Identificação e classificação de emoções a partir de expressões faciais utilizando redes neurais convolucionais

Davi Daniel Lourenco Sena
Centro de Ciências Computacionais - C3
Universidade Federal do Rio Grande - FURG
Rio Grande, Rio Grande do Sul, Brasil
davi.lsena@hotmail.com

Diana F. Adamatti
Centro de Ciências Computacionais - C3
Universidade Federal do Rio Grande - FURG
Rio Grande, Rio Grande do Sul, Brasil
dianaada@gmail.com

***Resumo*—Este artigo apresenta o processo completo de desenvolvimento do trabalho de conclusão de curso em engenharia de computação, que consistiu em criar um sistema de classificação de sete emoções humanas a partir de expressões faciais utilizando uma rede neural convolucional (Convolutional Neural Network - CNN) e imagens de diferentes bancos de dados. O projeto foi dividido em várias etapas, começando com uma revisão sistemática da literatura (RSL) que explorou desde conceitos básicos até pesquisas mais avançadas em inteligência artificial e computação afetiva, bem como projetos práticos relacionados. Com base nas informações coletadas, o desenvolvimento do sistema foi realizado seguindo as recomendações da literatura e adaptando a metodologia às necessidades do projeto. Foi necessário coletar imagens de voluntários para realizar diferentes inferências de resultados a partir de diferentes bancos de dados, e também incluir esses voluntários nos objetivos do projeto para que entendessem como suas imagens seriam utilizadas e a importância do papel que desempenhariam. Ao final do projeto, foram identificadas algumas limitações técnicas que impactaram o processo, incluindo a necessidade de um pré-processamento mais estruturado dos dados de entrada, limitações de hardware e software, e a qualidade variada do banco de dados utilizado. No entanto, essas limitações serviram como aprendizados valiosos e oportunidades para aprimorar futuros trabalhos na área. Além disso, espera-se que este projeto traga esclarecimentos sobre as possibilidades da pesquisa em computação afetiva e seja acessível por trazer adaptações em português de textos que geralmente estão disponíveis apenas em inglês.**

***Palavras-chave*—Facial Expression Recognition, FER, Convolutional Neural Network, CNN, reconhecimento de expressão facial, rede neural convolucional.**

I. Introdução

As expressões faciais são uma forma não verbal de comunicação das emoções humanas. Durante décadas, a decodificação dessas expressões tem sido objeto de pesquisa na psicologia [1], bem como na área de interação humano-computador [2]. Recentemente, os avanços tecnológicos na análise biométrica, aprendizado de máquina e reconhecimento de padrões, juntamente com a ampla disseminação de câmeras, têm desempenhado um papel fundamental no desenvolvimento de tecnologias de Reconhecimento de Expressões Faciais (*Facial Expression Recognition* - FER) [3].

A FER é uma metodologia utilizada para analisar expressões de sentimentos por diferentes fontes, como fotos e vídeos. Pertence a família de metodologias muitas vezes referida como *"computação afetiva"*, um campo multidisciplinar de pesquisa em capacidades do computador para reconhecer e interpretar emoções humanas e estados afetivos, amplamente baseando-se em tecnologias de Inteligência Artificial (IA) [3] e utilizando técnicas como as redes neurais convolucionais (*Convolutional Neural Network* - CNN).

Diversas áreas podem ser impactadas pela FER, incluindo tornar os carros mais seguros e personalizados; pesquisa de mercado para identificar as necessidades dos consumidores; e até entrevistas de candidatos para avaliar as expressões faciais e traços de personalidade.

Percebendo a importância que assuntos relacionados à IA tem, é de interesse que o tema seja abordado de forma mais aprofundada durante a graduação nas áreas da tecnologia, como o estímulo a participação em projetos e disseminação de conhecimento. O compromisso de todos os membros do ambiente universitário é estimular a pesquisa e explorar qualquer possibilidade, seguindo as normas éticas e legais, que permita elevar o conhecimento dos colegas.

Por conta disso, a proposta do projeto se baseou em **construir um sistema que utilizasse uma CNN capaz de discriminar em sete classes diversas expressões faciais distintas com uma acurácia de 60% ou superior**. Para aprofundamento prático e teórico do campo em questão, foram utilizados como principais referências, trabalhos de outros pesquisadores [4], [5]. Além disso, como parte da responsabilidade de gerar estímulo acadêmico, foram integrados voluntários para testes dos modelos criados, sendo necessário, no mínimo, cinco participantes, os quais receberiam esclarecimentos e explicações sobre toda a proposta do projeto até o resultado esperado.

II. Embasamento Teórico

A. Emoções e Expressões Faciais

Antes de determinar o escopo e os limites do projeto, buscou-se compreender como as emoções e suas expressões

faciais podem ser generalizadas, considerando a variação étnica e racial [6]. É importante entender que essas expressões nem sempre são autênticas e isso afeta significativamente os resultados dos testes e validações. Portanto, é essencial ter cuidado com estudos sobre microexpressões, pois são difíceis de aplicar o método científico devido à sua subjetividade. Com isso, foi possível delimitar com segurança sete grupos de expressões faciais: raiva, neutro, nojo, medo, felicidade, tristeza e surpresa (Figura 1).

