

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Eric Mendes Pereira

**Investigação de uma abordagem da arquitetura
CapsNet para detecção de Microexpressões**

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Eric Mendes Pereira

**Investigação de uma abordagem da arquitetura CapsNet
para detecção de Microexpressões**

Trabalho de conclusão de curso apresentado
à Faculdade de Engenharia Mecânica da Uni-
versidade Federal de Uberlândia, Minas Ge-
rais, como requisito exigido parcial à obten-
ção do grau de Bacharel em Engenharia Me-
catrônica.

Orientador: Marcelo Zanchetta do Nascimento

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Mecânica

Bacharelado em Engenharia Mecatrônica

Uberlândia, Brasil

2025

Eric Mendes Pereira

Investigação de uma abordagem da arquitetura CapsNet para detecção de Microexpressões

Trabalho de conclusão de curso apresentado
à Faculdade de Engenharia Mecânica da Uni-
versidade Federal de Uberlândia, Minas Ge-
rais, como requisito exigido parcial à obten-
ção do grau de Bacharel em Engenharia Me-
catrônica.

Marcelo Zanchetta do Nascimento
Orientador

Prof. Dr. Paulo Henrique R. Gabriel

Prof. Dr. Thiago Pirola Ribeiro

Uberlândia, Brasil
2025

Resumo

As emoções básicas são fundamentais na comunicação não-verbal humana e através delas é possível identificar os estados de alegria, medo, surpresa, tristeza, nojo e raiva. Ao contrário das expressões faciais básicas, que podem ser observadas e distinguidas com certa facilidade por pessoas e computadores, as microexpressões (ME) são difíceis de serem reconhecidas e carecem de outros mecanismos para detectá-las. O trabalho investiga a aplicação da arquitetura CapsNet para reconhecimento de microexpressões — expressões faciais breves, involuntárias que revelam uma tentativa de ocultar a emoção. A metodologia apresentada utiliza o banco de dados CASME II, que contém vídeos de microexpressões espontâneas, e combina a CapsNet com uma ResNet pré-treinada como *backbone*. Técnicas de aumento de dados, como recorte aleatório, espelhamento e variação de cores, foram aplicadas para aprimorar a capacidade de generalização do modelo. Os resultados indicam que a combinação da CapsNet com a ResNet50 e técnicas de aumento de dados é razoavelmente eficaz, em comparação a outros métodos da literatura, na classificação de microexpressões (arquitetura ResNet50 com UAR = 0,7493 e UF1 = 0,7727). No entanto, o alto custo computacional representa um desafio para sua aplicação em tempo real.

Palavras-chave: Microexpressões, Aprendizado de máquina, CapsNet, Aumento de dados, Reconhecimento de Microexpressões Faciais, Redes Neurais Convolucionais.

Lista de ilustrações

Figura 1 – Exemplo de UAs (AUs), representadas no rosto humano. Fonte: (EK-MAN; FRIESEN; HAGER, 2002)	14
Figura 2 – Representação do neurônio artificial <i>Perceptron</i> . Fonte: (DANTAS, 2022)	17
Figura 3 – Representação do <i>Perceptron</i> de várias camadas. Fonte: (DANTAS, 2022)	18
Figura 4 – Demonstração de uma convolução em uma matriz 5x5 pixels com filtro de dimensão 3x3, resultando em um mapa de características 3x3. Adaptado de (EDUCATION, 2020)	18
Figura 5 – Demonstração do <i>MaxPooling</i> em uma matriz 4 x 4 pixels, resultando em um segundo mapa de recursos 2x2. Adaptado de (KHOSLA, 2021)	19
Figura 6 – Demonstração do <i>AveragePooling</i> em uma matriz 4x4 pixels, resultando em um segundo mapa de recursos 2x2. Adaptado de (KHOSLA, 2021)	20
Figura 7 – A arquitetura básica da CapsNet com as três camadas citadas, onde o termo W_{ij} representa uma matriz de pesos que transforma cada cápsula de PrimaryCaps para outra em DigitCaps. Fonte: (SABOUR; FROSST; HINTON, 2017)	21
Figura 8 – Fluxograma com as principais etapas para reconhecimento das microexpressões.	26
Figura 9 – Arranjo do aparato para aquisição dos cliques de microexpressões. Adaptado de (YAN et al., 2014)	27
Figura 10 – Exemplo da sequência de frames em um clipe do banco de imagens CASME II. O frame em 110 ms representa o ápice de intensidade da microexpressão, indicando desgosto através das unidades de ação do FACS. Fonte: (YAN et al., 2014)	28
Figura 11 – Camadas da rede ResNet18. Fonte: (RAMZAN et al., 2019)	29
Figura 12 – Estrutura completa da arquitetura final de CapsNet. Adaptado de (SABOUR; FROSST; HINTON, 2017). Camadas ResNet se referem às camadas convolucionais da arquitetura <i>ResNet</i> , a camada <i>PrimaryCaps</i> diz respeito às cápsulas primárias que detectam características específicas e respectivas orientações. Por fim, a camada <i>DigitCaps</i> (cápsulas dígitos) é a camada responsável pela classificação. O termo $\ L_2\ $, dito norma do vetor, significa o comprimento desse vetor e pode ser interpretado como a probabilidade da presença de uma determinada classe.	30

Figura 13 – Essa técnica de <i>DA</i> consiste em rotacionar a imagem original (a) em um ângulo randômico, resultando na imagem (b). Fonte: (TRANSFORMS, 2020)	31
Figura 14 – Essa técnica de <i>DA</i> consiste em espelhar horizontalmente a imagem (a), resultando na imagem (b). Fonte: (TRANSFORMS, 2020)	32
Figura 15 – Essa técnica de <i>DA</i> consiste em alterar brilho, contraste, saturação ou matiz da imagem (a), resultando na imagem (b). Fonte: (TRANSFORMS, 2020)	32
Figura 16 – Essa técnica de <i>DA</i> aleatoriamente escolhe uma posição na imagem (a), e a recorta, resultando na imagem (b). Fonte: (TRANSFORMS, 2020)	33
Figura 17 – Essa técnica de <i>DA</i> aplica, com uma certa chance, na imagem (a), a transformação para escala de cinza, resultando na imagem (b). Fonte: (TRANSFORMS, 2020)	33
Figura 18 – Essa técnica de <i>DA</i> equaliza aleatoriamente o histograma dos pixels da imagem (a), resultando na imagem (b). Fonte: (TRANSFORMS, 2020)	34
Figura 19 – Essa técnica de <i>DA</i> ajusta aleatoriamente o contraste da imagem (a), e como resultado obtém-se a imagem (b). Fonte: (TRANSFORMS, 2020)	35
Figura 20 – Métrica UAR a cada iteração do método <i>LOSO</i> no experimento DA-3 ResNet50. Fonte: De autoria própria.	38
Figura 21 – Matriz de confusão obtida após classificação das microexpressões pela rede utilizando a arquitetura DA-3 ResNet50. Fonte: De autoria própria.	39

Lista de tabelas

Tabela 1	– Principais unidades de ação do FACS.(EKMAN; AND, 1969)	14
Tabela 2	– Distribuição das três classes de microexpressões.	28
Tabela 3	– Resultados obtidos utilizando os protocolos de aumento de dados propostos nas arquiteturas de <i>backbone ResNet18</i> e <i>Resnet50</i> .	38
Tabela 4	– Comparação dos resultados com os obtidos por outros trabalhos	39

Lista de abreviaturas e siglas

AM	Aprendizado de Máquina
AP	Average Pooling
AdaBoost	Adaptive Boosting
ASM	Active Shape Model
AU	Action Units
CapsNet	Capsule Network
CASME II	Chinese Academy of Sciences Micro-expression II
CNN	Convolutional Neural Network
DA	Data Augmentation
DR	Dynamic Routing
DRMF	Discriminative Response Map Fitting
EVM	Eulerian Video Magnification
FC	Fully Connected
FACS	Facial Action Coding System
HOG	Histograms of Oriented Gradients
IA	Inteligência Artificial
KNN	K-Nearest Neighbours
LBP-TOP	Local Binary Patterns from Three Orthogonal Planes
LOSO	Leave-One-Subject-Out
ME	MicroExpressões
MLP	Multi Layer Perceptron
MNIST	Modified National Institute of Standards and Technology
MP	Max Pooling

RF	Random Forest
RGB	Red, Green, Blue
ReLU	Rectified Linear Unit
ResNet	Residual Network
SVM	Support Vector Machine
TIM	Temporal Interpolation Model
UAR	Unweighted Average Recall
UF1	Unweighted F1-score

Sumário

1	INTRODUÇÃO	10
1.1	Objetivos	11
1.1.1	Geral	11
1.1.2	Específicos	11
1.2	Organização deste Trabalho de Conclusão de Curso	12
2	CONTEXTUALIZAÇÃO	13
2.1	Fundamentação Teórica	13
2.1.1	Emoção	13
2.1.2	Microexpressões	15
2.1.3	Reconhecimento das expressões e microexpressões	15
2.1.4	Inteligência Artificial	16
2.1.5	Redes Neurais Artificiais	17
2.1.6	CapsuleNet	20
2.2	Método de Avaliação	23
2.3	Estado da Arte	23
2.4	Considerações Finais	25
3	METODOLOGIA	26
3.1	Introdução	26
3.2	Banco de Imagens	26
3.3	Arquitetura CapsNet	28
3.4	Aumento de Dados	30
3.5	Configuração do Experimento	35
4	RESULTADOS	37
4.1	Avaliação dos Modelos e Aumento de Dados	37
4.2	Resultados	37
4.3	Análise em relação ao estado da arte	39
5	CONCLUSÃO	41
5.1	Desafios Encontrados	41
5.2	Trabalhos Futuros	41
	REFERÊNCIAS	43

1 Introdução

No estudo da expressão das emoções é notado que ela pode ser manifestada de várias formas. Por exemplo, na comunicação vocal, as variações na altura e na frequência da voz dizem respeito a diferentes estados emocionais. Além disso, a expressão corporal compreende a gesticulação e o movimento facial, que são expressões da emoção resultantes dos movimentos dos músculos da face de acordo com o estímulo. As emoções básicas são fundamentais na comunicação não-verbal humana e através delas é possível identificar os estados de alegria, medo, surpresa, tristeza, nojo e raiva.

Ao contrário das expressões faciais básicas, que podem ser observadas e distinguidas com certa facilidade por pessoas e computadores, as MicroExpressões (ME) são dificilmente reconhecidas a olho nu e, por isso, carecem de outros mecanismos para detectá-las. Segundo os autores no estudo ([EKMAN; AND, 1969](#)), as microexpressões são observadas através de uma expressão facial genuína, breve e involuntária, em indivíduos que estão tentando esconder uma emoção básica. Apesar do formato das microexpressões ser o mesmo que as expressões faciais básicas, a intensidade do movimento facial é menor e o tempo de duração mais curto, sendo adequado definí-la pela sua duração total ser inferior a 0,5 segundos ([YAN et al., 2013](#)).

O fato de que a taxa de reconhecimento de microexpressões por indivíduos treinados é baixa, torna o desenvolvimento de técnicas para detecção automática de grande interesse. Atualmente, ainda são encontradas dificuldades no processo de reconhecimento automático principalmente pelo baixo número de bancos de dados contendo amostras consideráveis de microexpressões faciais espontâneas ([BARBIERI, 2018](#)). Algumas das possíveis aplicações do reconhecimento automático de microexpressões consistem em interrogatório policial, psicoterapia, diagnóstico clínico e negociação ([YAN et al., 2014](#)).

As etapas para reconhecimento automático de microexpressões consistem em primeiramente no pré-processamento da imagem ou sequência de imagem, onde a resolução será padronizada e possíveis ruídos corrigidos. A próxima etapa, é de detecção de face e de suas regiões de interesse que como olhos, testa, sobrancelhas, boca e nariz. Tendo as regiões de interesse isoladas, o próximo passo é extrair informações delas através dos descritores e posteriormente classificá-las ([KUMARI; RAJESH; POOJA, 2015](#)).

