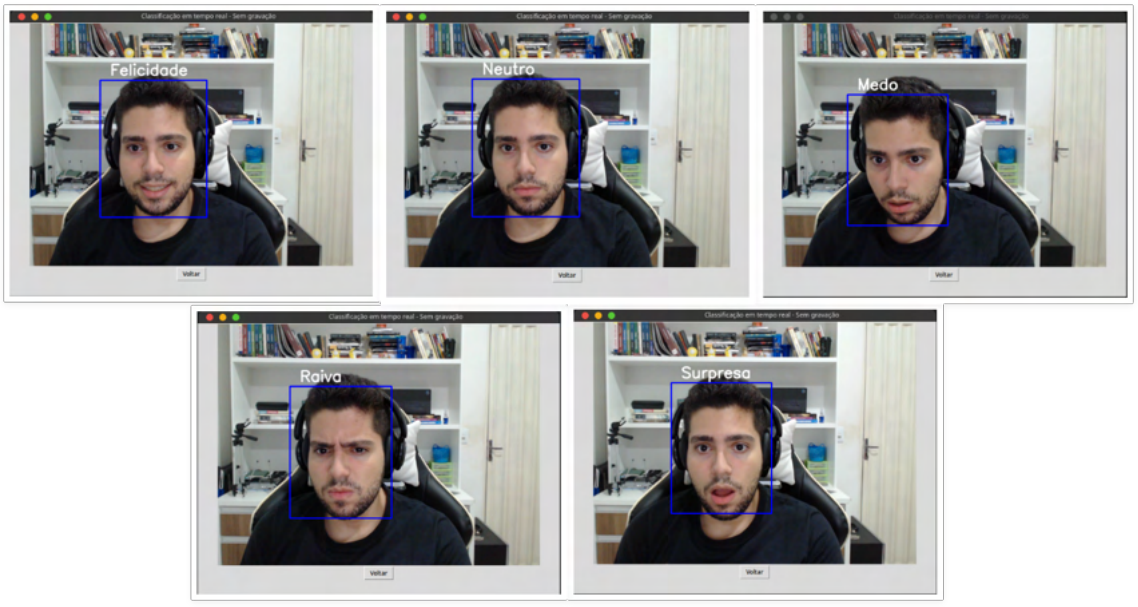

Figura 19 – Exemplos da base dos sentimentos de nojo e raiva da base de dados FER 2013

Fonte: Autor (2022)

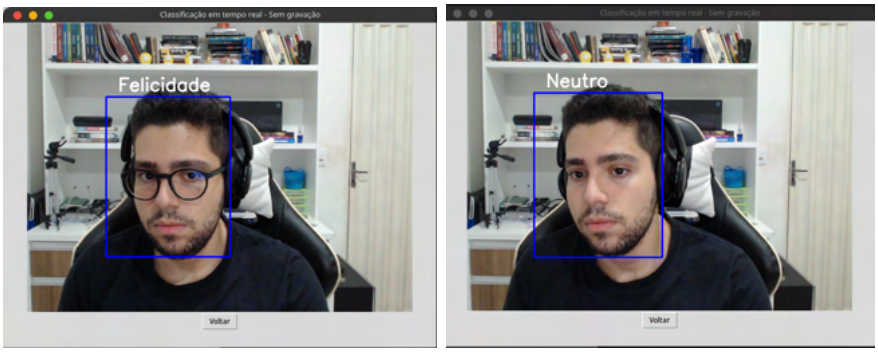

Figura 20 – Emoções detectadas, da esquerda para a direita de cima para baixo: felicidade, neutro, medo, surpresa e raiva a

Fonte: Autor (2022)

Figura 21 – Tela principal

Fonte: Autor (2022)

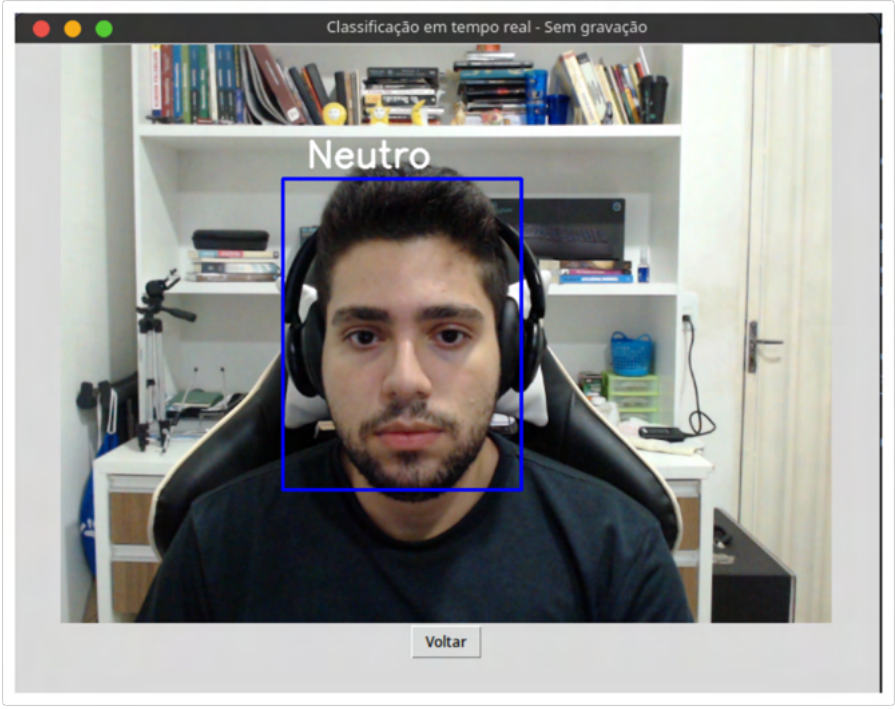

Figura 22 – Tela de classificação em tempo real(sem gravação)

