



Fundação Universidade Federal do ABC

Pró reitoria de pesquisa

Av. dos Estados, 5001, Santa Terezinha, Santo André/SP, CEP 09210-580

Bloco L, 3ºAndar, Fone (11) 3356-7617

iniciacao@ufabc.edu.br

Projeto de Iniciação Científica
submetido para avaliação no
Edital: **04/2022**

Título do projeto: Central de Acompanhamento de Expressões Faciais.

Palavras-chave do projeto: Expressões faciais. Processamento multimídia. Rede de comunicação em sistemas embarcados.

Área do conhecimento do projeto: Engenharia de Informação

Sumário

1	Resumo	3
2	Introdução e Justificativa	4
3	Objetivos	4
4	Metodologia	5
4.1	Processamento de imagem	5
4.2	Interface gráfica	7
4.3	Arquitetura de rede	8
4.3.1	Rede de Área Local (LAN)	8
4.3.2	Comutador de rede	8
4.3.3	Rede Privada Virtual (VPN)	9
4.3.4	Vantagens e limitações entre os modos de conexão	10
5	Viabilidade	10
6	Cronograma de atividades	10
6.1	Estágios para o desenvolvimento do sistema	11
6.2	Planejamento de execução	11
	Referências	12

1 Resumo

As expressões faciais constituem uma grande faceta da comunicação não-verbal humana, e sua importância para a transmissão de informações incentivou pesquisas em reconhecimento de emoções, visando oferecer melhorias à qualidade de vida por meio do entendimento e processamento de dados relacionados à expressão humana. Nesse contexto, estudos recentes fazem amplo uso do Reconhecimento de Expressões Faciais - REF, para o levantamento de dados sobre a percepção humana de acordo com diferentes estímulos. Todavia, no contexto acadêmico, há pouca documentação acerca da aplicação do REF para o acompanhamento ao-vivo de expressões faciais, sobretudo de forma remota e centralizada. Assim, este projeto visa desenvolver e avaliar uma central para o acompanhamento remoto de expressões faciais e estados emotivos de pessoas em determinados espaços, tal como laboratórios, permitindo a análise de padrões de comportamento de grupos de indivíduos de forma centralizada e remota. Para isso, serão implementados modelos de REF em sistemas embarcados – como Raspberry Pi – para a detecção e transmissão de expressões faciais para um servidor WEB, compilando e hospedando os dados coletados em uma página WEB, a qual poderá ser acessada por meio de um navegador de internet, seja via um celular ou computador. Em paralelo, serão levantados dados a respeito da exequibilidade do processamento de imagens em sistemas embarcados, além de como diferentes formas de conexão e transmissão de dados impactam o sistema remoto. Desse modo, o desenvolvimento do sistema permitiria avaliar numericamente padrões de emoções de grupos de pessoas em dados ambientes, além de determinar limitações da execução de modelos de REF em sistemas embarcados.

Palavras-chaves: Expressões faciais. Processamento multimídia. Rede de comunicação em sistemas embarcados.

2 Introdução e Justificativa

As expressões faciais – tal como a felicidade, raiva e tristeza – constituem uma grande faceta da comunicação não-verbal humana, e sua importância para a transmissão de informações, independente da intensidade ou duração, é evidenciada em sua natureza de percepção e reação subconsciente [1].

Tendo isso em mente, o ramo de estudo da Computação Afetiva ganhou espaço em centros de pesquisa como o MIT Media Lab [2] visando, sobretudo, oferecer melhorias à qualidade de vida por meio do entendimento e processamento de dados relacionados à afeição humana.

Estudos recentes fazem uso do Reconhecimento de Expressões Faciais (REF) para o levantamento de reações humanas em frente a diferentes estímulos, tal como sonoros em centros urbanos [3], ou a percepção de stress de um motorista em diferentes ambientes [4].

Entretanto, apesar da crescente notoriedade da Computação Afetiva no contexto acadêmico, há um grande enfoque em aplicações de escopo local, mas pouca representatividade de sistemas remotos. Dessa forma, este projeto objetiva levantar dados e estudar a exequibilidade de REF em sistemas embarcados, sobretudo da perspectiva do acesso e acompanhamento remoto.