Para o reconhecimento de microexpressões assistido por computador, é importante utilizar os descritores de imagens que extraem características importantes como cor, forma, movimento e textura. Através disso, as técnicas de classificação utilizando algoritmos de aprendizado de máquina se tornam mais acurados. Alguns dos descritores utilizados nos trabalhos de reconhecimento de faces são histograma de gradiente orientado

(do inglês, *histogram of oriented gradients* - HOG) (DALAL; TRIGGS, 2005), histograma de fluxo óptico (do inglês, *histogram of optical flow gradients* - HOF) (SZELISKI, 2010), espaço temporal local (PFISTER et al., 2011), entre outros. As técnicas de classificação são expressamente exploradas e consistem desde algoritmos de aprendizado de máquina mais simples como máquinas de vetor de suporte (do inglês, *support vector machine* - SVM) (BARBIERI, 2018), até algoritmos de aprendizagem profunda utilizando redes neurais artificiais (MIAO et al., 2019). Um modelo importante para essa etapa é as redes *CapsuleNetworks* (*CapsNet*) que são capazes de buscar informações espaciais e características mais importantes para superar a perda de informações, que é vista em operações de agrupamento (do inglês, *pooling*) em modelos tradicionais. Apesar das relevantes contribuições já presentes na literatura ainda há desafios nessa área devido a complexidade de detecção das regiões com a presença das *ME*.

Este estudo explora investigar um modelo *CapsNet* no reconhecimento de micro-expressões, que é um aprimoramento das redes neurais convolucionais (do inglês, *convolutional neural network* - CNN), extensivamente utilizadas quando há a necessidade de extrair características de imagens bem como classificá-las. As CNNs são compostas por camadas de neurônios, porém uma particularidade delas são que parte dessas camadas é destinada a extração de *features* das imagens de entrada, que passam essas informações em forma de matrizes para as camadas de neurônios totalmente conectadas (do inglês, *fully connected* - FC) que se encarregaram da classificação. A particularidade trazida pela *CapsNet* é que o problema de perda de informações que acontece nas CNNs devido à operação de *pooling* é resolvida através das cápsulas e suas saídas em forma de vetores contendo informações espaciais e recursos importantes das imagens.

1.1 Objetivos

1.1.1 Geral

Investigar o desempenho da arquitetura *CapsNet* no reconhecimento das microexpressões faciais, entendendo sua capacidade de capturar relações espaciais entre elementos das imagens e superar limitações presentes em outras abordagens, como as CNNs.

1.1.2 Específicos

- Desenvolver uma implementação da arquitetura *CapsNet* para o reconhecimento de microexpressões faciais.
- Comparar o desempenho da *CapsNet* com diferentes arquiteturas como *backbone*, *ResNet18* e *ResNet50*, no reconhecimento de microexpressões.

- Investigar como o aumento de dados impacta na capacidade do modelo em classificar das MEs.
- Identificar desafios e limitações no uso da CapsNet, bem como comparar com outras abordagens.
- Propor melhorias para trabalhos futuros.

1.2 Organização deste Trabalho de Conclusão de Curso

Neste capítulo, foram apresentadas as considerações iniciais, motivação e contribuições do desenvolvimento deste projeto. O restante deste trabalho possui a seguinte organização:

- Capítulo 2: Introduz os principais conceitos para a compreensão e validação do trabalho proposto, bem como trabalhos relacionados a ele;
- Capítulo 3: Descreve o trabalho proposto, bem como as tecnologias, técnicas e metodologias desenvolvidas;
- Capítulo 4: Apresenta-se os resultados do trabalho proposto e comparações com o estado da arte;
- Capítulo 5: São feitas conclusões acerca do trabalho proposto, bem como considerações finais e trabalhos futuros.

2 Contextualização

Neste capítulo são abordados conceitos importantes para o entendimento do trabalho proposto. Temas das áreas de Psicologia e Computação serão apresentados de maneira introdutória como forma de contextualizar o trabalho.

2.1 Fundamentação Teórica

2.1.1 Emoção

Em linhas gerais, a emoção é uma característica importante no relacionamento interpessoal e da experiência humana. Em (GROSS, 2015), as emoções compreendem certos padrões complexos de mudanças fisiológicas, expressões faciais, comportamentos e experiências subjetivas cujos gatilhos podem ser um estímulo ou evento emocionalmente significativo. O autor ainda reitera que a duração das emoções são de segundos ou minutos, e estão intrinsecamente ligadas às avaliações subjetivas de eventos relacionados aos objetivos e necessidades pessoais. Na mesma linha de raciocínio, o autor em (LEDOUX, 1998) define as emoções como uma condição complexa da mente que surge como resposta involuntária e rápida a um estímulo relevante para o organismo. Dessa forma, uma série de modificações acontecem no funcionamento do organismo. Essas mudanças consistem em disparo do sistema nervoso, alterações neuroquímicas, hormonais e da experiência subjetiva.

Em um estudo proposto por (MIGUEL, 2015), existem três principais funções das emoções na complexidade do desenvolvimento humano como espécie. A primeira é de sinalização para agir. O autor ainda ressalta que as emoções preparam o corpo para uma ação instantânea. Logicamente que cada emoção está relacionada a uma forma específica de resposta, sendo ela comportamental ou cognitiva, que programam o indivíduo para responder de maneira adaptativa a situações específicas. A segunda função principal das emoções, segundo o autor, é na comunicação entre os indivíduos. Isso compreende as expressões faciais, vocalizações e linguagem corporal que são os meios pelos quais as emoções são externalizadas e compartilhadas. Essa comunicação emocional possibilita a compreensão do estado emocional entre indivíduos. Por fim, as emoções têm um papel importante também na forma como o indivíduo se relaciona com o ambiente que está inserido. Isto é, elas auxiliam na interpretação e avaliação dos eventos quanto a sua relevância emocional. Por consequência disso, as emoções são fundamentais em qualquer tomada de decisões do indivíduo, guiando o seu comportamento em direção às metas, vontades e necessidades.

Para análise das expressões faciais das emoções, Ekman e Friesen definiram o

Sistema de Codificação de Ação Facial (do inglês, *facial action coding system* - *FACS*), (EKMAN; AND, 1969), visto na Figura 1 que mapeia a ativação muscular da face humana por meio das unidades de ação (do inglês, *action units* - *AUs*) facilitando dessa forma relacioná-las AUs com expressões faciais. Contudo, em outro estudo Ekman e Keltner (EKMAN; AND, 1969) afirmam que nem todos AUs representam uma expressão facial. Dessa forma, é necessário entender o contexto do movimento facial antes de associá-lo a uma expressão. Praticamente, todas abordagens no reconhecimento de expressões faciais assistido por computador utilizam o FACS para categorizar as amostras de cada exemplo das emoções. Isso porque ele fornece uma linguagem objetiva e abrangente para descrever as expressões faciais. A Tabela 1 apresenta as principais características das AUs para representação das emoções.

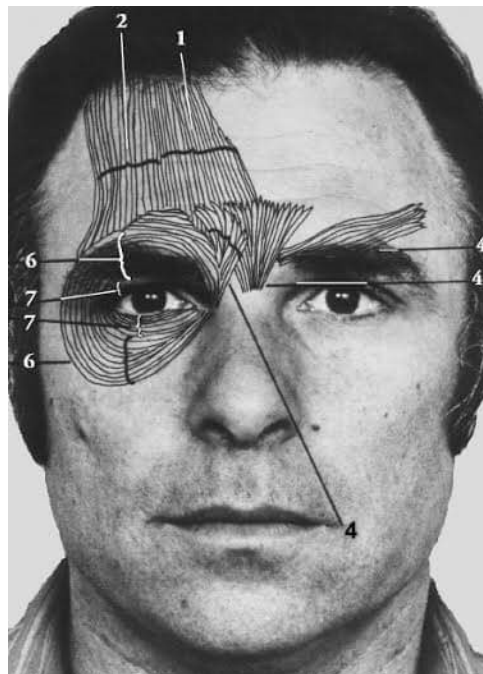


Figura 1 – Exemplo de UAs (AUs), representadas no rosto humano. Fonte: (EKMAN; FRIESEN; HAGER, 2002)

Tabela 1 – Principais unidades de ação do FACS.(EKMAN; AND, 1969)

AU	Músculo	Ação Principal
2	Frontal (parte externa)	Levantamento da porção externa da sobrancelha
4	Corrugador do Supercílio e Próceros	Tração da sobrancelha para baixo e medialmente
5	Levantador da pálpebra	Levantamento da pálpebra superior
7	Orbicular do olho	Tensionar as pálpebras
9	Levantador do lábio superior	Fransar o nariz
10	Levantador do lábio superior	Elevação do lábio superior
12	Zigomático maior	Puxa o canto dos lábios para cima
14	Bucinator	Compressão da bochecha contra os dentes molares
15	Depressor do ângulo da boca	Depressão do ângulo da boca
17	Mento	Elevação do queixo
27	Pterigoide e Digástrico	Abertura da boca

Os números na Figura 1 representam diferentes AUs conforme o FACS. Nele, cada AU se refere a um movimento específico produzido por um músculo individual ou em conjunto para gerar uma expressão facial conhecida. Com detalhes está representada a anatomia muscular da face que mostra a orientação das fibras e consequentemente a direção que são contraídas, produzindo movimentos faciais.

2.1.2 Microexpressões

As MicroExpressões (MEs) fazem parte da expressão da emoção, mais especificamente, as expressões faciais. No entanto, elas têm uma peculiaridade, pois são sutis, involuntárias e ocorrem em frações de segundos. Segundo os autores em (EKMAN; AND, 1969), elas podem acontecer quando há uma tentativa de ocultar ou reprimir uma emoção. Deste modo, as MEs têm curta duração e magnitude quando comparada com a expressão facial. Devido a essas características, a dificuldade de reconhecê-las é maior e, por esse motivo, as abordagens deste problema necessitam um maior aprofundamento.

As MEs são relevantes porque podem fornecer informações mais profundas sobre o estado emocional do indivíduo, ainda que ele esteja tentando ocultá-lo. Por esse motivo, a detecção das MEs tem sido amplamente estudada para possível aplicação em áreas como psicologia (RUSSELL; CHU; PHILLIPS, 2006) e detecção de mentiras (BARATHI, 2016).

2.1.3 Reconhecimento das expressões e microexpressões

Durante um processo de análise e reconhecimento, uma sequência de passos são aplicados aos dados com objetivo da realização da tarefa. Entre essas etapas, há o pré-processamento que consiste em extrair a sequência de *frames* dos cliques e, a partir de cada um, recortar o rosto das imagens para que seja possível analisá-las sem interferência do plano de fundo. Uma maneira de fazer esse recorte é através do modelo de forma ativa (do inglês, *active shape model* - ASM) (COOTES et al., 1995), que detecta pontos de referência em torno da face e através deles o contorno é extraído. Posteriormente, é interessante fazer um registro de face para que qualquer movimento de cabeça não afete o resultado de reconhecimento de microexpressões. Deste modo, esse processo é feito alinhando cada rosto detectado a um rosto de referência, usando pontos-chave faciais. Para esse passo pode ser utilizada a técnica ajuste de mapas de resposta discriminativos (do inglês, *discriminative response map fitting* - DRMF) (ASTHANA et al., 2013).

Como forma de evidenciar a magnitude de uma ME, que é essencialmente sutil, a ampliação de movimento pode ser utilizada. O método ampliação de vídeo Euleriana (do inglês, *eulerian video magnification* - EVM) (WU et al., 2012), que é frequente utilizado com esse objetivo, recebe uma sequência de vídeo como entrada realizando uma decomposição espacial e posterior uma filtragem temporal nos *frames*. O sinal resultante é então

amplificado para deixar as informações sutis mais aparentes.

Outro problema, além da baixa magnitude da *ME*, é a rapidez com que ela acontece. Essa baixa duração da expressão implica em um problema de poucos dados para poder alimentar o algoritmo de classificação. Como forma de resolvê-lo, é interessante utilizar de uma interpolação temporal como forma minimizar os problemas de escassez de dados bem como padronizar o comprimento da amostra. O modelo de interpolação temporal (do inglês, *temporal interpolation model* - *TIM*) é com frequência utilizado com esse objetivo nesta etapa.

O próximo passo é a escolha do tipo de entrada do modelo, podendo optar por uma sequência de imagens ou vídeos. A vantagem de utilizar imagem é principalmente a menor complexidade computacional em comparação a processar o clipe inteiro. Além disso, a abordagem utilizando apenas o *apexframe*, isto é, o *frame* em que a representação de uma ME está mais intensa, poderia obter um resultado semelhante em comparação a usar a sequência de *frame* (LI; HUANG; ZHAO, 2018). A vantagem do vídeo é que as informações espaço-temporal não são perdidas, porém a redundância de informações torna esse tipo de entrada mais custosa computacionalmente. Neste trabalho é utilizado apenas o *apexframe* como entrada do modelo.