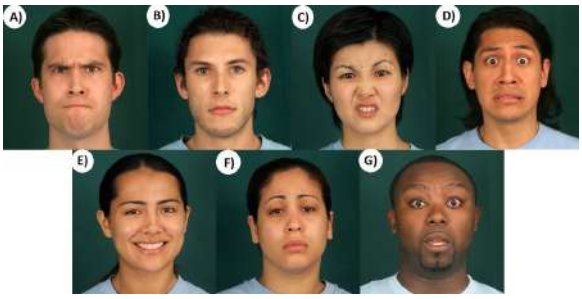

Figura 1. Emoções e suas expressões faciais [7]:
A) raiva; B) neutro; C) nojo; D) medo
E) felicidade; F) tristeza; G) surpresa

B. Visão Computacional

A visão computacional, campo da IA, possui diversas técnicas capazes de transformar dados de imagem em informações significativas, simulando a visão e aprendizado humano. Com isso é possível realizar a detecção de objetos, identificar padrões ou analisar contextos numa determinada imagem a partir do carregamento de modelos construídos anteriormente [8]. A Figura 2 representa a aplicação de visão computacional na detecção da região facial, tendo o rosto do autor como área de interesse.

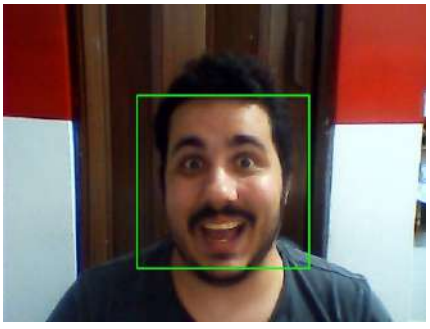

Figura 2. Região facial demarcada através de técnicas de visão computacional

C. Redes Neurais Convolucionais (Convolutional Neural Networks - CNN)

CNN é um tipo de rede neural artificial profunda frequentemente aplicada para lidar com tarefas nas quais os dados possuem altas correlações locais, como imagens visuais, previsão de vídeo e categorização de texto, pois essa rede específica pode capturar o mesmo padrão localizado em diferentes regiões [9]. Basicamente, o compartilhamento de peso aplica traduções de invariância no modelo de rede neural para auxiliar na filtragem do recurso de aprendizagem, independentemente das propriedades espaciais [10]. A ideia da rede neural é simular um cérebro humano e fazer tomadas de decisão "inteligentes".

A rede adota um formato *feed-forward*, permitindo codificar informações importantes contidas nos dados de entrada com muito menos parâmetros do que em outros modelos de aprendizado profundo [11]. Sua estrutura padrão é formada por camadas de convolução (*Convolution Layers* ou Conv), camadas de agrupamento (*Pool* ou *Pooling*) e, por fim, camadas totalmente conectadas (*Fully-Connected*) [12]. Começando pela camada de convolução (Figura 3) tem a função de extrair recursos ou características efetivas dos dados de entrada por meio de seus múltiplos *kernels* (janelas) convolucionais e definir pesos que determinam a possível relevância dos dados. A camada de *pooling* recebe os dados da camada convolução e determina quais dessas características são mais relevantes, reduzindo a dimensão dos dados ao remover características menos relevantes. A camada totalmente conectada é formada por um número pré-determinado de neurônios, cada um com um viés inicial. Eles são responsáveis por integrar todos os recursos de cada mapa de características gerado, formando um conjunto global de características. A partir de diversos cálculos internos, é possível determinar padrões nos dados obtidos e definir as chances dos dados de entrada pertencerem a cada classe predeterminada [9].

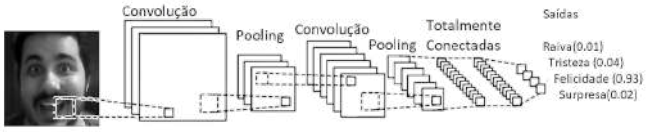

Figura 3. Estrutura de uma rede neural convolucional

Camadas de normalização também podem ser adicionadas para preparar os dados e melhorar o desempenho da rede, junto das camadas de *Dropout* que melhoram a capacidade de generalização e evitam o *overfitting* (sobreajuste), condição onde a CNN consegue classificar muito bem os dados de treino, mas tem baixa acurácia nos dados de validação.