Ademais, este projeto é uma extensão de um projeto de pesquisa anterior próprio do proponente, no qual foi criado um sistema para o reconhecimento e transmissão de expressões faciais entre computadores em uma rede de área local – além de um estudo comparativo de diferentes métodos de pré-processamento e REF.

3 Objetivos

Este projeto visa desenvolver e avaliar uma central para o acompanhamento remoto de expressões faciais e estados emotivos de pessoas em determinados espaços, tal como laboratórios, por meio de câmeras em sistemas embarcados – permitindo a análise de padrões de comportamento de grupos de pessoas de forma centralizada e remota. Além disso, será feito um estudo do desempenho de diferentes métodos de transmissão de dados e conexão dos sistemas embarcados em rede.

O projeto abrange dois objetivos principais:

1. Implementar o sistema de Reconhecimento de Expressões Faciais em sistemas embarcados, criando uma central interativa de acompanhamento – com pontos de acesso remotos.
2. Realizar um estudo comparativo sobre o impacto de como diferentes formas de transmissão e processamento de dados afetam o desempenho do sistema como um todo – tal como a alteração de protocolos (TCP, UDP), grau de processamento local de dados (transmitir vídeo diretamente da câmera, enviar somente a imagem de rostos detectados, ou enviar as expressões faciais já processadas).

4 Metodologia

A princípio, visa-se criar uma central para o acompanhamento de expressões faciais e estados emotivos de fácil expansão, utilizando sistemas embarcados – tal como Raspberry Pi – como pontos de monitoramento (câmeras, integradas com REF local), enviando os dados coletados em tempo real para um Servidor WEB, que poderá ser acessado remotamente, por meio de navegadores de internet. A ordem dos processos está elucidada na Figura 1, a seguir:

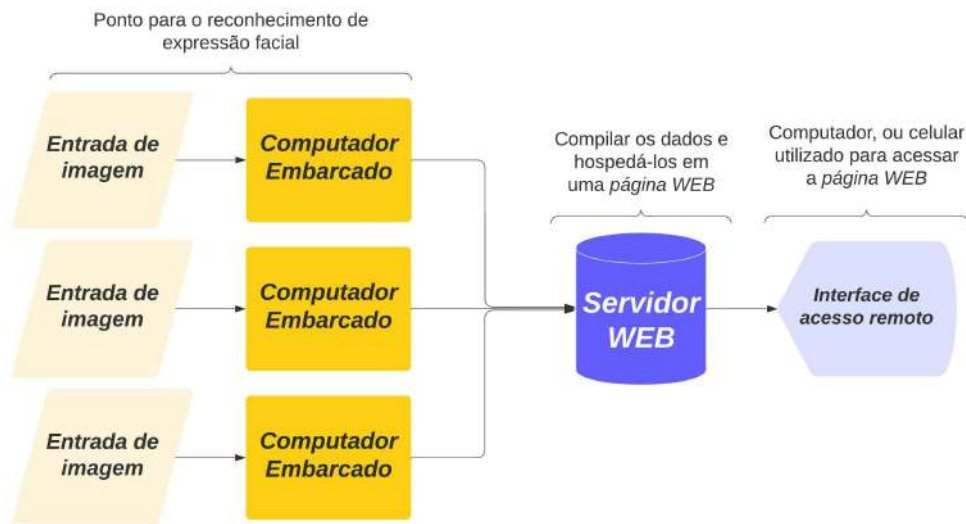


Figura 1: Fluxograma geral.

Fonte: Diagrama elaborado pelo proponente.

A Figura 1 apresenta a ordem em que ocorrem os processos do sistema. Primeiramente, a imagem é coletada a partir de uma câmera, e processada em um computador embarcado. Feito o processamento, os dados são transferidos para o Servidor WEB, onde serão compilados e exibidos em uma página WEB. Por fim, esta página pode ser acessada a partir de um celular ou computador.

A Metodologia foi dividida em três subseções: A subseção 4.1, sobre o processamento de imagens e extração de expressões faciais, a 4.2, detalhando a interface gráfica e a 4.3, quanto as diferentes formas de conexão em rede do sistema.

4.1 Processamento de imagem

A princípio, as imagens podem ser pré-processadas a fim de homogenizar a entrada de dados, contribuindo para uma melhor performance do sistema. Para isso, existem três etapas: detecção de rostos, alinhamento frontal dos rostos detectados e – em casos específicos de alta ou baixa exposição à luz – pode ser utilizado a equalização de histograma.