O passo subsequente consiste em utilizar algum método para extração de características da sequência de imagens. Ele pode ser baseado no espectro espaço-temporal, como por exemplo, LBP-TOP (do inglês, *local binary patterns from three orthogonal planes* - *LBP-TOP*) (LI et al., 2018), que não é necessário um treinamento ou baseado em algoritmos de aprendizado de máquina. Por fim, é a etapa de classificação, onde é treinado e testado um modelo baseado em algoritmos de aprendizado de máquina ou redes neurais para entender a relação das características extraídas com as respectivas expressões.

2.1.4 Inteligência Artificial

Em um dos primeiros trabalhos conhecido sobre o tema, o paradigma da possibilidade de uma máquina ser inteligente foi discutido por Alan Turing (TURING, 1950). O autor discorria sobre a hipótese de que um ser humano poderia interrogar um computador e se ele poderia distinguir as questões e trazer respostas. Assim, a máquina poderia se comportar como o ser humano em que usa as informações para resolver problemas e tomar decisões. Em seu trabalho denominado "Computing Machinery and Intelligence", publicado em (TURING, 1950), ele discutiu como construir computadores inteligentes e como avaliar essa inteligência. Na Literatura, Turing é considerado como um dos principais autores a explorarem pela primeira vez o tema de Inteligência Artificial (*IA*).

Em 1956, na Conferência de Dartmouth, o termo *IA* foi formalizado e durante 2 meses assuntos como computadores automáticos, computador programado para utilizar

uma linguagem, redes de neurônios, teoria do tamanho de um cálculo, auto aperfeiçoamento, abstrações e aleatoriedade foram debatidos (MOOR, 2006). Dessa forma, pode-se considerar que as seguintes formalizações compõem grande parte da *IA*, são elas: aprendizado de máquina, processamento da linguagem natural, representação do conhecimento e raciocínio automatizado (RUSSELL; NORVIG, 2010).

O Aprendizado de Máquina (*AM*), por ser um subcampo da *IA* que necessita de exemplos para aprender, é uma excelente abordagem quando há uma imensa quantidade de dados para construir conhecimento e é desejável generalizá-lo para prever novas circunstâncias. No caso das microexpressões, técnicas tradicionais de *AM* já foram utilizadas para entender e detectar padrões nelas. Contudo, o desempenho limitado dessas abordagens justifica a exploração de modelos mais robustos, como as redes neurais (ADEGUN; VADAPALLI, 2020), que surgem com a tentativa de imitar o processo de aprendizagem que ocorre no cérebro, discutidas a seguir.

2.1.5 Redes Neurais Artificiais

Em um dos primeiros trabalhos que tentaram representar o sistema de aprendizagem humano, através das adaptações dos neurônios no processo de aprendizado, foi apresentado por Rosenblatt o *Perceptron* (ROSENBLATT, 1958) (Figura 2), que consistia em um neurônio artificial que conseguia aprender através dos dados, de forma supervisionada. Porém, uma limitação importante do *perceptron* é que ele conseguia apenas classificar atributos linearmente separáveis, fato que limitava sua utilização prática. Na Figura 2 são demonstradas as entradas do neurônio artificial j que são representadas por y_n , os pesos atrelados a cada uma delas é w_n e v é a saída do neurônio.

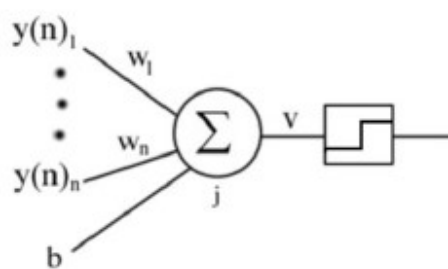


Figura 2 – Representação do neurônio artificial *Perceptron*. Fonte: (DANTAS, 2022)

Para resolver os problemas não linearmente separáveis, foi proposto criar camadas de não apenas um neurônio artificial, mas vários deles. Desta forma, se tem uma rede de neurônios conhecida como *Perceptron* de várias camadas (do inglês, *multi layer perceptron* - *MLP*) (ver Figura 3), que é uma rede neural artificial. Por definição, ela tem uma camada de entrada, que recebe os dados para processamento, as camadas ocultas, as quais realizam os cálculos dos pesos, e a camada de saída onde ocorre a classificação efetivamente. Outras

definições importantes são os pesos que indicam a influência daquele neurônio e a função de ativação, a qual define a saída baseada nas entradas através da função escolhida.

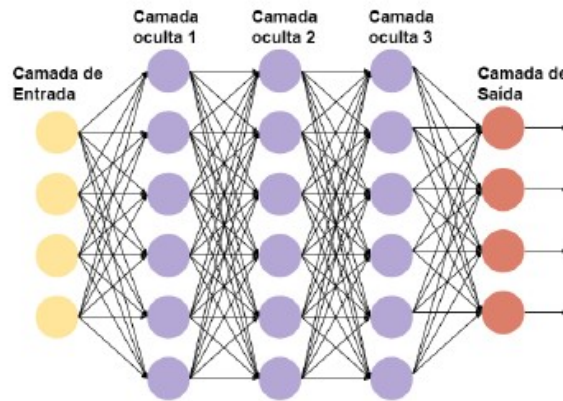


Figura 3 – Representação do *Perceptron* de várias camadas. Fonte: (DANTAS, 2022)

Quando o problema de classificação envolve imagens, as redes neurais convolucionais (do inglês, *Convolutional Neural Networks - CNN*) podem tratar as imagens como um todo, sendo possível explorar as propriedades das intensidades de brilho dos níveis de cinza. Diferente da MLP, a CNN possui três tipos de camadas principais, sendo elas: camada convolucional, camada de agrupamento e a camada totalmente conectada.

A camada convolucional é a parte principal de uma CNN, em que nela ocorre a extração de características simples e complexas das imagens, a depender do tamanho dessa camada. O funcionamento dela envolve a utilização de um filtro ou *kernel* e um mapa de características. Considerando que os dados de entrada é uma imagem, isto é, uma matriz de bidimensional, a camada convolucional irá percorrer essa matriz fazendo uma multiplicação escalar entre ela e o filtro, por meio da operação de convolução. A matriz resultante dessa operação é o mapa de características. Caso a imagem seja composta pelos 3 canais do modelo RGB, vermelho, verde e azul (RGB), a convolução é feita em todos canais, tendo um *kernel* para cada um deles. Um exemplo pode ser visto na Figura 4.

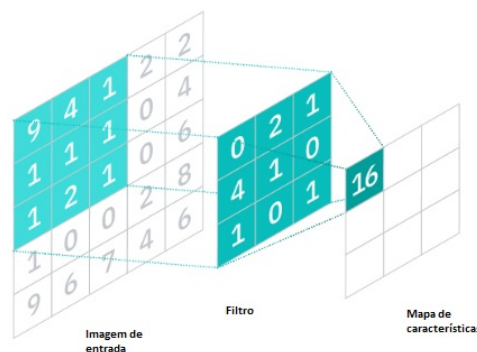


Figura 4 – Demonstração de uma convolução em uma matriz 5x5 pixels com filtro de dimensão 3x3, resultando em um mapa de características 3x3. Adaptado de (EDUCATION, 2020)

Após cada processo de convolução da imagem, uma função de ativação, preferencialmente a unidade linear retificada (do inglês, *Rectified Linear Unit*, *ReLU*) (NAIR; HINTON, 2010), é aplicada ao mapa de características com objetivo de introduzir uma não-linearidade ao modelo e aumentar sua eficiência e desempenho. Tendo o mapa de característica sido obtido e transformado, a camada de agrupamento (do inglês, *Pooling Layer*) realizam a redução de dimensionalidade da entrada, assim diminuindo o número de parâmetros nela. De certa forma, ela se parece com a camada convolucional, varrendo a entrada com um filtro, porém esse *kernel* não possui pesos. Em vez disso, o *kernel* aplica uma função de agregação que preenche a matriz de saída de acordo com os principais métodos de agrupamento.

O método agrupamento máximo (do inglês, *max pooling* - *MP*) consiste em selecionar o elemento de maior valor da região coberta pelo filtro inserido no mapa de características. Assim, a saída após a camada *MP* será um mapa de características dos recursos mais sobressalentes obtido através do mapa anterior. Na Figura 5 é apresentado a operação de *MaxPolling* sobre uma matriz de tamanho 4 x 4 *pixels*.

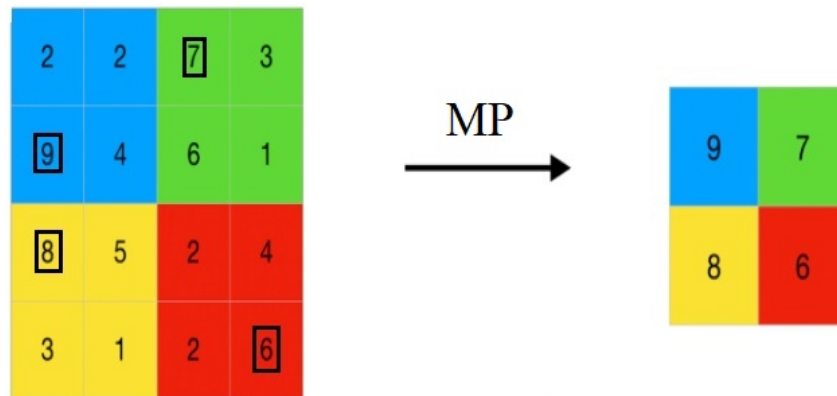


Figura 5 – Demonstração do *MaxPooling* em uma matriz 4 x 4 pixels, resultando em um segundo mapa de recursos 2x2. Adaptado de (KHOSLA, 2021)

O outro método comum é o agrupamento médio (do inglês, *average pooling* - *AP*), que diferente do primeiro não escolhe apenas o maior número dentre os elementos da região analisada, mas sim a média daqueles valores. Na Figura 6 há um exemplo do uso da operação de *AP* sobre uma matriz de tamanho 4 x 4 pixels.

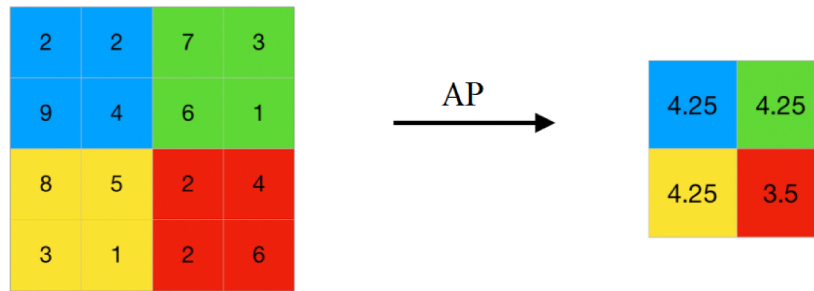


Figura 6 – Demonstração do *Average Pooling* em uma matriz 4x4 pixels, resultando em um segundo mapa de recursos 2x2. Adaptado de (KHOSLA, 2021)

A última camada, a camada totalmente conectada (do inglês, *Fully-Connected Layer*, *FC*), define que cada nó na camada de saída está diretamente conectado diretamente a um nó na camada anterior. Essa camada pode conter vários níveis, todos sempre totalmente conectados. O último nível da *FC* é responsável pela classificação baseada nas características extraídas anteriormente. As camadas *FC* geralmente utilizam a função de ativação *Softmax* para classificar entradas (EDUCATION, 2020). Para treinamento do modelo, a saída do último nível da *FC* é destinada à retro propagação, ou seja, a cada iteração o modelo é levado a aprender as características principais de cada classe (ROSEBROCK, 2017b).

Existem algumas arquiteturas prontas de *CNN* que foram projetadas e testadas em problemas específicos da literatura, como por exemplo, a *ResNet* (HE et al., 2015). Neste trabalho será utilizada de forma parcial, ou seja, parte de suas camadas escolhidas para alimentar outro tipo de arquitetura chamada *CapsuleNet*.

2.1.6 CapsuleNet

Uma *CNN* pode ser uma estratégia adequada em detectar as características das imagens que estão sendo analisadas, mas não é tão efetiva em entender e explorar a relação espacial entre elementos que as constituem. Devido aos métodos *MaxPooling* e *AveragePooling*, discutidos anteriormente, muitas informações a respeito de posições de entidades podem ser perdidas no processo (TOMASINI et al., 2022). Portanto, o trabalho de Hinton (SABOUR; FROSST; HINTON, 2017) introduziu o conceito de cápsulas que englobam grupos de neurônios cujos vetores de atividades representam os parâmetros de instanciação de um tipo único de entidade, como um objeto ou parte dele. É relevante notar que, diferente das *CNNs*, as quais operam com escalares nas saídas dos neurônios nas camadas convolucionais, as cápsulas as tratam como vetores. O comprimento desses vetores é usado para representar a probabilidade de que a entidade exista, e sua direção é usada para representar os parâmetros de instanciação. Um nível de cápsulas ativas prevê parâmetros de instanciação por meio da matriz de transformação. Quando uma boa parte

das previsões são corretas, a "cápsula pai" fica ativa (SABOUR; FROSST; HINTON, 2017). A estrutura Pai se refere a uma cápsula que está em um nível mais alto da rede de cápsulas e recebe informações de várias "filhas", isto é, cápsulas de um nível inferior. É possível fazer uma analogia com uma árvore genealógica de informações, onde as filhas recebem informações de partes menores ou mais simples de um objeto, e a cápsula pai as combinam para reconhecer uma estrutura mais complexa.