D. Revisão Sistemática

Grande parte do embasamento teórico e prático desta pesquisa se originou através da extensiva análise pela Revisão Sistemática da Literatura (RSL), baseando-se no sistema proposto por Mariano [13], o qual divide a RSL em quatro etapas: definição de protocolo, coleta de referências, avaliação dos dados e interpretação dos resultados. A partir desse sistema foi possível determinar os artigos mais relevantes para o estudo, junto de uma análise dos resultados obtidos de forma que possibilitasse dividir os artigos de diferentes formas, e assim criar estratégias quanto a releitura e qual o propósito a qual serviriam nos primeiros passos do projeto. A Tabela I refere-se a usabilidade dos artigos no projeto, enquanto a Tabela II

refere-se aos bancos de dados utilizados, a acurácia alcançada e qual artigo pertence os resultados.

Tabela I
USABILIDADE DOS ARTIGOS

Referenciais teóricos e métodos diversos	Resumo de técnicas
[14], [15], [16]	[17], [18]
Aplicação de técnicas e métodos avançados	Semelhantes ao projeto
[19], [20], [21], [22]	[23], [24], [21], [14], [15]

Tabela II
BANCOS DE DADOS E ACURÁCIA POR ARTIGO RELACIONADO

Banco de dados	Acurácia
FER2013	51,43% [23], 66% [24] 92% [14], 96,77% [21]
CK+	92,1% [20], 96,62% [23] 99% [16], 99,38% [19]

III. MATERIAIS E MÉTODOS

A. Ambiente de Desenvolvimento e Bibliotecas

Como ambiente de desenvolvimento, foi utilizada a plataforma Kaggle por ser bem reconhecida no ramo da ciência de dados e prover a maioria das bibliotecas necessárias de forma nativa. Foi possível executar testes através do uso de unidades de processamento gráfico (*Graphics processing unit* - GPU), o que diminuiu o tempo de execução drasticamente nas operações de processamento de imagem e modelagem da rede neural. O projeto desenvolvido consta publicado no perfil do autor [25], sendo utilizadas as bibliotecas Keras, Tensorflow, Scikit Learn, Seaborn, Pandas, MatPlotLib, Numpy e OpenCV.

As configurações e especificações utilizadas foram as seguintes:

- Acelerador: 2 GPU Tesla T4 (14.8GB de uso máximo cada)
- RAM: 13GB de uso máximo
- CPU: Não informado.
- Persistência: Arquivos somente.
- Linguagem: Python (Versão 3.7.12)
- Cota máxima Diária/Semanal: 12h/30h

B. Bancos de Dados e Testes

Os bancos de dados utilizados neste estudo foram o FER2013 [26], que foi utilizado no desenvolvimento e validação do modelo; o banco construído a partir de voluntários da entidade estudantil Byte Júnior, que foi utilizado apenas na etapa de testes; e o banco CK+ [27], adaptado e publicado na plataforma Kaggle [28] para trabalhos futuros. A escolha do banco de dados principal foi baseada em diversas considerações, como o volume de dados e a tolerância a imagens de baixa qualidade, conforme apresentado na Tabela III. Optou-se por utilizar um banco de dados que oferecesse as melhores condições para criação do modelo.

Tabela III
BANCOS DE DADOS UTILIZADOS

Banco de dados	Amostras	Considerações
FER2013	35.887	Volumoso; 7 classes de expressões; Imagens ruidosas; Rotação facial
CK+	920	Imagens padronizadas; 8 classes de expressões Baixo volume; Alta qualidade visual
Voluntários	61	Desenvolvido para testes; 7 classes de expressões Baixo volume; Qualidade variada

O banco de dados FER2013 adquirido foi uma versão adaptada do original, publicada também na plataforma Kaggle [29]. As imagens são carregadas a partir de um CSV que contém todos os dados divididos nas colunas *emotion, pixels e usage* (Figura 4) que representam imagens em 48x48 pixels (Figura 5).

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

Figura 4. Banco de dados FER2013 em formato CSV

Figura 5. Amostragem de dados dos banco de dados FER2013

C. Métodos de Detecção Facial

A detecção da face é crucial na FER [30]. Um sistema eficiente deve reconhecer o rosto de forma espontânea em imagens ou vídeos estáticos. As características faciais, como contornos, cor da pele e textura são utilizadas para detectar um rosto em uma sequência de imagens. Essas características destacam-se em relação ao fundo da imagem e a imagem é segmentada em região facial e não-facial [31]. Utilizando um robusto classificador Haar-cascade [32] para detecção de faces, na versão *haarcascade_frontalface_default*, foi possível transformar as imagens cedidas pelos participantes voluntários para o mesmo formato do banco de dados principal e assim dar continuidade aos testes e avaliações (Figura 6).

Figura 6. Exemplo de padronização das imagens do voluntários.

D. Técnicas de extração de recursos

Após a detecção da face, o próximo passo é a extração de recursos. Com o objetivo de obter representações dos componentes do rosto sem perder informações, a técnica principal da CNN é a convolução, em que é aplicado filtros (também chamados de *kernels* ou núcleos) a sub-regiões dos dados de entrada, produzindo mapas de recursos. Estes mapas são usados para identificar padrões e realizar a classificação ou regressão. Além disso, as CNNs podem usar técnicas adicionais, como normalização de camada e regularização, para melhorar o desempenho do modelo [33].