Segue a Figura 2, a fim de elucidar as etapas do pré-processamento:

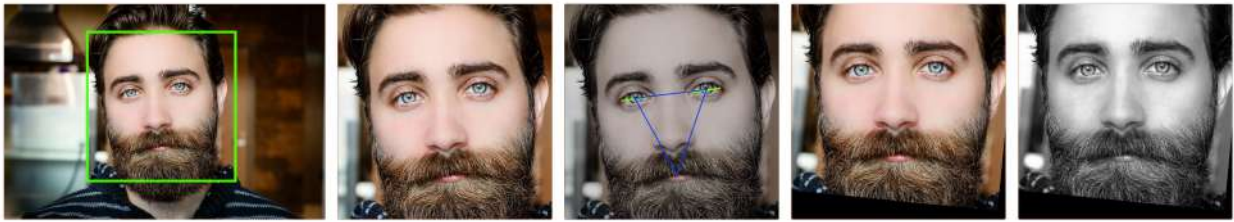


Figura 2: Estágios de pré-processamento.

Fonte: Diagrama elaborado pelo proponente, imagem original livremente licenciada [5].

As imagens da Figura 2 representam, sequencialmente: Detecção facial, recorte da área de interesse, detecção de pontos de interesse (características faciais), alinhamento frontal da face, equalização do histograma.

Com esse intuito, podem ser utilizados os seguintes métodos para o pré-processamento:

- Detecção de rostos – poderá ser feito via: OpenCV [6], MTCNN [7] ou Retina Face [8];
- Detecção de pontos de interesse (olhos e boca) – poderá ser feito via: DLIB [9], OpenCV, MTCNN, RetinaFace;
- Equalização de histograma – poderá ser feito via: OpenCV [10].

Os métodos citados consistem de:

- OpenCV [11]: Uma biblioteca licenciada em código aberto – presente em C++, Python, Java e MATLAB – voltada para a implementação de sistemas de visão computacional e aprendizado e máquina;
- RetinaFace: Modelo para detecção simultânea de rostos e pontos de interesse (olhos e boca), de imagens em ambientes não controlados – a partir de uma etapa detecção.
- MTCNN (Multi-task Cascaded Convolutional Networks): Modelo para detecção simultânea de rostos e pontos de interesse, de imagens em ambientes não controlados – a partir de caixas de regressão em cascata;
- DLIB [12]: Uma "caixa de ferramentas" (toolkit) licenciada em código aberto – disponível para uso em C++ e Python – contendo, sobretudo, modelos de aprendizado de máquina e algoritmos numéricos.

Feito isso, podem ser utilizados os seguintes modelos com a finalidade de extrair a expressão facial da imagem já pré-processada: Residual Masking Network [13], FER (Py-feat) [14] ou SVM (Py-feat) [14].

- Residual Masking Network: é baseado em uma Rede Residual a partir de blocos de Mascaramento. O modelo é fornecido pelos autores pré-treinado nas bases de dados: *VEMO* (dos autores) e *Fer2013* [15].
- FER (Py-feat): Modelo pré-treinado em Rede neural convolucional, disponível por Py-feat.

- SVM (Py-feat): Modelo pré-treinado em Histograma de Gradientes Orientados, disponível por Py-feat.

Dito isso, um projeto de pesquisa anterior do proponente, foram comparados vários modelos de REF e pré-processamento de imagens – sobretudo os previamente citados. Desse modo, o software já desenvolvido será ajustado para as novas condições de detecção.

Ademais, os diferentes modelos para detecção citados apresentam necessidades de processamento diferentes, e serão utilizados a depender da capacidade de hardware disponível. É válido mencionar que, a depender dos requisitos mínimos e andamento geral, o processamento de imagens pode ser integralmente direcionado para o servidor WEB, restando ao sistema embarcado transmitir o feed de vídeo.

4.2 Interface gráfica

Feito o processamento das imagens e roteamento de dados, basta exibir as expressões faciais detectadas em uma interface gráfica.

É necessário que os gráficos sejam atualizados automaticamente, seja de acordo com a entrada de novos dados, ou em um temporizador – permitindo a visualização em tempo real das entradas.