A arquitetura básica de uma *CapsuleNet* consiste em uma primeira camada convolucional com a função de ativação *ReLU* e sem agrupamento. Os mapas de características resultantes são passados para a segunda camada chamada *PrimaryCaps*. Ela também é uma camada convolucional, porém utiliza agrupamentos de neurônios (cápsulas) na convolução e gera vetores na saída ao invés de escalares. A última camada representará cada classe possível no problema de classificação através de uma cápsula, onde todas elas estarão conectadas às cápsulas da camada anterior por meio de uma matriz de pesos. Neste vetor estará a informação da chance daquela cápsula representar a imagem analisada. A representação dessa arquitetura pode ser vista na Figura 7, onde ela foi desenvolvida, inicialmente, para a base de dados do Instituto Nacional de Padrões e Tecnologia Modificado (do inglês, *modified national institute of standards and technology* - MNIST (LECUN et al., 1998)).

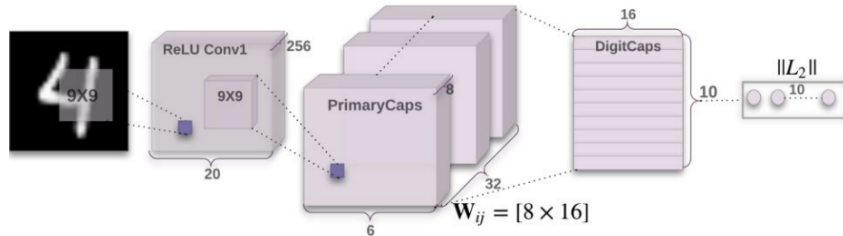


Figura 7 – A arquitetura básica da CapsNet com as três camadas citadas, onde o termo W_{ij} representa uma matriz de pesos que transforma cada cápsula de PrimaryCaps para outra em DigitCaps. Fonte: (SABOUR; FROSST; HINTON, 2017)

Para contornar as limitações discutidas do *MaxPooling*, na *CapsNet* é implementado o roteamento dinâmico (do inglês, *dynamic routing* - DR) que permite que os neurônios em uma camada ignorem os detectores de características menos relevantes nas camadas anteriores, o qual é também utilizado para computar a saída de uma cápsula (SABOUR; FROSST; HINTON, 2017). Esse processo ocorre da seguinte forma, considerando i , uma cápsula na camada *PrimaryCaps*, e a cápsula j , na camada *DigitCaps*, a ativação da cápsula j é determinada a partir das ativações de todas as cápsulas na camada anterior. O coeficiente de acoplamento c_{ij} mede a paridade entre as cápsulas i e j que é computada por um roteamento *Softmax* (SABOUR; FROSST; HINTON, 2017) dado por:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}, \quad (2.1)$$

onde b_{ij} é a notação para probabilidade da cápsula i ser acoplada com a cápsula denominada pai j , a qual é inicializada com valor 0 e refinada de acordo com o processo de roteamento dinâmico (SABOUR; FROSST; HINTON, 2017). Então, são calculadas as entradas da cápsula j pela Equação 2.2 (SABOUR; FROSST; HINTON, 2017):

$$x_j = \sum_i c_{ij} W_{ij} u_i, \quad (2.2)$$

onde x_j é o vetor de entrada da cápsula j , c_{ij} é o coeficiente de paridade, W_{ij} é a matriz de transformação e u_i é a saída da cápsula i .

Na cápsula de saída, o tamanho do vetor x_j representa a probabilidade daquela cápsula representar se a entidade existe, enquanto a orientação do vetor indica suas propriedades. Uma função de compressão denominada *squash* é necessária para garantir que o tamanho do vetor esteja no intervalo de 0 a 1 e introduz a não linearidade a ele, deixando a orientação intacta. A função *squash* pode ser calculada pela Equação 2.3 (SABOUR; FROSST; HINTON, 2017):

$$v_j = \frac{\|x_j\|^2}{1 + \|x_j\|^2} \frac{x_j}{\|x_j\|}, \quad (2.3)$$

onde v_j é o vetor final de saída da cápsula j .

Na arquitetura *CapsNet*, a camada de saída (*DigitCaps*) possui a funcionalidade de não somente reconhecer a entrada, mas também de reconstruir a imagem. Como ela foi designada para o MNIST, onde as imagens possuem resolução 28 x 28 pixels em um canal de cor somente, a reconstrução não requer tanto processamento da rede. No entanto, no reconhecimento de microexpressões as imagens possuem três canais de cores com resolução 224 x 224 pixels. Esse fato inviabiliza a utilização da *CapsNet* para reconstrução de imagens. Por esse motivo, a *CapsNet* com uma arquitetura adaptada pode ser aplicada, onde a última camada só será utilizada para reconhecimento das microexpressões.

Por fim, para treinamento da rede é utilizada uma função de margem de erro (L_k), calculada pela Equação 2.4:

$$L_k = T_k \cdot \max(0, m^+ - \|\mathbf{v}_k\|)^2 + \lambda \cdot (1 - T_k) \cdot \max(0, \|\mathbf{v}_k\| - m^-)^2 \quad (2.4)$$

onde T_k é igual a 1 se um objeto da classe k está presente na imagem, m^+ igual a 0,9, m^- igual a 0,1 e λ por padrão é igual a 0,5 (SABOUR; FROSST; HINTON, 2017). Dessa forma, a perda marginal é zero se $\|v_c\|$ é maior que 0,9 e T_k igual a 1 ou $\|v_c\|$ menor que 0,1 e T_k igual a zero.

2.2 Método de Avaliação

O conjunto dos *frames* foram divididos em conjuntos de treinamento e teste usando um protocolo de validação *LOSO* (do inglês, *Leave-One-Subject-Out*). Em cada iteração de validação cruzada, o conjunto de amostras de um sujeito é escolhida para teste, enquanto todas as outras constituem o conjunto de treinamento. Desta forma, a avaliação do experimento, de certo modo, independe do sujeito. O protocolo *LOSO* é o protocolo padrão dos estudos de reconhecimento das microexpressões (OH et al., 2018).

Duas métricas são avaliadas nos experimentos, sendo elas: *Recall* não ponderado (UAR) e *F1-score* não ponderado (UF1). Tanto UAR quanto UF1 são métricas que visam balancear a avaliação dos resultados considerando a base de dados desbalanceada (de acordo com a Tabela 2). A medida UF1 é calculada a partir do F1-score (F1) obtido em cada classe e calculando a média desse valor, de acordo com as Equações 2.5 e 2.6:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (2.5)$$

$$UF1 = \frac{1}{C} \sum_c F1_c \quad (2.6)$$

onde C é o número de classes, FP é a quantidade de falsos positivos e FN o valor de falsos negativos.

Por fim, a métrica UAR pode ser considerada a acurácia equilibrada e é obtida através da seguinte Equação 2.7:

$$UAR = \frac{1}{C} \sum_c Acc_c \quad (2.7)$$

Acc pode ser definida como a acurácia por classe. Deste modo, O UAR é útil porque traz um balanço da acurácia do modelo para todas as classes, independentemente do número de amostras em cada uma.

2.3 Estado da Arte

Os trabalhos que abordam o reconhecimento automático de microexpressões diferem, sobretudo, da maneira como transcrevem as informações contidas nas imagens, ou sequência delas, em dados capazes de serem utilizados pelo modelo para classificá-los. Outro ponto a se considerar é quais classificadores são usados, podendo ser algoritmos de aprendizado de máquina, redes neurais ou combinações destes.

Dados essas características, o estudo proposto por Jun Ou (OU et al., 2010) explorou o reconhecimento automático das seis expressões faciais universais (raiva, tristeza,

alegria, surpresa, medo e desgosto) utilizando os 28 pontos-chave de características faciais para detecção da face juntamente com um filtro *Gabor* para análise de textura. O método foi testado no banco de imagens *Cohn-Kanade AU-Coded Facial Expression Database* (KANADE; COHN; TIAN, 2000), cujas amostras consistem nas seis expressões faciais em aproximadamente 500 imagens de 100 pessoas diferentes. Para classificação, os autores utilizaram o algoritmo de vizinhos *k*-mais próximos (do inglês, *K-nearest neighbour - KNN*) que funciona basicamente calculando os vizinhos mais próximos a um parâmetro de teste dos casos de treinamento, com base nas distâncias euclidianas entre eles. Com essa configuração, o estudo alcançou 83% de taxa de reconhecimento das seis expressões faciais. A limitação que os autores apresentaram é que a eficácia da extração das característica de expressões faciais é completamente dependente do pré-processamento das imagens originais, sendo necessário buscar uma metodologia eficiente.

O trabalho que introduziu o banco de imagens aplicado a estudar microexpressões, o CASME II, foi apresentado pelos autores em (YAN et al., 2014), banco de imagens com 247 amostras de vídeo a partir de 26 participantes, com exemplos de microexpressões espontâneas de cinco categorias diferentes, sendo elas felicidade, repressão, desgosto, surpresa e outros. Nesse estudo, utilizou-se do método padrões binários locais de três planos ortogonais *LBP - TOP* para descrição das sequências de imagens. Essa técnica descreve um padrão de textura local comparando e limitando os valores de cinza de seus pixels vizinhos. Como classificador, adotou-se a técnica de aprendizado de máquina, máquina de vetores de suporte (do inglês, *support vector machine - SVM*). Para as 5 classes de microexpressões, a melhor performance obtida foi de 63,41% de acurácia, utilizando o método de validação cruzada. Os autores destacam que os métodos não são apropriados para observação e detecção de microexpressões.

Há também tentativas de avaliar o método de fluxo óptico para entender os movimentos da face durante as microexpressões. O estudo de Liu (LIU et al., 2016) desenvolve um modelo para computar o fluxo óptico em cada *frame* do clipe da microexpressão. A partir disso, o algoritmo de classificação SVM foi escolhido. A taxa de reconhecimento utilizando esse método para a base de dados CASME II foi 67,37%, utilizando a validação cruzada *LOSO* e superando o estudo base (YAN et al., 2014). Nesse método de validação, o conjunto de dados é dividido de maneira com que os dados de um único sujeito fique fora do treinamento a cada iteração. Dessa forma, o *LOSO* permite que o modelo aprenda padrões generalizáveis das microexpressões - como AUs específicas de uma ME específica - e não características individuais. Além disso, o desempenho desse modelo, quanto a complexidade computacional, também foi melhor. Isso se deve ao fato da dimensão do recurso fluxo óptico ser baixa. Apesar disso, uma limitação desse trabalho é que os ganhos na taxa de reconhecimento das microexpressões não foram expressivos em relação a trabalhos prévios e dessa forma necessita uma maior exploração e refinamento das metodologias para essa finalidade.

O trabalho de Barbieri (BARBIERI, 2018) consistiu em analisar não somente um método de descrição e classificação, mas a combinação de descritores como características geométricas faciais, fluxo óptico, LBP-TOP e histogramas de gradiente orientado (do inglês, *Histograms of Oriented Gradients*, *HOG*) juntamente com classificadores como SVM, KNN, floresta aleatória (do inglês, *Random Forests*, *RF*) e estímulo adaptativo (do inglês, *Adaptive Boosting*, *AdaBoost*). O melhor desempenho para a base de dados CASME II com o LOSO foi de 68,57% de acurácia no reconhecimento de microexpressões. A autora destaca que aplicando algoritmos de redes neurais profundas em combinação com os descritores abordados pode-se obter resultados promissores no reconhecimento de microexpressões.