A convolução funciona aplicando uma operação matemática sobre uma janela deslizante do tensor de entrada (geralmente uma imagem) e um filtro. O resultado da operação é uma saída 2D, que é chamada de ”mapa de recursos”. Um exemplo seria o uso de 16 filtros, que significa que o modelo irá executar a operação 16 vezes, cada uma com um filtro diferente, gerando assim 16 mapas de recursos diferentes (Figura 7). Cada filtro é inicializado com valores aleatórios que se ajustam durante o treinamento do modelo. A combinação dos mapas de recursos resulta em uma representação mais complexa dos dados de entrada, que é usada pelas camadas subsequentes da rede até a saída nas classes predeterminadas. Em resumo, a convolução permite a extração de características da entrada de uma forma que é ideal para o processamento por redes neurais [34].

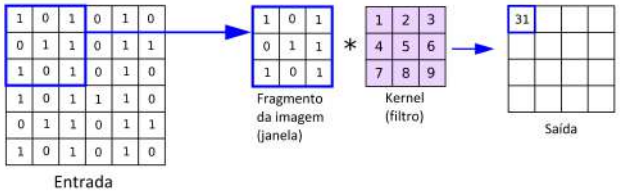

Figura 7. Exemplo da técnica de convolução [35].

IV. Sistema Proposto

A Figura 8 representa o pipeline do sistema, desde o carregamento dos dados até a saída das tabelas e gráficos. Inicialmente, o banco de dados principal foi carregado e submetido a um pré-processamento que consistiu na normalização dos dados para um formato que permitisse o melhor funcionamento da rede neural. Em seguida, aconteceu um fluxo de dados contendo as imagens através das camadas da CNN seguindo as métricas estabelecidas, até estabilizar em um pico de acurácia. Posteriormente, o sistema carregou as imagens dos voluntários e os modelos necessários para criar um novo banco de dados e realizar a predição das imagens, possibilitando a realização de comparações e análises.

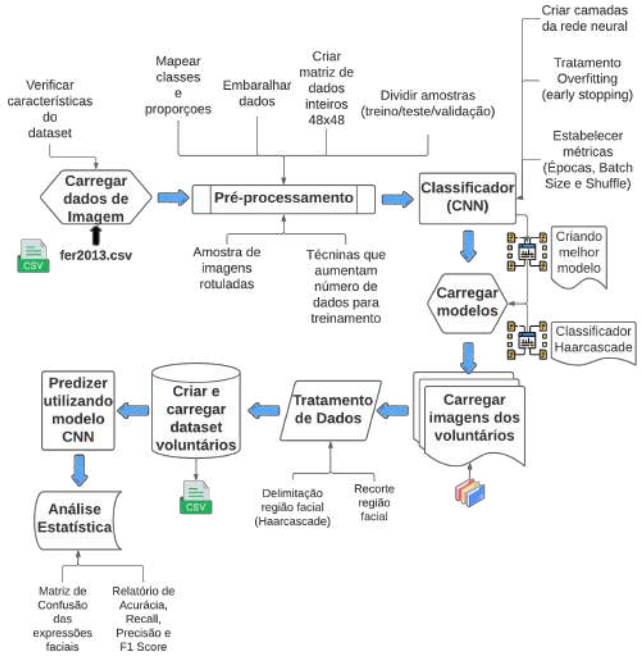

Figura 8. Pipeline do Sistema

A seguir, apresenta-se a versão da CNN utilizada em todos os testes e análises, tendo sido realizadas adaptações com base no material referencial [5]. A fim de melhorar a qualidade do fluxo da rede neural, as camadas de convolução foram gradativamente tornadas mais densas e combinadas com camadas de normalização [36]. Ademais, o uso de camadas de *dropout* foi a técnica empregada para aprimorar a generalização da rede e lidar de forma mais eficaz com imagens ruidosas.

```
def cnn_model():
    model= tf.keras.models.Sequential()
    model.add(Conv2D(32, kernel_size=(3, 3),
                     padding='same', activation='
                         relu',
                     input_shape=(48,48,1)))
    model.add(Conv2D(64, kernel_size=(3, 3),
                     padding='same', activation='
                         relu'))
    model.add(BatchNormalization())
    model.add(MaxPool2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Conv2D(128, kernel_size=(5, 5),
                     padding='same', activation='
                         relu'))
    model.add(BatchNormalization())
    model.add(MaxPool2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
```

```
    model.add(Conv2D(512, kernel_size=(3, 3),
                     padding='same', activation='relu',
                     kernel_regularizer=regularizers.l2(0.01)))
    model.add(BatchNormalization())
    model.add(MaxPool2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Conv2D(512, kernel_size=(3, 3),
                     padding='same', activation='relu',
                     kernel_regularizer=regularizers.l2(0.01)))
    model.add(BatchNormalization())
    model.add(MaxPool2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Conv2D(512, kernel_size=(3, 3),
                     padding='same', activation='relu',
                     kernel_regularizer=regularizers.l2(0.01)))
    model.add(BatchNormalization())
    model.add(MaxPool2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Flatten())
    model.add(Dense(512, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))
    model.add(Dense(512, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))
    model.add(Dense(7, activation='softmax'))
    model.compile(
    optimizer = Adam(lr=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy'])
    return model
```