Fonte: Autor (2022)

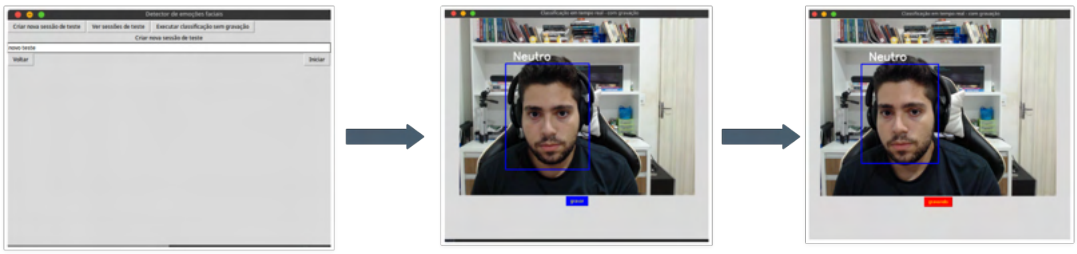

Figura 23 – Telas de gravação de sessão, seguindo fluxo de criação à gravação

Fonte: Autor (2022)

Figura 24 – Tela de listagem de relatórios das sessões

Fonte: Autor (2022)

Figura 25 – Tela de relatório da sessão

Fonte: Autor (2022)

5 CONCLUSÃO

A pesquisa desenvolvida iniciou-se com um estudo bibliográfico sobre visão computacional, expressões faciais e redes neurais. Como contribuição destaca-se o desenvolvimento de um classificador e um módulo de captura de imagens capaz de realizar análise em tempo real. O objetivo foi alcançado com algumas resalvas: o modelo apresenta instabilidade pelo fato de não conseguir detectar a emoção de nojo e ter dificuldade para classificar a emoção medo. Mesmo assim as demais emoções emoções conseguiram ser detectadas.

O Classificador teve os problemas citados principalmente por ser uma base com poucas amostras e estar desbalanceada. Na classificação em tempo real houve uma variacão na classificação devido a movimentações faciais.

O Módulo de captura conseguiu ser desenvolvido com sucesso, atingindo os objetivos propostos, de executar a classificação em tempo real e o cadastro e leitura das sessões de teste, disponibilizando o vídeo da sessão e também o CSV com as imagens. Porém a gravação teve a velocidade do vídeo maior devido a um bug de gravação que não foi detectado.

5.1 TRABALHOS FUTUROS

A partir deste trabalho pode-se trilhar duas trilhas de aprimoramento e continuidade da pesquisa.

A primeira é o aprimoramento e até mesmo o desenvolvimento de modelos mais acurados e com menor perda, seja utilizando novas arquiteturas, conjuntos de dados e padrões diferentes, mas que mantenham a performance para que possam ser utilizados em tempo real.

A construção de novas bases de dados também é muito importante, quanto mais dados existirem, mais insumos os algoritmos(independente de qual) terão, e será possível entender melhor as características que formam as classes preditas.

Por fim, pode-se criar um sistema para gerenciamento das sessões de teste mais avançado, conforme pensado inicialmente, onde seja possível fazer um sistema web, com clientes móveis, de modo que até o celular possa ser utilizado para registrar a sessões e controlá-las remotamente, como também deixar o acesso aos testes disponíveis em qualquer máquina com acesso a internet.

REFERÊNCIAS

AGARWAL, M. et al. Face recognition using principle component analysis, eigenface and neural network. In: **2010 International Conference on Signal Acquisition and Processing**. [S.l.: s.n.], 2010. p. 310–314. Citado na página 18.

ALMARRI, S. B. S. **Real-Time Facial Emotion Recognition Using Fast R-CNN**. Dissertação (Mestrado) — SAN JOSE STATE UNIVERSITY, 2019. Citado na página 20.

ARAúJO, e. F. H. D. Redes neurais convolucionais com tensorflow: Teoria e prática. sociedade brasileira de computaÇÃo. **III Escola Regional de Informática do Piauí. Livro Anais-Artigos e Minicursos, v. 1**, p. 382–406, 2017. Citado nas páginas 23, 24 e 25.

BACKES, A. R.; JUNIOR, J. J. de M. S. **Introdução à visão computacional**. [S.l.]: Casa do Código, 2018. Citado nas páginas 16 e 17.