Em 2019, o método proposto pelos autores em (QUANG; CHUN; TOKUYAMA, 2019) propõe a utilização de Redes Neurais de Cápsula (do inglês, *Capsule Neural NetWork* - *CapsNet*) que consistem em cápsulas como grupo de neurônios que representam várias propriedades de entidades em um vetor ao invés de um escalar. A classificação das microexpressões foi baseada somente em um *frame*, o *apexframe*. Essa rede é capaz tanto de extrair informações da imagem quanto utilizá-las na classificação da amostra. Uma diferença é que as classes de microexpressão são simplificadas em 3 grupos. Deste modo, para a base de imagens CASME II foi obtida uma acurácia ponderada (do inglês, *unweighted average recall* - *UAR*) de 0,7. A grande desvantagem dessa metodologia é que a *CapsNet*, por se tratar de uma rede neural profunda, requer um alto poder computacional e, em relação aos outros métodos discutidos, é o método que necessita de mais tempo para treinamento do modelo.

Nessa mesma linha de raciocínio, em (SHU et al., 2023), os autores propõe uma abordagem de cápsulas, mas utilizando o modelo *Res2Net* para extrair características. Além disso, é incorporado um módulo de atenção em uma de suas camadas para enfatizar as informações faciais mais determinantes. Os autores também utilizaram o *apexframe* para basear a classificação das MEs e o modelo demonstrou desempenho no banco de dados CASME II para UF1 e UAR igual a 0.8122 e 0.8131, respectivamente.

2.4 Considerações Finais

Diante dos estudos acerca do reconhecimento de microexpressões, a exploração da rede *CapsuleNet* como metodologia desse desafio ainda é bem escassa. Por esse motivo, neste trabalho foram explorados parâmetros da rede para sua otimização, bem como maior exploração no aumento de dados com objetivo de torná-la mais robusta e eficiente.

3 Metodologia

Esta seção traz as informações sobre o banco de imagens utilizado no trabalho, a arquitetura da CapsNet para o reconhecimento de microexpressões bem como as técnicas de aumento de dados empregadas e avaliadas.

3.1 Introdução

Nesta seção, são descritas as etapas aplicadas ao banco de imagens, bem como as etapas exploradas para o desenvolvimento da metodologia proposta. Os estágios da metodologia estão relacionados às etapas de análise e reconhecimento das microexpressões. Na Figura 8, são ilustradas as principais etapas investigadas nesse estudo.

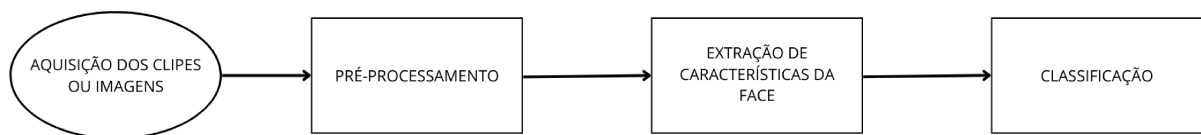


Figura 8 – Fluxograma com as principais etapas para reconhecimento das microexpressões.

Todos os experimentos foram executados no ambiente de serviços de notebooks hospedados do Jupyter, Google Colaboratory ([COLABORATORY, 2017](#)) em Python versão 3.7.13 e utilizando as seguintes bibliotecas: PyTorch ([PASZKE et al., 2017](#)), NumPy ([HARRIS et al., 2020](#)), scikit-learn ([PEDREGOSA et al., 2011](#)) e pandas ([MCKINNEY, 2010](#)). [Link](#) para o código no github.

3.2 Banco de Imagens

Para o reconhecimento assistido por computador das expressões faciais é necessário primeiramente um banco de imagens confiável, de modo a padronizar as imagens, não causando distorções nas estratégias computacionais que serão investigadas para a tarefa de detecção e reconhecimento das microexpressões. O banco de imagens CASME II é de domínio público e empregado em diversos estudos da literatura ([YAN et al., 2014](#)). Ele foi selecionado para esse estudo devido as características de controle presentes na obtenção das imagens.

Na realização deste trabalho foi empregada a base de dados produzida pela Academia Chinesa de Ciências (CASME II), que consiste em vídeo clipes com demonstrações de microexpressões espontâneas ([YAN et al., 2014](#)). Item a) Para tentar reproduzir com maior fidelidade as microexpressões, foi pedido aos participantes que assistissem a clipes

de filmes que transmitiam uma emoção, mas eles deveriam manter a face neutra, ou seja, não demonstrar o sentimento que estavam sentindo. Desta forma, esse arranjo (ver Figura 9) fazia com que os participantes escondessem o que estavam sentindo, criando uma situação favorável para expressões faciais involuntárias dos indivíduos. Essa é uma base de domínio pública disponível com autorização dos autores.



Figura 9 – Arranjo do aparato para aquisição dos cliques de microexpressões. Adaptado de (YAN et al., 2014)

Para filmagem foi utilizada uma câmera de alta velocidade que captura até 200 *frames* por segundo e uma resolução de 280 x 340 *frames* na região facial. Foram recrutados 35 participantes no total com 22,03 média de idade, sendo que em 19 desses foi possível distinguir pelo menos uma microexpressão. No total foram observados 247 exemplos divididos entre 5 classes, denominadas felicidade, repressão, surpresa, nojo e outro. É importante notar que as MEs foram rotuladas baseadas nas AUs, relatos dos participantes e conteúdos dos vídeos (YAN et al., 2014). Na Figura 10 é possível observar a sequência de uma das amostras disponíveis.

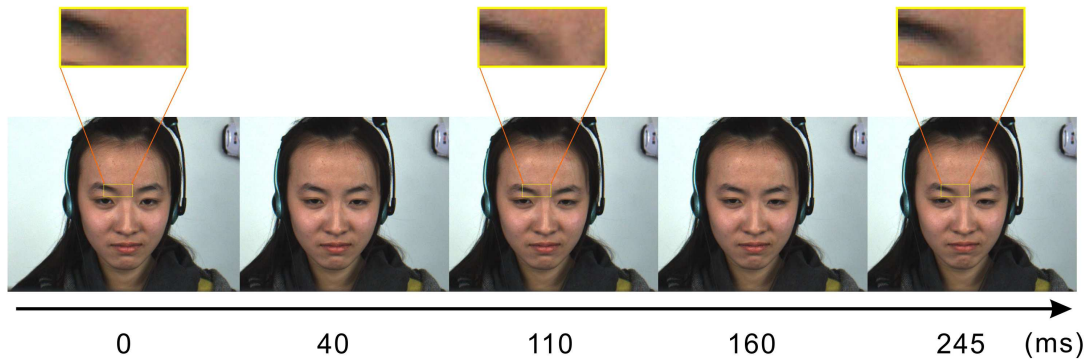


Figura 10 – Exemplo da sequência de frames em um clipe do banco de imagens CASME II. O frame em 110 ms representa o ápice de intensidade da microexpressão, indicando desgosto através das unidades de ação do FACS. Fonte: (YAN et al., 2014)

Para facilitar o reconhecimento das microexpressões e a comparação dos resultados com outros trabalhos, as categorias foram agrupadas em três tipos: positiva, negativa e surpresa, restando 145 amostras. Dessa forma, elas ficaram distribuídas da seguinte forma:

Tabela 2 – Distribuição das três classes de microexpressões.

Classe	Quantidade
Positiva	32
Negativa	88
Surpresa	25

Para treinamento do modelo não foram utilizados todos os *frames* de uma sequência, apenas aqueles rotulados como quadros de ápice (do inglês, *apex frames*) e seus quatro vizinhos (QUANG; CHUN; TOKUYAMA, 2019), isto é, um total de cinco *frames* de cada amostra de ME foram escolhidos e passados como entrada do modelo separadamente. Isso funciona como um aumento de dados e ajuda o modelo não decorar as entradas e perder sua capacidade de generalizar, visto que as amostras estão em poucas quantidades.

As imagens não foram pré-processadas, pois os autores do banco de imagens CASME II já fornecem as sequências de *frames* com as faces recortadas. Também não foi necessário desenvolver um método para obter-se os *apex frames*, visto que estes foram disponibilizados pelos autores.

3.3 Arquitetura CapsNet

Como a primeira camada de uma *CapsuleNet* foi projetada para processar imagens de dimensões menores em relação as consideradas neste trabalho, não é interessante redimensioná-las, pois isso implicaria em perdas de informações importantes para o reconhecimento das microexpressões, levando em conta que a intensidade delas é mínima.

Dessa forma, uma maneira de reduzir a dimensão da entrada sem perder tantas informações é incorporar camadas convolucionais (*backbone*) antes da *PrimaryCaps*. Para isso, optou-se utilizá-las a partir de redes previamente treinadas em outro banco de imagem em que neste caso foi a *ResNet*. As vantagens que a rede *ResNet* (*Residual Network*) traz são os blocos residuais, que possuem a capacidade de pular uma ou várias camadas se esta estiver causando prejuízos na classificação. Dessa maneira, o problema de saturação do gradiente causado pela alta profundidade da rede é contornado (YU LAURENCE T. YANG, 2021).

A quantidade de camada da rede *ResNet* pode variar, como exemplo, existem as arquiteturas com 18, 50, 34, 101 ou 152 camadas. Neste trabalho são utilizados os modelos *ResNet18* e *ResNet50* pré treinados no banco de imagem ImageNet (DENG et al., 2009). Como exemplo, na Figura 11 mostra a arquitetura da *ResNet18*.

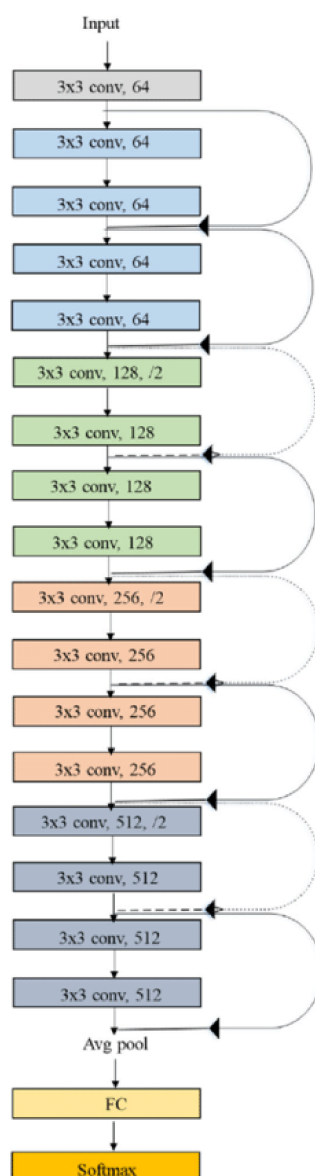


Figura 11 – Camadas da rede ResNet18. Fonte: (RAMZAN et al., 2019)

As camadas que foram utilizadas para obter as características das imagens e assim servir como entrada para *CapsNet* foram: a primeira camada de convolução (representada na cor cinza na Figura 11) e os três próximos blocos de camadas convolucionais (representados nas cores azul, verde e vermelho na Figura 11), não sendo aproveitado o último bloco nem a camada totalmente conectada (representados nas cores cinza e amarelo na Figura 11). O que diferencia as arquiteturas *ResNet18* e *ResNet50* é a quantidade de camadas convolucionais em cada bloco e o tamanho do filtro para *MaxPooling*. Enquanto a primeira tem apenas duas camadas convolucionais em cada um dos três blocos utilizados, a rede *ResNet50* possui três no primeiro, quatro no segundo e seis no terceiro.

Deste modo, a medida que uma arquitetura similar com a representação na Figura 7 recebe os dados dos blocos convolucionais da rede *ResNet* e as cápsulas da camada *PrimaryCaps* são construídas. Cada uma das cápsulas primárias encapsula as informações e características de uma entidade em um vetor onde a atividade dos neurônios em uma cápsula descreve os parâmetros dessa entidade. Na última camada *DigitCaps* estão três cápsulas somente, cada uma referenciando uma classe. A arquitetura final para classificação das microexpressões pode ser vista na Figura 12.

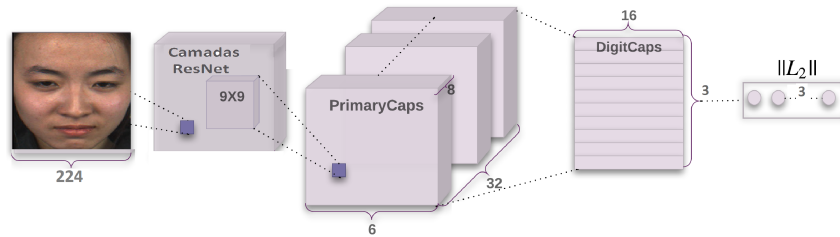


Figura 12 – Estrutura completa da arquitetura final de CapsNet. Adaptado de (SABOUR; FROSST; HINTON, 2017). Camadas ResNet se referem às camadas convolucionais da arquitetura *ResNet*, a camada *PrimaryCaps* diz respeito às cápsulas primárias que detectam características específicas e respectivas orientações. Por fim, a camada *DigitCaps* (cápsulas dígitos) é a camada responsável pela classificação. O termo $\|L_2\|$, dito norma do vetor, significa o comprimento desse vetor e pode ser interpretado como a probabilidade da presença de uma determinada classe.