Durante o treinamento da rede, é importante monitorar a precisão (accuracy) da validação e salvar os melhores pesos do modelo para evitar o *overfitting*. A função a seguir utiliza instancias para lidar com isso, sendo a primeira delas o *EarlyStopping*, que interrompe o treinamento se a precisão da validação não melhorar após 10 épocas e mantém somente os melhores pesos do modelo. A segunda função é o *ModelCheckpoint* que salva os pesos do modelo a cada época, caso a precisão da validação melhore. Ele salva apenas o melhor modelo, evitando que o modelo seja sobrescrito por modelos de menor qualidade.

```
    checkpointer = [EarlyStopping(monitor = 'val_accuracy', verbose = 1,
        restore_best_weights=True,mode="max",
        patience = 10), ModelCheckpoint('best_model.h5',monitor="val_accuracy",
        verbose=1, save_best_only=True,mode="max")]
```

Já o próximo bloco de código representa a função responsável por iniciar o treinamento da rede, definindo parâmetros de treinamento, como número de épocas, tamanho do lote utilizado para treinamento, nível de detalhamento do treinamento, uso de *callbacks*, dados de validação e se o modelo irá embaralhar os dados a cada época.

```
    history = model.fit(train_generator,
                        epochs=100,
                        batch_size=128,
                        verbose=2,
                        callbacks=[checkpointer],
                        validation_data=val_generator,
                        shuffle=True)
```

A distribuição dos dados de imagem do conjunto de dados para o treinamento é apresentada na Tabela IV. Os dados foram selecionados de forma aleatória, mas seguindo as proporções de 81% para treinamento, 9% para validação e 10% para teste.

Tabela IV
SEPARAÇÃO DAS AMOSTRAS PARA CADA ETAPA

Grupo	Amostragem	Formato
Treino Validação Teste	29068 imagens 3230 imagens 3589 imagens	48x48 pixels

V. TESTES E INFERÊNCIAS

Com base nas mesmas métricas e especificações, foram criados diversos modelos que reorganizam somente os dados de imagem utilizados. A partir deles, foram extraídos dois cenários diferentes, possibilitando a realização de comparativos e análises, conforme descrito em Tabela V e Tabela VIII. Para isso, foram utilizadas as proporções citadas na Tabela IV para validação do modelo. Além disso, as imagens cedidas pelos voluntários nos casos de teste continham sempre as mesmas 61 imagens.

Os resultados desses testes e análises podem ser extremamente úteis para aprimorar a precisão e eficiência dos modelos de CNN, possibilitando a classificação mais acurada de imagens em diversas áreas de aplicação, como saúde, segurança e reconhecimento de padrões. Com isso, pode-se esperar uma melhora significativa no desempenho de tais modelos e, consequentemente, uma maior confiabilidade nas informações obtidas a partir deles.

A. *Teste I*

Tabela V
DISTRIBUIÇÃO DE TESTES I

Index	Expressão Facial	FER2013 Treino I	FER2013 Validação I	FER2013 Teste I	Voluntários Teste I
0	Raiva	4013	468	472	6
1	Nojo	443	51	53	7
2	Medo	4148	443	530	8
3	Felicidade	7281	817	891	13
4	Neutro	5120	497	581	8
5	Tristeza	5222	412	443	9
6	Surpresa	2841	542	619	10
Total		29068	3230	3589	61

Com base no modelo criado, foi possível alcançar uma taxa de acurácia de 70,08% durante os testes (Figura 9).

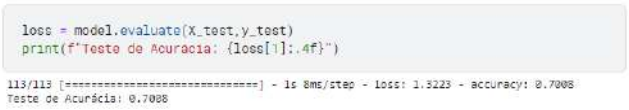

Figura 9. Teste de Acurácia do Modelo 1

A partir de uma matriz de confusão, foi possível gerar relatórios detalhados sobre o desempenho do modelo em relação aos conjuntos de dados FER2013 (Tabela VI) e voluntários (Tabela VII) durante o Teste I. Esses relatórios fornecem informações importantes sobre como o modelo se comporta em diferentes cenários de classificação de imagens [37], permitindo que os desenvolvedores façam ajustes e melhorias em seu desempenho. Com base nessas informações, é possível otimizar o modelo para obter resultados mais precisos e confiáveis em aplicações futuras.