No momento, é planejada a exibição em dois gráficos, um em disco, comparando os valores de forma relativa (porcentagem), e um em barra – comparando os valores de forma absoluta (numericamente). Os gráficos podem ser traçados em Python, a partir da biblioteca Matplotlib [16], ou em Java, a partir da biblioteca XChart [17].

A Figura 3, a seguir, contém um esboço da interface para visualização de dados.



Figura 3: Esboço da interface gráfica.
Fonte: Imagem elaborada pelo proponente.

O esboço da interface gráfica, na Figura 3, contém os dois gráficos planejados – em forma de disco e barra – para a avaliação dos dados em tempo real. O gráfico em disco é útil para comparar a proporção dos dados entre si, enquanto o gráfico em barra permite a avaliação numérica dos dados.

4.3 Arquitetura de rede

Após o reconhecimento de expressões faciais, estas informações devem ser transferidas ao Servidor WEB. A conexão entre os sistemas embarcados e o servidor WEB poderá ser feita através de soquetes – utilizando as linguagens de programação Python ou Java como base para a lógica.

Feito isso, o servidor deverá compilar os dados e exibi-los em uma interface gráfica. A qual poderá ser acessada da mesma rede, ou de uma rede diferente.

Para acessar da mesma rede local, basta buscar o IP da página WEB do Servidor em um navegador – todavia, para acessar remotamente, é necessário acessá-lo a partir de uma rede comutada, ou de um túnel VPN (Rede privada virtual).

4.3.1 Rede de Área Local (LAN)

Para o transporte de dados em uma Rede de Área Local, pode ser utilizado inicialmente um Switch e, conforme necessário, pode-se implementar um roteador de internet, assim como na Figura 4.

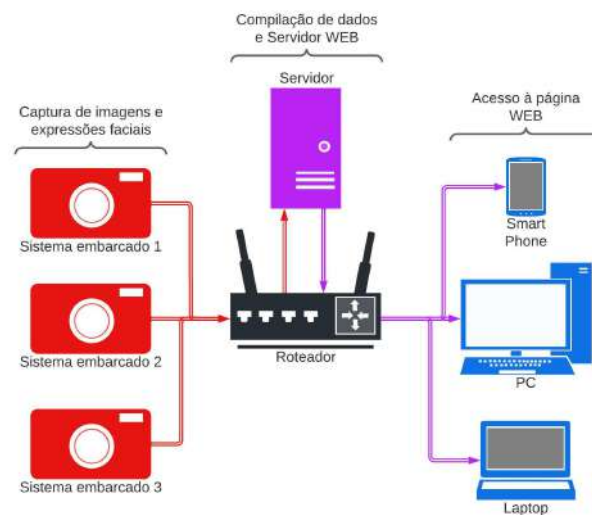


Figura 4: Conexão do sistema em uma Rede de Área Local (LAN).

Fonte: Diagrama elaborado pelo proponente.

Na Figura 4, todos os equipamentos estão conectados a uma mesma rede Ethernet, representada pelo roteador. Nota-se que o fluxo de dados é feito em uma única rede.

4.3.2 Comutador de rede

Além disso, poderão ser feitos testes acessando o servidor de diferentes redes, como por meio de um Comutador de rede, como demonstrado na Figura 5, conectando as duas sub-redes diretamente.