3.4 Aumento de Dados

O aumento de dados (do inglês, *Data Augmentation*, *DA*) consiste nas diversas técnicas utilizadas na geração de novas amostras para treinamento a partir das originais (ROSEBROCK, 2017a). O objetivo dessa prática é melhorar a generalização do modelo, isto é, aumentar a quantidade de classificações corretas em amostras de dados novas. Basicamente, as técnicas de DA geram novas amostras com pequenas modificações que permitem que a rede aprenda características mais robustas (ROSEBROCK, 2017a). Essas técnicas são aplicadas somente nos dados de treinamento e podem ser feitas para aumentar

o tamanho total do grupo de teste ou apenas modificá-lo. Na maioria dos casos, a acurácia no teste irá aumentar ([ROSEBROCK, 2017a](#)).

O primeiro exemplo da modificação implantada pelo *DA* e que é utilizado e avaliado neste trabalho é a rotação aleatória da imagem. Essa técnica escolherá um ângulo de rotação randômico entre 0 e 359 graus e aplicará ao lote de imagens selecionados pelo algoritmo. Essa técnica a rotação randômica ajuda o modelo a aprender a reconhecer objetos independentemente da orientação. O resultado pode ser visto na [Figura 13a](#) e [Figura 13b](#).



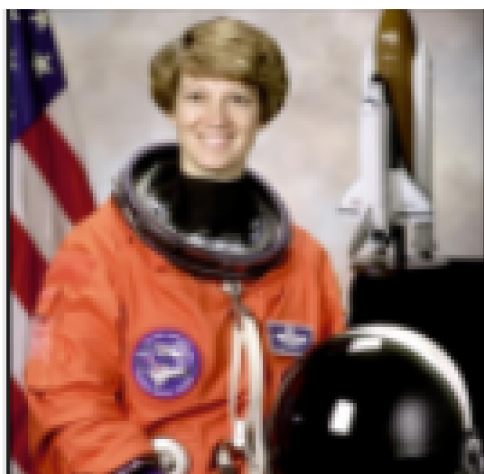
(a) Imagem original.



(b) Imagem após rotação randômica.

Figura 13 – Essa técnica de *DA* consiste em rotacionar a imagem original (a) em um ângulo randômico, resultando na imagem (b). Fonte: ([TRANSFORMS, 2020](#))

A próxima técnica de *DA* avaliada é o espelhamento horizontal da imagem que consiste em invertê-la de acordo com o eixo horizontal ([Figura 14a](#) e [Figura 14b](#)). Ela será aplicada ao lote de imagens com uma determinada probabilidade. Essa técnica auxilia o modelo a focar nas características que definem as microexpressões (como as UAs) em vez de memorizar orientações específicas.



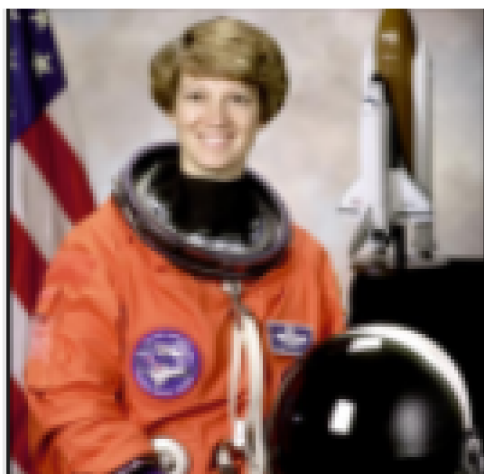
(a) Imagem original.



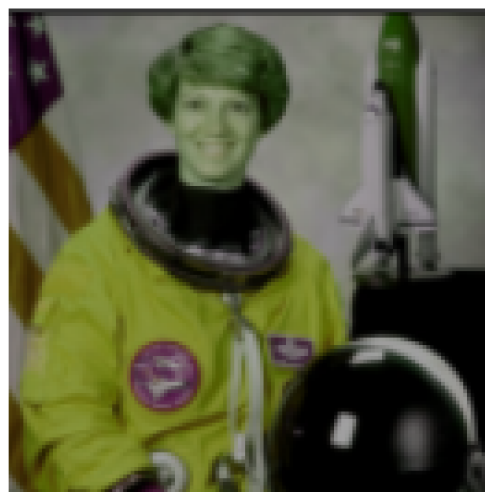
(b) Imagem após espelhamento.

Figura 14 – Essa técnica de *DA* consiste em espelhar horizontalmente a imagem (a), resultando na imagem (b). Fonte: ([TRANSFORMS, 2020](#))

Em seguida, a técnica de aumento de dados verificada consiste na variação de cor (do inglês, *Collor Jitter, CJ*). Essa técnica modifica aleatoriamente o brilho, contraste, saturação e matiz da imagem original. Os efeitos causados pela técnica *CJ* são vistos na [Figura 15a](#) e [Figura 15b](#). Ao introduzir mudanças aleatórias nas cores, essa técnica ajuda o modelo a aprender a identificar padrões importantes nas imagens, mesmo quando as cores ou a iluminação mudam de forma branda.



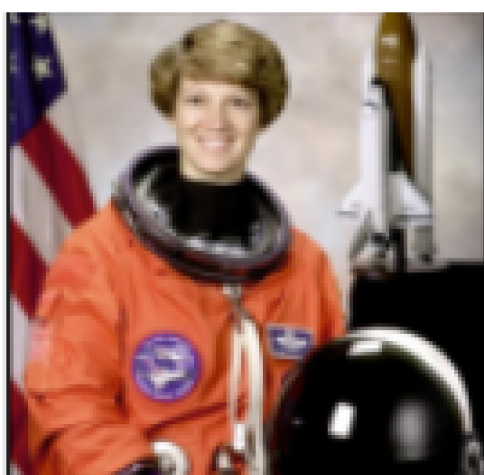
(a) Imagem original.



(b) Imagem após instabilidade na cor.

Figura 15 – Essa técnica de *DA* consiste em alterar brilho, contraste, saturação ou matiz da imagem (a), resultando na imagem (b). Fonte: ([TRANSFORMS, 2020](#))

Na sequência, a técnica analisada é a recorte aleatório (do inglês, *Random Crop*) que corta a imagem em uma posição aleatória. O tamanho do recorte deve ser configurado previamente no algoritmo. Dessa forma pode-se observar o resultado dessa técnica na [Figura 16a](#) e [Figura 16b](#). Utilizando essa técnica de recortes aleatórios, o modelo aprende a reconhecer objetos mesmo que não estejam centralizados, tornando-se mais generalista.



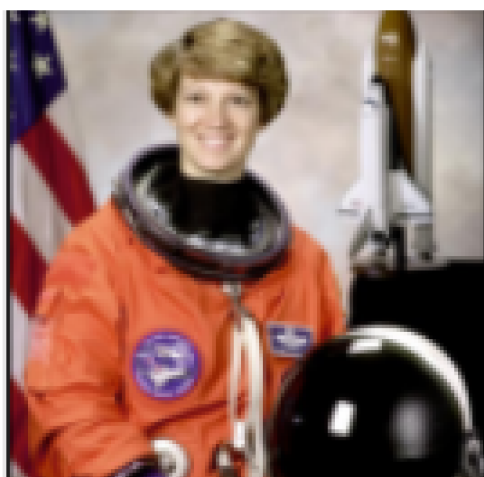
(a) Imagem original.



(b) Imagem após recorte aleatório.

Figura 16 – Essa técnica de *DA* aleatoriamente escolhe uma posição na imagem (a), e a recorta, resultando na imagem (b). Fonte: ([TRANSFORMS, 2020](#))

Além disso, a técnica de aumento de dados chamada escala de cinza foi examinada. Ela tem uma chance de mudar as imagens daquele lote para escala de cinza, isto é, para uma escala onde a intensidade de cada pixel pode variar de preto total (intensidade mínima) a branco total (maior intensidade possível). Sua aplicação é mostrada na [Figura 17a](#) e [Figura 17b](#). Ao treinar o modelo para aprender as características por meio de imagens em preto e branco, o risco de que as cores possam influenciar negativamente no processo é diminuído.



(a) Imagem original.

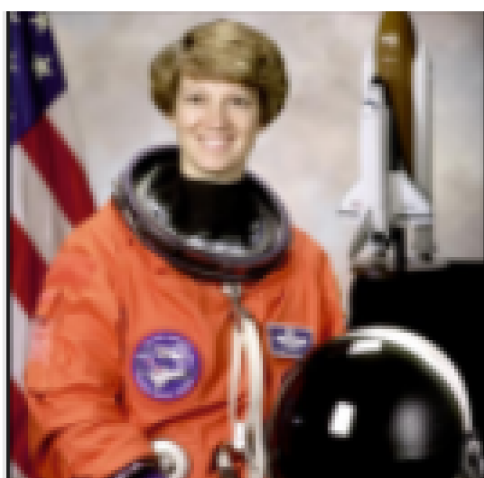


(b) Imagem após a transformação para escala de cinza.

Figura 17 – Essa técnica de *DA* aplica, com uma certa chance, na imagem (a), a transformação para escala de cinza, resultando na imagem (b). Fonte: ([TRANSFORMS, 2020](#))

Ademais, a técnica de *DA* equalização randômica (do inglês, *Random Equalize*) também foi analisada. Essa técnica consiste em aplicar, com uma probabilidade determi-

nada, a equalização de histograma de maneira aleatória em um lote de imagens durante o pré-processamento dos dados de treinamento. O histograma de uma imagem se diz respeito à distribuição de intensidades de pixel em toda a imagem. A equalização do histograma é uma transformação que redistribui os *pixels* de uma imagem para chegar a uma distribuição mais uniforme. O resultado da equalização é visto na [Figura 18a](#) e [Figura 18b](#). O uso dessa técnica pode melhorar o desempenho do modelo ao equalizar o contraste nas imagens durante o treinamento, permitindo que ele aprenda tanto com as regiões mais claras quanto com as mais escuras, desta forma, tornando-se mais tolerante a variações de iluminação e diferenças de textura.



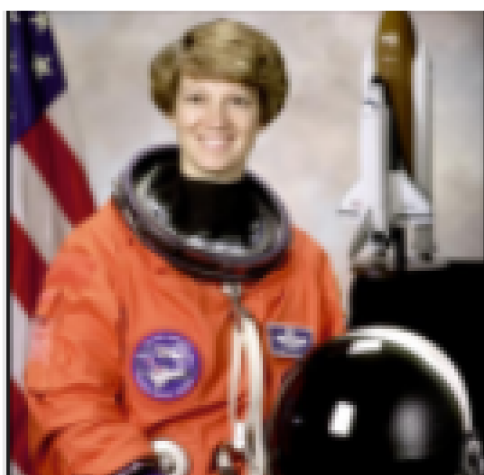
(a) Imagem original.



(b) Imagem após equalização.

Figura 18 – Essa técnica de *DA* equaliza aleatoriamente o histograma dos pixels da imagem (a), resultando na imagem (b). Fonte: ([TRANSFORMS, 2020](#))

Por fim, a última técnica de *DA* analisada é o ajuste randômico do contraste (do inglês, *Random Auto Contrast - RAC*). Nela, é feito um ajuste automático de contraste no lote de imagens durante o pré-processamento. Neste caso, o contraste é a diferença entre as intensidades de pixel em diferentes partes da imagem e a correção automática dele é um processo que visa melhor adequar a distribuição de intensidades de pixel para ampliar a faixa de contraste e melhorar a visualização dos detalhes. O resultado do ajuste é observado na [Figura 19a](#) e [Figura 19b](#). No cenário dos cliques de microexpressões, apesar de todo controle com o cenário, as condições de iluminação variam. O ajuste randômico do contraste pode simular essas variações, ajudando o modelo a aprender a reconhecer objetos sob diferentes iluminações.



(a) Imagem original.



(b) Imagem após ajuste no contraste.

Figura 19 – Essa técnica de *DA* ajusta aleatoriamente o contraste da imagem (a), e como resultado obtém-se a imagem (b). Fonte: ([TRANSFORMS, 2020](#))

Em suma, cada técnica de aumento de dados promoverá, de forma aleatória, a variação dos dados de entrada. Isso permitirá aumentar o número de amostras para o modelo e, com isso, a abordagem passa a não depender de aprender detalhes específicos, como cores, texturas e orientação, mas deverá focar mais nos atributos mais importantes que apareçam em todos os dados de entrada, como por exemplos as microexpressões. Isso é importante para a capacidade de generalização do modelo porque mitiga a sua excessiva dependência em um conjunto particular e específico de características nos dados.