Tabela VI
RELATÓRIO FER2013 - TESTE I

Index	Precisão	Recall	F1-Score	Amostras
0	0.583	0.697	0.635	472
1	0.763	0.547	0.637	53
2	0.700	0.334	0.452	530
3	0.860	0.908	0.883	891
4	0.575	0.597	0.586	581
5	0.864	0.745	0.800	443
6	0.611	0.798	0.692	619
Acurácia			0.701	3589
Média aritmética	0.708	0.661	0.669	3589
Média ponderada	0.710	0.701	0.692	3589

Tabela VII
RELATÓRIO VOLUNTÁRIOS - TESTE I

Index	Precisão	Recall	F1-Score	Amostras
0	0.667	0.667	0.667	6
1	1.000	0.143	0.250	7
2	0.500	0.125	0.200	8
3	1.000	0.769	0.870	13
4	0.333	0.625	0.435	8
5	0.500	0.222	0.308	9
6	0.348	0.800	0.485	10
Acurácia			0.508	61
Média aritmética	0.621	0.479	0.459	61
Média ponderada	0.634	0.508	0.488	61

B. Teste II

Tabela VIII
DISTRIBUIÇÃO DE TESTES II

Index	Expressão Facial	FER2013 Treino II	FER2013 Validação II	FER2013 Teste II	Voluntários Teste II
0	Raiva	3981	462	510	6
1	Nojo	443	46	58	7
2	Medo	4158	471	492	8
3	Felicidade	7241	810	938	13
4	Neutro	5020	575	603	8
5	Tristeza	5376	330	371	9
6	Surpresa	2849	536	617	10
Total		29068	3230	3589	61

A partir de um novo teste com dados reorganizados (Tabela VIII), o segundo modelo atingiu uma taxa de acurácia de 67,62% (Figura 10).

Figura 10. Teste de Acurácia do Modelo 2

Os relatórios derivados das matrizes de confusão da FER2013 (Tabela IX) e dos voluntários (Tabela X) do teste II apresentaram resultados mais satisfatórios, apesar da taxa de acurácia ser menor. Isso evidencia a possibilidade de realizar diversas inferências a respeito do que pode ter ocorrido, considerando que os parâmetros da rede neural permaneceram inalterados.

Tabela IX
RELATÓRIO FER2013 - TESTE II

Index	Precisão	Recall	F1-Score	Amostras
0	0.604	0.635	0.620	510
1	0.857	0.621	0.720	58
2	0.672	0.337	0.449	492
3	0.885	0.900	0.892	938
4	0.586	0.657	0.619	603
5	0.865	0.725	0.789	371
6	0.586	0.781	0.669	617
Acurácia			0.701	3589
Média aritmética	0.722	0.665	0.680	3589
Média ponderada	0.712	0.701	0.695	3589

Tabela X
RELATÓRIO VOLUNTÁRIOS - TESTE II

Index	Precisão	Recall	F1-Score	Amostras
0	0.625	0.833	0.714	6
1	1.000	0.286	0.444	7
2	0.500	0.250	0.333	8
3	1.000	0.769	0.870	13
4	1.000	0.500	0.667	8
5	0.600	0.333	0.429	9
6	0.357	1.000	0.526	10
Acurácia			0.590	61
Média aritmética	0.726	0.567	0.569	61
Média ponderada	0.733	0.590	0.587	61

C. Inferências

1) Os testes demostraram que os modelos possuíram uma tendência a classificar as expressões como “Surpresa”. Na Figura 11, recorte das matrizes de confusão, é possível perceber como a predição da expressão foi amplamente distribuída entre os valores reais. Não foi possível determinar a causa do *overfitting* para surpresa.

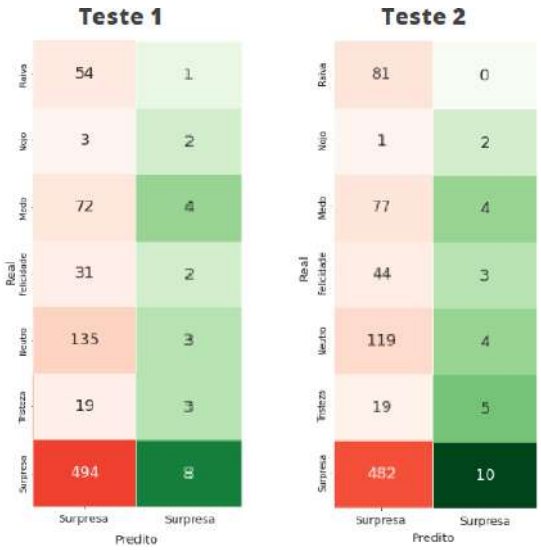

Figura 11. Inferência 1: Matriz de confusão comparativa entre FER2013 (laranja) e voluntários (verde).

2) A avaliação da acurácia durante as épocas não garante que o modelo final será o melhor ou que sua qualidade seja definitiva. Na Figura 12, é apresentada uma comparação entre dois testes, onde o primeiro obteve uma acurácia superior ao segundo. Entretanto, ao analisar os relatórios referentes aos voluntários, foi constatado que o teste 2 apresentou um desempenho superior ao teste 1, evidenciando que a acurácia não é o único fator determinante na escolha do melhor modelo.