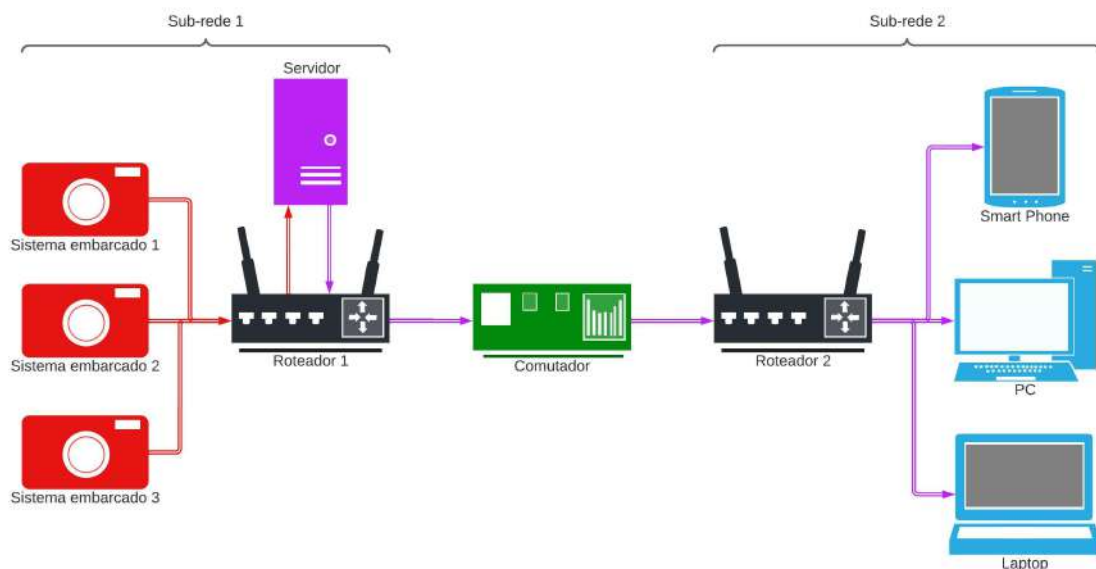


Figura 5: Conexão de sub-redes a partir de um comutador.

Fonte: Diagrama elaborado pelo proponente.

Neste caso, na Figura 5, é evidente a presença de duas sub-redes, uma responsável pelo roteamento de dados entre os sistemas embarcados e o servidor – e outra pela comunicação entre o servidor e o ponto de acesso (computador, celular).

4.3.3 Rede Privada Virtual (VPN)

A conexão remota poderá ser feita também via um túnel de Rede Privada Virtual (Virtual Private Network, ou como popularmente conhecida, VPN), representado na Figura 6 – conectando os computadores individualmente ao servidor WEB, perante avaliação manual.

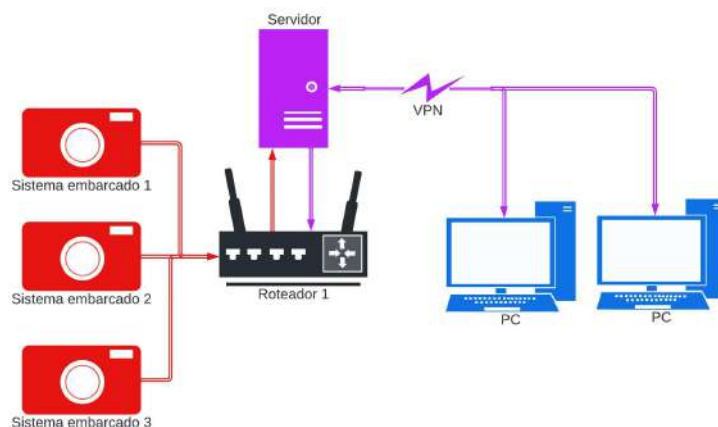


Figura 6: Conexão de pontos de acesso via VPN.

Fonte: Diagrama elaborado pelo proponente.

Na Figura 6, há uma rede designada para a comunicação entre os sistemas embarcados e o servidor – mas a comunicação entre o servidor e o ponto de acesso é feita a partir de um túnel VPN.

4.3.4 Vantagens e limitações entre os modos de conexão

A princípio, o modo de conexão mais simples consiste em acessar a página WEB diretamente da mesma rede local, porém isso só é possível quando o usuário está fisicamente próximo do roteador.

Do ponto de vista do acesso remoto, é evidente que o uso de VPNs manualmente liberadas é mais seguro, porém apresenta uma maior dificuldade de escalabilidade para o projeto. Em contraponto, a implementação de um Comutador é escalável, mas exige que sejam alocados mais recursos para a segurança e controle de acesso ao servidor.

Ainda mais, será possível testar a conexão remota entre os sistemas usando dois laboratórios distantes, ambos pertencentes à Universidade Federal do ABC: um no Campus de Santo André, e outro no Campus de São Bernardo do Campo.

5 Viabilidade

O proponente tem experiência prévia com linguagens de programação – dentre elas Python, CLP (LD, SFC) e Java. Além de experiência com implementação de modelos de Reconhecimento de expressões faciais, processamento de imagens, e lógica computacional para servidores.

Além disso, há códigos para o pré-processamento de imagens e reconhecimento de expressões faciais que já foram desenvolvidos em um projeto anterior do proponente, e poderão ser parcialmente reaproveitados.