3.5 Configuração do Experimento

No contexto deste trabalho, o *batch size* pode ser definido como o tamanho do grupo que o modelo divide os dados de treinamento, para que estes sejam usados de uma só vez para calcular as atualizações de parâmetros. Como exemplo, o *batchsize* é definido como 32, logo 32 imagens separadas para o treinamento são usados para calcular as atualizações de parâmetros antes de passar para o próximo lote (*batch*).

A taxa de aprendizado é o parâmetro que determina a magnitude das atualizações feitas a cada passo de treinamento. Dessa forma, a taxa de aprendizado controla a velocidade com que as atualizações são feitas. Uma taxa de aprendizado maior permite atualizações maiores nos parâmetros, enquanto uma taxa de aprendizado menor resulta em atualizações menores. Já a taxa de decaimento do aprendizado é utilizada para ajustar a taxa de aprendizado à medida que o treinamento avança e seu valor é padronizado como 0,0001.

Por fim, o número de épocas é a quantidade de vezes que o algoritmo de treinamento percorre o conjunto de dados de treinamento durante o treinamento do modelo.

Durante o treinamento de um modelo, os dados de treinamento são divididos em lotes *batches*, como explicado anteriormente, e alimentados ao modelo em iterações. Uma época somente é finalizada quando todos os *batches* de treinamento se esgotarem e forem utilizados para atualização dos parâmetros do modelo. Como forma de padronização nos experimentos, os seguintes parâmetros são fixados no algoritmo: tamanho de lote (do inglês, *batch size*) igual a 32, a taxa de aprendizado igual a 0,0001, a taxa de decaimento do aprendizado de 0,9 e a quantidade de épocas de treinamento igual a 30, de acordo com o trabalho de Nguye ([QUANG; CHUN; TOKUYAMA, 2019](#)).

4 Experimento e Resultados

Neste capítulo será apresentada a estratégia adotada para avaliação do método proposto para o reconhecimento de microexpressões. Os resultados serão discutidos e comparados com os relatados na literatura.

4.1 Avaliação dos Modelos e Aumento de Dados

O experimento consiste em comparar a eficácia de dois *backbones* em diferentes protocolos de aumento de dados no reconhecimento de microexpressões. Para isso são utilizadas as arquiteturas *ResNet18* e *ResNet50*, como *backbone*, e três protocolos diferentes de aumento de dados, cada um deles com quantidades de técnicas divergentes. Dessa forma, além de avaliar os *backbones* no reconhecimento das microexpressões, pode-se entender se uma maior variedade de técnicas de *DA* está relacionada a melhora acurácia da rede para este propósito.

O primeiro protocolo de *DA* (DA-1) consiste em aplicar apenas o recorte aleatório (Figura 16b) nos lotes de imagens designadas ao treinamento. Dessa forma, esse protocolo é o mais brando quanto a utilização de técnicas de aumento de dados. O segundo protocolo (DA-2) consiste em quatro técnicas de aumento de dados. São elas: rotação aleatória (Figura 13b), espelhamento horizontal (Figura 14b), *ColorJitter* (Figura 15b) e recorte aleatório (Figura 16b). Esse é o protocolo intermediário de *DA*. Por fim, o terceiro protocolo (DA-3) é o mais severo e utiliza todas as técnicas descritas no capítulo passado. No total, são sete técnicas de aumento de dados. Dessa forma, são seis experimentos onde são investigadas as arquiteturas *ResNet18* e *Resnet50* com os três protocolos de aumento de dados.

4.2 Resultados

Tendo como objetivo examinar o impacto de diferentes protocolos de aumento de dados em conjunto com arquiteturas de rede neural na classificação de microexpressões, os experimentos são executados e os resultados mostrados na Tabela 3.

Tabela 3 – Resultados obtidos utilizando os protocolos de aumento de dados propostos nas arquiteturas de *backbone* *ResNet18* e *Resnet50*.

Métricas Experimento	UAR	UF1	Tempo de execução (s)
DA-1 ResNet18	0,6075	0,5714	1665,541
DA-2 ResNet18	0,6577	0,6778	3498,379
DA-3 ResNet18	0,6077	0,6133	4553,945
DA-1 ResNet50	0,5955	0,6029	2461,220
DA-2 ResNet50	0,6636	0,6795	5109,096
DA-3 ResNet50	0,7493	0,7727	6653,682

Como o protocolo LOSO foi utilizado, foi computado o resultado para UAR para cada iteração dele, isto é, quando um sujeito é deixado de fora do treinamento e é realocado para o grupo dos dados de teste. Desta forma, pode-se observar a acurácia balanceada por classe para cada iteração do protocolo na [Figura 20](#).

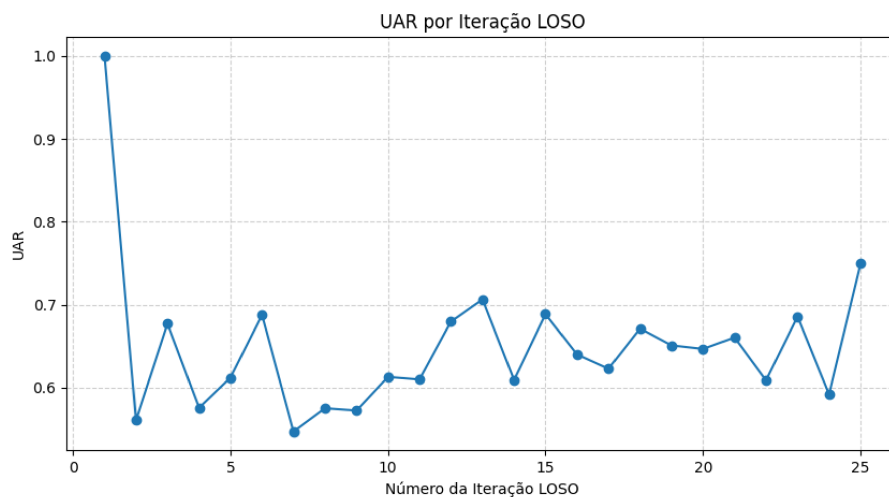


Figura 20 – Métrica UAR a cada iteração do método *LOSO* no experimento DA-3 ResNet50. Fonte: De autoria própria.

Como forma de entender e visualizar as classificações feitas pela rede com melhor desempenho, uma matriz de confusão é construída com auxílio da biblioteca *scikit-learn* (PEDREGOSA et al., 2011). Através dela, é possível identificar o número de verdadeiros positivos, verdadeiros negativos, falsos positivos para cada classe. Dessa forma, pode-se compreender se a rede está tendo dificuldades para distinguir exemplos de amostras de uma classe. Na [Figura 21](#) é apresentada a matriz de confusão obtida a partir das classificações do experimento DA-3 ResNet50.

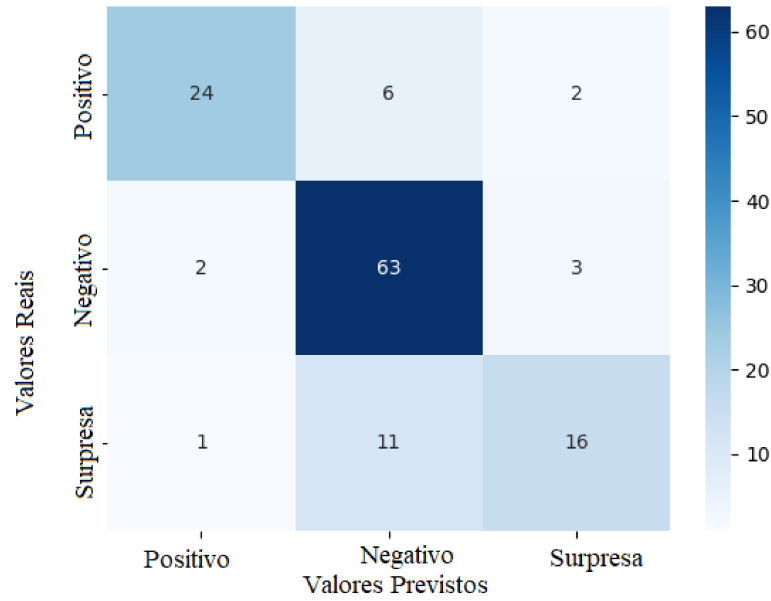


Figura 21 – Matriz de confusão obtida após classificação das microexpressões pela rede utilizando a arquitetura DA-3 ResNet50. Fonte: De autoria própria.

4.3 Análise em relação ao estado da arte

Os resultados são comparados com aqueles encontrados no Estado da Arte e expostos na Tabela 4.

Tabela 4 – Comparação dos resultados com os obtidos por outros trabalhos

Métricas	UAR	UF1
Métodos		
LBP-TOP (YAN et al., 2014)	0,7429	0,7026
CapsuleNet (QUANG; CHUN; TOKUYAMA, 2019)	0,7018	0,7068
RES-CapsNet (SHU et al., 2023)	0,8131	0,8122
Optical Flow (LIU et al., 2016)	0,6159	0,6247
DA-3 ResNet50	0,7493	0,7727

Em relação ao estado da arte, os resultados obtidos para $UF1$ e UAR na configuração DA – 3ResNet50 apresentou resultados comparáveis aos estudos investigados. No entanto, devido ao grande custo computacional desse modelo é necessário considerar a sua viabilidade da aplicação prática.

A princípio, o experimento busca reproduzir os resultados do trabalho de Nguye (QUANG; CHUN; TOKUYAMA, 2019) utilizando uma configuração de experimento semelhante àquela descrita pelos autores, bem como a rede CapsNet com a arquitetura ResNet18 como *backbone*. Os resultados obtidos por este trabalho para UAR e $UF1$, respectivamente, são 0,6577 e 0,6778. O trabalho de Nguye obteve 0,7018 e 0,7068 para as mesmas mensurações. Apesar de próximos, não foi possível a reprodução dos resultados exatos. Isso se deve, provavelmente, às particularidades de configuração na condução do

experimento que não foram expostas pelos autores, impossibilitando assim a replicação exata do experimento.

Em relação ao aumento de dados e o *backbone ResNet18*, pode-se observar através da Tabela 3 que há um platô no benefício da quantidade de perturbação por ele causada, diminuindo a precisão de previsões corretas no experimento *DA-3 Resnet18*. Já quando o *backbone ResNet50* é utilizado, o maior emprego de técnicas de DA melhorou a capacidade da rede em prever corretamente as microexpressões. Esses resultados são compreensíveis, visto que quanto mais parâmetros um *backbone* possui, ou seja, mais complexa ele é, com mais facilidade um *overfitting* poderá ocorrer. Desta forma, a diversidade de técnicas de aumento de dados permite o modelo a generalizar melhor para dados não vistos, principalmente quando se trabalha com quantidades de dados bem limitados, como é o caso neste trabalho.

No que diz respeito a classificação de microexpressões é factível que há uma confusão do modelo na classificação, principalmente, das classes Negativo e Surpresa, sendo que das 28 amostras pertencentes à classe Surpresa, 11 delas são classificadas como Negativo. Há duas hipóteses para explicar o fato. Primeiro que o banco de imagem não é balanceado, isto é, a quantidade de amostras da classe Negativo é mais que o dobro que as de classe surpresa. Isso pode enviesar a rede e distorcer as suas previsões em favor da maior quantidade de amostras, mesmo que no algoritmo exista uma forma de balancear as amostras que são carregadas para treinamento. A segunda tese é da ambiguidade da caracterização das emoções. Por exemplo, nos vídeos que são apresentados aos participantes, enquanto uma ME específica é esperada, as reações verdadeiras são extremamente subjetivas e dependem de como cada participante encara aquela circunstância. Portanto, a própria construção do banco de imagens está sujeita a imprecisões, dado o contexto.

5 Conclusão

Nesse trabalho, foi pesquisado o uso de arquitetura CapsNet no reconhecimento de microexpressões faciais, buscando tirar proveito da sua capacidade de manter relações espaciais entre elementos das imagens. Foram realizados experimentos que buscaram comparar distintas arquiteturas como entrada da CapsNet, como ResNet18 e ResNet50. Além disso foi investigado o efeito de diferentes técnicas de aumento de dados na performance do modelo em classificar MEs.

Os resultados mostram que o modelo baseado em CapsNet teve uma performance razoável em comparação com as abordagens do estado da arte. No entanto, este modelo requer um grande poder computacional devido a complexidade dele. Desta forma, em aplicações em tempo real pode não ser uma abordagem viável.