Figura 12. Inferência 2: Comparação de acurácia entre os testes

3) A composição e qualidade dos dados de treino/validação/teste afetam diretamente a qualidade do modelo. Como os dados são embaralhados para criar modelos diferentes, é difícil determinar se os mesmos representam imagens "limpas" ou ideais, por isso é possível criar modelos melhores que outros dependendo somente da forma que os dados são distribuídos usando o bancos de dados FER2013.

Figura 13. Inferência 3: Exemplos de Expressão Predita (Expressão Real)

4) Uma amostragem de dados reduzida de uma classe não representou uma predição pior, onde o nojo obteve valores superiores ao medo independente do número de amostras (Figura 14).

		Precisão	Recall	F1-Score	Suporte
Teste 1	Nojo	0.763	0.547	0.637	53
	Raiva	0.700	0.334	0.452	530
Teste 2	Nojo	0.857	0.621	0.720	58
	Raiva	0.672	0.337	0.449	492

Figura 14. Inferência 4: Relatório Parcial FER2013.

5) Identificou-se através dos testes e revisão da literatura que a probabilidade de ocorrência de *overfitting* é diretamente proporcional a quantidade de classes e características a serem identificadas (Figura 15), reduzindo um pouco a significância da qualidade dos dados e ajustes na CNN quanto aos resultados dos modelos.

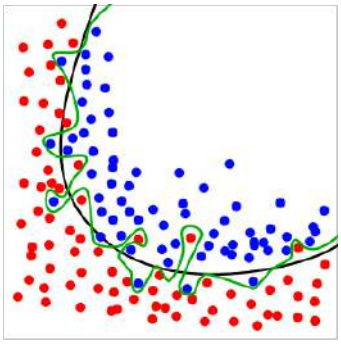

Figura 15. Inferência 5: Exemplo de *overfitting* com duas classes que partilhas de mesmas características. [37]

6) Dados com características ruidosas podem resultar em saltos de aprendizado incongruentes que afetam a qualidade do modelo. Na primeira expressão com fundo preto e o personagem usando óculos escuros (Figura 16), as

características mais importantes dificilmente são identificáveis, o que pode levar a detecção inadequada dos pontos de interesse. É crucial levar em conta esses detalhes para evitar a criação de modelos tendenciosos em relação à cor da pele. Outro caso, são imagens com marca d'água cobrindo as regiões de interesse que também podem afetar negativamente o aprendizado do modelo, já que ele pode aprender erroneamente que a marca d'água faz parte das características de interesse.

Figura 16. Inferência 6: Amostragem do FER2013.

7) Ao remodelar as imagens dos voluntários em 48x48 pixels, ocorreu um aumento de características indesejadas e ocultação de regiões de interesse em alguns casos, especialmente em imagens originalmente em alta resolução. A Figura 17 destaca em que a boca, uma região de interesse, ficou ligeiramente deformada, e os olhos receberam detalhes de brilho que não eram relevantes para a classificação.

Figura 17. Inferência 7: Exemplo de remodelagem em um dos voluntários.

VI. Conclusões e Trabalhos Futuros

Este estudo teve como objetivo implementar uma metodologia para prever emoções a partir de expressões faciais, utilizando redes neurais convolucionais. A principal tarefa foi classificar as expressões em até sete categorias de emoções, com uma expectativa de acurácia de pelo menos 60%. Além disso, o projeto teve como objetivo fornecer uma experiência prática e teórica para os sete estudantes de Engenharia da Computação que participaram como voluntários para criar um conjunto de dados e entender como suas imagens seriam utilizadas em cada etapa do processo.

Apesar das limitações técnicas, de prazo e de qualidade do conjunto de dados, este estudo demonstrou a viabilidade de utilizar redes neurais para classificar expressões faciais. No entanto, para obter resultados mais precisos, é necessário realizar um pré-processamento bem estruturado dos dados de entrada e utilizar técnicas de computação e hardware mais eficientes. Além disso, pesquisas futuras poderiam explorar abordagens utilizadas em classificadores mais atuais, bem como outras técnicas de classificação de expressões faciais, como Floresta Profunda (Deep Forest), Máquina de vetores de suporte (Support Vector Machine - SVM), Rede Neural Direta (Feed-Forward Neural Networks - FNN) e Redes Neurais Artificiais (Artificial Neural Networks - ANN).

Com relação ao preparo para trabalhos futuros, foram transformadas e extraídas 920 imagens do conjunto de dados CK+ para a mesma estrutura no formato CSV usada no FER2013 [29], que está publicado na plataforma Kaggle [28]. Isso pode ajudar outros pesquisadores a trabalhar com esses dados de maneira mais eficiente e padronizada.