BARELLI, F. **Introdução à visão computacional**. [S.l.]: Casa do Código, 2018. Citado nas páginas 16, 17, 18 e 19.

BERG, C. H. **Ferramenta para identificação de emoções a partir de onomatopeias para pessoas com diferentes habilidades visuais**. Tese (Doutorado) — Universidade de Federal de Santa Catarina, Florianópolis, 2017. Citado na página 21.

CHIU, K.; RASKAR, R. Computer vision on tap, computer vision and pattern recognition workshops. **CVPR Workshops 2009. IEEE Computer Society Conference on**, p. 31–38, 2009. Citado na página 13.

COSSETI, M. J. **Reconhecimento De Expressões Faciais Utilizando Redução De Dimensionalidade Para Estratégia De Classificação Um-Contra-Um**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Paran, Curitiba, 2015. Citado na página 20.

CROWLEY, J. L.; CHRISTENSEN, H. I. **Vision as Process**. [S.l.]: Springer-Verlag, 2011. Citado na página 16.

EKMAN, P. Universals and cultural differences in facial expressions of emotion. **Nebraska Symposium on Motivation**, p. 207–283, 1972. Citado na página 20.

EKMAN, W. V. F. P. Pictures of facial affect. **Consulting Psychologists Press**, 1976. Citado na página 20.

FARIAS, R. de S. **Aplicação para detecção e reconhecimento de expressões faciais com redes neurais convolucionais**. Patos, 2019. Citado na página 27.

FERREIRA, K. G. **Teste de Usabilidade**. [S.l.], 2002. Citado na página 21.

FONSECA, F. O. G. da. **Detector de faces utilizando filtros de características**. 111 p. Dissertação (Mestrado) — Universidade de Federal Fluminense, Niteri, 2016. Citado na página 18.

GOUVEIA, W. da R. **Detecção de Faces Humanas em Imagens Coloridas Utilizando Redes Neurais Artificiais**. Dissertação (Mestrado) — Universidade Federal de São Carlos, São Carlos, 2010. Citado na página 18.

GUEDES, A. B. S. **Reconhecimento de Gestos usando Redes Neurais Convolucionadas**. Dissertação (Mestrado) — Universidade Federal de Brasília, Brasília, 2017. Citado nas páginas 21 e 22.

HAGAN HOWARD B. DEMUTH, M. H. B. r M. T. **Neural network design**. [S.l.]: Martin Hagan, 2014. Citado na página 22.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. Porto Alegre: Bookman, 2001. Citado na página 22.

HIX, D.; HARTSON, H. R. Developing user interfaces, ensuring usability through product process. **New York: John Wiley Sons, Inc.**, 1993. Citado na página 13.

JAIN, D. K.; SHAMSOLMOALI, P.; SEHDEV, P. Extended deep neural network for facial emotion recognition. **Pattern Recognition Letters**, v. 120, p. 69–74, 2019. ISSN 0167-8655. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016786551930008X>. Citado na página 19.

KAGGLE. **FER 2013**. 2021. Disponível em: <https://www.kaggle.com/msambare/fer2013>. Acesso em: 4 mar. 2022. Citado na página 29.

LI, S. Z.; JAIN, A. K. **Handbook of Face Recognition (2nd ed)**. [S.l.]: Springer Publishing Company, Incorporated, 2011. Citado na página 18.

LIBRALON, G. L. **Detector de faces utilizando filtros de características**. Tese (Doutorado) — Universidade de São Paulo, São Carlos, 2014. Citado na página 20.

MCCULLOCH, W. P. W. S. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, p. 115–133, 1943. Citado na página 22.

MENDONçA, T. D. B. **Sistema de reconhecimento de expressões faciais para classificação de emoções de usuários em sistemas computacionais**. Russas, 2018. Citado na página 27.

PARADA, D. M. P. **Reconhecimento de expressões faciais compostas em imagens 3D: ambiente forçado vs ambiente espontâneo**. Dissertação (Mestrado) — Universidade de Federal do Paran, Curitiba, 2017. Citado nas páginas 13 e 20.

PARADA, D. M. P. **Improving Facial Emotion Recognition with Image processing and Deep Learning**. Dissertação (Mestrado) — SAN JOSE STATE UNIVERSITY, 2021. Citado nas páginas 27, 33 e 36.

PERUMANOOR, T. J. **What is VGG16? — Introduction to VGG16**. 2021. Disponível em: <https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615>. Acesso em: 4 mar. 2022. Citado na página 26.

ROSEBROCK, A. **Transfer Learning with Keras and Deep Learning**. 2019. Disponível em: <https://pyimagesearch.com/2019/05/20/transfer-learning-with-keras-and-deep-learning/>. Acesso em: 4 mar. 2022. Citado nas páginas 26, 31, 32 e 33.

RUBIN, J. **Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests**. New York: John Wiley Sons, Inc., 2018. 330 p. Citado na página 21.

SAMER, C. H.; RISHI, K.; ROWER. **Image Recognition Using Convolutional Neural Networks**. [S.l.], 2015. 1-12 p. Citado nas páginas 24 e 25.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. arXiv, 2015. Citado nas páginas 26 e 30.