5.1 Desafios Encontrados

Os desafios encontrados foram o seguintes:

- Um dos principais desafios enfrentados ao longo deste trabalho foi a necessidade de aprofundamento em um tema não abordado de forma completa na engenharia: a psicologia, em particular, na pesquisa das microexpressões faciais.
- Outro desafio encontrado foi o estudo dos conceitos de aprendizado profundo que, apesar de ter disciplinas de programação na engenharia mecatrônica, não são efetivamente abordadas na graduação. Desta maneira, entender e aplicar esses conceitos necessitou de um estudo mais aprofundado do tema.
- Na mesma linha de raciocínio, a complexidade computacional do modelo desenvolvido também impôs dificuldade para testar as diversas configurações discutidas, pois o ambiente de teste escolhido Google Colab tem limitações quanto ao uso de suas unidades computacionais disponíveis. Desta forma, levou-se um tempo maior que o estimado inicialmente para finalizar o teste de todas configurações propostas.

5.2 Trabalhos Futuros

Para um trabalho futuro, espera-se bancos de dados de MEs mais robustos, isto é, com abundância de amostras de Microexpressões de forma balanceadas entre as classes, estejam disponíveis para garantir uma generalização e confiabilidade mais ampla do modelo resultante. Assim sendo, poder-se-á desenvolver melhorias e ajustes ao modelo

proposto. Como exemplo, é válido testar outras arquiteturas como *backbone* da Capsnet e combiná-la algoritmos de aprendizado de máquina para classificação das MEs, criando arquiteturas híbridas. Ademais, pensando na complexidade computacional, é válido em trabalhos futuros investigar a extração de características - como a própria ResNet, HOG, LBP-TOP - e seu impacto na acurácia do modelo Capsnet.

Referências

- ADEGUN, I. P.; VADAPALLI, H. B. Facial micro-expression recognition: A machine learning approach. *Scientific African*, v. 8, p. e00465, 2020. ISSN 2468-2276. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2468227620302039>>. Citado na página 17.
- ASTHANA, A. et al. Robust discriminative response map fitting with constrained local models. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2013*. United States: IEEE, 2013. (IEEE Conference on Computer Vision and Pattern Recognition), p. 3444–3451. ISBN 978-0-7695-4989-7. 10.1109/CVPR.2013.442 ; 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013 ; Conference date: 23-06-2013 Through 28-06-2013. Citado na página 15.
- BARATHI, C. S. Lie detection based on facial micro expression body language and speech analysis. *International Journal of Engineering Research & Technology*, v. 5, n. 2, p. 337–343, 2016. Citado na página 15.
- BARBIERI, L. *Facial microexpression recognition based on descriptor and classifier combinations*. 2018. Citado 3 vezes nas páginas 10, 11 e 25.
- COLABORATORY. Google, 2017. Disponível em: <<https://research.google.com/colaboratory/intl/pt-BR/faq.html>>. Citado na página 26.
- COOTES, T. et al. Active shape models-their training and application. *Computer Vision and Image Understanding*, Elsevier BV, v. 61, n. 1, p. 38–59, jan 1995. Disponível em: <<https://doi.org/10.1006%2Fcvui.1995.1004>>. Citado na página 15.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, v. 1, p. 886–893, 2005. Citado na página 11.
- DANTAS, A. *Abordagem computacional para aprimoramento das habilidades com as emoções em indivíduos com autismo*. Tese (Doutorado), 2022. Disponível em: <<https://doi.org/10.14393/ufu.te.2022.211>>. Citado 3 vezes nas páginas 4, 17 e 18.
- DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Miami, FL, USA: IEEE Computer Society, 2009. p. 248–255. ISBN 978-1-4244-3991-1. Citado na página 29.
- EDUCATION, I. C. *What are convolutional neural networks?* 2020. Disponível em: <<https://www.ibm.com/cloud/learn/convolutional-neural-networks>>. Citado 3 vezes nas páginas 4, 18 e 20.
- EKMAN, P.; AND, W. V. F. Nonverbal leakage and clues to deception†. *Psychiatry*, Routledge, v. 32, n. 1, p. 88–106, 1969. PMID: 27785970. Disponível em: <<https://doi.org/10.1080/00332747.1969.11023575>>. Citado 4 vezes nas páginas 6, 10, 14 e 15.

- EKMAN, P.; FRIESEN, W. V.; HAGER, J. C. *Facial Action Coding System: Manual and Investigator's Guide*. Salt Lake City, UT: Research Nexus, 2002. ISBN 0931835011. Citado 2 vezes nas páginas 4 e 14.
- GROSS, J. J. (Ed.). *Handbook of Emotion Regulation*. 2. ed. New York: The Guilford Press, 2015. ISBN 9781462520732. Citado na página 13.
- HARRIS, C. R. et al. Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <<https://doi.org/10.1038/s41586-020-2649-2>>. Citado na página 26.
- HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>. Citado na página 20.
- KANADE, T.; COHN, J. F.; TIAN, Y. Comprehensive database for facial expression analysis. In: *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000)*. Los Alamitos, CA, USA: IEEE Computer Society, 2000. p. 46–53. Citado na página 24.
- KHOSLA, S. *CNN: Introduction to pooling layer*. 2021. Disponível em: <<https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>>. Citado 3 vezes nas páginas 4, 19 e 20.
- KUMARI, J.; RAJESH, R.; POOJA, K. Facial expression recognition: A survey. *Procedia Computer Science*, p. 486–491, 2015. Citado na página 10.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Institute of Electrical and Electronics Engineers (IEEE), v. 86, n. 11, p. 2278–2324, 1998. Disponível em: <<https://doi.org/10.1109%2F5.726791>>. Citado na página 21.
- LEDOUX, J. *The Emotional Brain: The Mysterious Underpinnings of Emotional Life*. Simon & Schuster, 1998. (Business book summary). ISBN 9780684836591. Disponível em: <<https://books.google.com.br/books?id=7EJN5I8sk2wC>>. Citado na página 13.
- LI, X. et al. Towards reading hidden emotions: A comparative study of spontaneous micro-expression spotting and recognition methods. *IEEE Trans. Affect. Comput.*, IEEE Computer Society Press, Washington, DC, USA, v. 9, n. 4, p. 563–577, out. 2018. ISSN 1949-3045. Disponível em: <<https://doi.org/10.1109/TAFFC.2017.2667642>>. Citado na página 16.
- LI, Y.; HUANG, X.; ZHAO, G. Can micro-expression be recognized based on single apex frame? In: *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*. Athens, Greece: IEEE, 2018. p. 3094–3098. Citado na página 16.
- LIU, Y.-J. et al. A main directional mean optical flow feature for spontaneous micro-expression recognition. *IEEE Transactions on Affective Computing*, Institute of Electrical and Electronics Engineers (IEEE), v. 7, n. 4, p. 299–310, out. 2016. Disponível em: <<https://doi.org/10.1109/taffc.2015.2485205>>. Citado 2 vezes nas páginas 24 e 39.
- MCKINNEY, W. Data structures for statistical computing in python. In: WALT, S. van der; MILLMAN, J. (Ed.). *Proceedings of the 9th Python in Science Conference (SciPy 2010)*. Austin, Texas, USA: SciPy, 2010. p. 56–61. Citado na página 26.

- MIAO, S. et al. Recognizing facial expressions using a shallow convolutional neural network. *IEEE Access*, v. 7, p. 78000–78011, 2019. Citado na página 11.
- MIGUEL, F. K. Psicologia das emoções: uma proposta integrativa para compreender a expressão emocional. *Psico-USF*, FapUNIFESP (SciELO), v. 20, n. 1, p. 153–162, abr. 2015. ISSN 1413-8271. Disponível em: <<http://dx.doi.org/10.1590/1413-82712015200114>>. Citado na página 13.
- MOOR, J. The dartmouth college artificial intelligence conference: The next fifty years. *AI Magazine*, v. 27, n. 4, p. 87, Dec. 2006. Disponível em: <<https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1911>>. Citado na página 17.
- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2010. (ICML'10), p. 807–814. ISBN 9781605589077. Citado na página 19.
- OH, Y.-H. et al. A survey of automatic facial micro-expression analysis: Databases, methods, and challenges. *Frontiers in Psychology*, v. 9, 2018. ISSN 1664-1078. Disponível em: <<https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2018.01128>>. Citado na página 23.
- OU, J. et al. Automatic facial expression recognition using gabor filter and expression analysis. In: *2010 Second International Conference on Computer Modeling and Simulation*. IEEE, 2010. Disponível em: <<https://doi.org/10.1109/iccms.2010.45>>. Citado na página 23.
- PASZKE, A. et al. Automatic differentiation in pytorch. In: *NIPS 2017 Workshop on Autodiff*. Long Beach, California, USA: [s.n.], 2017. Disponível em: <<https://openreview.net/forum?id=BJJsrnfCZ>>. Citado na página 26.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, v. 12, n. Oct, p. 2825–2830, 2011. Citado 2 vezes nas páginas 26 e 38.
- PFISTER, T. et al. Recognising spontaneous facial micro-expressions. In: *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, 2011. p. 1449–1456. Citado na página 11.
- QUANG, N. V.; CHUN, J.; TOKUYAMA, T. Capsulenet for micro-expression recognition. *14th IEEE International Conference on Automatic Face Gesture Recognition*, 2019. Citado 4 vezes nas páginas 25, 28, 36 e 39.
- RAMZAN, F. et al. A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks. *Journal of Medical Systems*, v. 44, 12 2019. Citado 2 vezes nas páginas 4 e 29.
- ROSEBROCK, A. *Deep Learning for Computer Vision with Python: Practitioner Bundle*. [S.l.]: PyImageSearch, 2017. Citado 2 vezes nas páginas 30 e 31.
- ROSEBROCK, A. *Deep Learning for Computer Vision with Python: Starter Bundle*. PyImageSearch, 2017. Disponível em: <<https://books.google.com.br/books?id=9UI-tgEACAAJ>>. Citado na página 20.

- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, v. 65, p. 386–408, 1958. Citado na página 17.
- RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3. ed. [S.l.]: Prentice Hall, 2010. Citado na página 17.
- RUSSELL, T. A.; CHU, E.; PHILLIPS, M. L. A pilot study to investigate the effectiveness of emotion recognition remediation in schizophrenia using the micro-expression training tool. *British journal of clinical psychology*, Wiley Online Library, v. 45, n. 4, p. 579–583, 2006. Citado na página 15.
- SABOUR, S.; FROSST, N.; HINTON, G. E. *Dynamic Routing Between Capsules*. arXiv, 2017. Disponível em: <<https://arxiv.org/abs/1710.09829>>. Citado 5 vezes nas páginas 4, 20, 21, 22 e 30.
- SHU, X. et al. Res-capsnet: An improved capsule network for micro-expression recognition. *Multimedia Systems*, v. 29, n. 3, p. 1593–1601, Mar 2023. Citado 2 vezes nas páginas 25 e 39.
- SZELISKI, R. *Computer Vision: Algorithms and Applications*. London: Springer, 2010. ISBN 978-1-84882-934-3. Citado na página 11.
- TOMASINI, U. M. et al. *How deep convolutional neural networks lose spatial information with training*. 2022. Disponível em: <<https://arxiv.org/abs/2210.01506>>. Citado na página 20.
- TRANSFORMS. 2020. Disponível em: <https://pytorch.org/vision/main/auto_examples/plot_transforms.html#sphx-glr-auto-examples-plot-transforms-py>. Citado 6 vezes nas páginas 5, 31, 32, 33, 34 e 35.
- TURING, A. M. Computing machinery and intelligence. *Mind*, Oxford University Press on behalf of the Mind Association, v. 59, n. 236, p. 433–460, 1950. ISSN 00264423. Disponível em: <<http://www.jstor.org/stable/2251299>>. Citado na página 16.
- WU, H.-Y. et al. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph.*, Association for Computing Machinery, New York, NY, USA, v. 31, n. 4, jul 2012. ISSN 0730-0301. Disponível em: <<https://doi.org/10.1145/2185520.2185561>>. Citado na página 15.
- YAN, W. et al. Casme ii: An improved spontaneous micro-expression database and the baseline evaluation. *PLoS One*, 2014. Citado 7 vezes nas páginas 4, 10, 24, 26, 27, 28 e 39.
- YAN, W. et al. How fast are the leaked facial expressions: The duration of micro-expressions. *Journal of Nonverbal Behavior*, 2013. Citado na página 10.
- YU LAURENCE T. YANG, Q. Z. D. A. M. J. D. H. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. *Elsevier*, 2021. Citado na página 29.