Referências

[1] P. Ekman and W. Friesen, *Unmasking the Face: A Guide to Recognizing Emotions from Facial Clues*. No. v. 10 in Spectrum book, Malor Books, 2003.
[2] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.
[3] K. Vemou, T. Zerdick, A. Horvath, and E. D. P. Supervisor, *EDPS TechDispatch: Facial Emotion Recognition. Issue 1, 2021*. EDPS TechDispatch, Publications Office of the European Union, 2021.
[4] A. Balaji, "Emotion detection using deep learning." https://github.com/atulapra/Emotion-detection, 2020.
[5] Öykü Eravcı, *Emotion Detection using CNN*. Kaggle, 2018.
[6] P. Ekman, "Universal facial expressions of emotion," Master's thesis, University of San Francisco, Califórnia, mar. 1970.
[7] P. Ekman, *Universal Emotions*. Paul Ekman Group, 2022.
[8] R. Szeliski, *Computer Vision: Algorithms and Applications*. Berlin, Heidelberg: Springer-Verlag, 1st ed., 2010.
[9] C. Tian, J. Ma, C. Zhang, and P. Zhan, "A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network," *Energies*, vol. 11, no. 12, 2018.
[10] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.
[11] D. Zhang, L. Tian, M. Hong, F. Han, Y. Ren, and Y. Chen, "Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification," *IEEE Access*, vol. 6, pp. 73750–73759, 2018.
[12] A. M. Tudose, D. O. Sidea, I. I. Picioroaga, V. A. Boicea, and C. Bulac, "A cnn based model for short-term load forecasting: A real case study on the romanian power system," in *2020 55th International Universities Power Engineering Conference (UPEC)*, pp. 1–6, 2020.
[13] D. Mariano, C. Leite, L. Santos, R. Rocha, and R. Melo-Minardi, "A guide to performing systematic literature reviews in bioinformatics," 07 2017.
[14] P. A. Riyantoko, Sugiarto, and K. M. Hindrayani, "Facial emotion detection using haar-cascade classifier and convolutional neural networks," *Journal of Physics: Conference Series*, vol. 1844, p. 012004, mar 2021.
[15] R. S. Deshmukh, V. Jagtap, and S. Paygude, "Facial emotion recognition system through machine learning approach," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 272–277, 2017.

[16] A. Fathallah, L. Abdi, and A. Douik, "Facial expression recognition via deep learning," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 745–750, 2017.
[17] M. Moolchandani, S. Dwivedi, S. Nigam, and K. Gupta, "A survey on: Facial emotion recognition and classification," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1677–1686, 2021.
[18] E. Kodhai, A. Pooveswari, P. Sharmila, and N. Ramiya, "Literature review on emotion recognition system," in *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pp. 1–4, 2020.
[19] E. Sőnmez and A. Cangelosi, "Convolutional neural networks with balanced batches for facial expressions recognition," 11 2016.
[20] S. Gupta, "Facial emotion recognition in real-time and static images," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 553–560, 2018.
[21] S. Das, M. K. Synyal, S. K. Upadhyay, and S. Chatterjee, "An intelligent approach for predicting emotion using convolution neural network," *Journal of Physics: Conference Series*, vol. 1797, p. 012014, feb 2021.
[22] S.-Y. Lin, Y.-W. Tseng, C.-R. Wu, Y.-C. Kung, Y.-Z. Chen, and C.-M. Wu, "A continuous facial expression recognition model based on deep learning method," in *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 1–2, 2019.
[23] A. Kandeel, M. Rahmanian, F. Zulkernine, H. M. Abbas, and H. Hassanein, "Facial expression recognition using a simplified convolutional neural network model," in *2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pp. 1–6, 2021.
[24] R. Jaiswal, "Facial expression classification using convolutional neural networking and its applications," in *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pp. 437–442, 2020.
[25] D. Sena, *Davi Sena - FER using CNN applied to Byte Jr.* Kaggle, 2023.
[26] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, *Challenges in Representation Learning: A report on three machine learning contests*. Zenodo, 2013.
[27] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," pp. 94 – 101, 07 2010.
[28] D. Sena, *CK+ Dataset*. Kaggle, 2023.
[29] R. Verma, O. Eravci, and P. V. Linh, *fer2013*. Kaggle, 2018.
[30] J. Shenk, *FER - Facial Expression Recognition*. Paul Ekman Group, 2022.
[31] S. Rajan, P. Chenniappan, S. Devaraj, and N. Madian, *Facial expression recognition techniques: a comprehensive survey*. The Institution of Engineering and Technology, 2019.
[32] Y.-L. Tian, T. Kanade, and J. F. Cohn, *Facial Expression Analysis*, pp. 247–248. Springer, Sept. 2005.
[33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.
[35] A. H. Reynolds, *Convolutional Neural Networks (CNNs)*. ANH H. REYNOLDS, 2019.
[36] M. Bamelis, "How to improve the performance of cnn model for a specific dataset? getting low accuracy on both training and testing dataset?." Cross Validated, 2022. URL:https://stackoverflow.com/a/70579136 (version: 2022-01-04).
[37] G. Schade, *Machine Learning: métricas para Modelos de ClassificaçãoNN*. IMasters, 2019.