Não será necessária a aquisição de novos materiais, uma vez que o aluno já possui os componentes necessários para a execução do projeto, no caso: Raspberry Pi, Arduino, Computadores, Switch, Roteador, Cabos de rede, Câmera e Webcam.

Caso surja a necessidade de executar a proposta em home-office, é possível realizar o desenvolvimento de sistemas e montagem dos componentes à distância, restando apenas testar os sistemas nos laboratórios.

Por fim, para o desenvolvimento de estudos comparativos e levantamento de dados, serão utilizadas bases de dados disponíveis para uso acadêmico, já estudadas e utilizadas em um projeto anterior do proponente – perante a permissão de seus respectivos autores. Dessa forma, não será necessária a aprovação da comissão de ética em pesquisa.

6 Cronograma de atividades

O Cronograma de atividades foi dividido em duas subseções: Uma contendo tópicos relacionados aos principais pontos de cada etapa do projeto, e outro contendo a Tabela 1 – delineando a ordem e o tempo previsto para o desenvolvimento de cada etapa.

6.1 Estágios para o desenvolvimento do sistema

1. Sistemas embarcados;
 - (a) Instalação de Webcam em Raspberry PI;
 - (b) Detecção de vídeo;
 - i. Detecção facial;
 - ii. Reconhecimento de expressões faciais;
 - (c) Transferência de dados para computador.
2. Comunicação em rede;
 - (a) Implementação de rede de área local;
 - (b) Configuração de soquetes;
 - i. Aprovação de conexões;
 - ii. Teste e correção de erros de transporte;
 - (c) Compilação de dados.
3. Interface de acompanhamento;
 - (a) Lógica de acesso e permissões;
 - (b) Listagem de dados ao vivo;
 - (c) Gráficos em barra e circular;
4. Teste de sistema e levantamento de dados.

6.2 Planejamento de execução

As etapas citadas anteriormente, na seção 6.1, foram categorizadas de acordo com o tempo previsto necessário para sua realização ao longo dos dez meses de execução do projeto, e distribuídas ao longo de sua duração, na Tabela 1, a seguir:

Etapa	Mês											
	1	2	3	4	5	6	7	8	9	10	11	12
1.a	X											
1.b		X										
1.c			X									
2.a				X								
2.b					X							
2.c						X						
3.a							X					
3.b								X				
3.c									X			
4										X	X	X

Tabela 1: Cronograma de atividades.

Outrossim, é intencionado que o sistema central esteja funcional até o décimo mês de desenvolvimento, para que hajam três meses disponíveis para corrigir possíveis erros de código ou de arquitetura.

Referências

- [1] DIMBERG, U.; THUNBERG, M.; ELMEHED, K. Unconscious facial reactions to emotional facial expressions. *Psychological Science*, v. 11, p. 86–89, 01 2000. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/11228851/>>. Acesso em: 2022-06-17.
- [2] MIT MEDIA LAB. *Group Overview < Affective Computing – MIT Media Lab*. 2019. Disponível em: <<https://www.media.mit.edu/groups/affective-computing/overview/>>. Acesso em: 2022-06-17.
- [3] MENG, Q. et al. On the effectiveness of facial expression recognition for evaluation of urban sound perception. *Science of The Total Environment*, v. 710, p. 135484, 03 2020. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0048969719354774>>. Acesso em: 2022-06-17.
- [4] BUSTOS, C. et al. Predicting driver self-reported stress by analyzing the road scene. *arXiv:2109.13225 [cs]*, 09 2021. Disponível em: <<https://arxiv.org/abs/2109.13225>>. Acesso em: 2022-06-17.
- [5] FREE Image on Pixabay - Beard, Face, Man, Model, Mustache. 2016. Disponível em: <<https://pixabay.com/photos/beard-face-man-model-mustache-1845166/>>. Acesso em: 2021-09-19.
- [6] OpenCV: Cascade Classifier. Disponível em: <https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html/>. Acesso em: 2022-04-22.
- [7] ZHANG, K. et al. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, v. 23, n. 10, p. 1499–1503, 2016.
- [8] DENG, J. et al. Retinaface: Single-stage dense face localisation in the wild. *arXiv.org*, 2019. Disponível em: <<https://arxiv.org/abs/1905.00641>>. Acesso em: 2022-06-25.
- [9] DLIB - Detector facial em cinco pontos. Disponível em: <http://dlib.net/face_alignment.py.html>. Acesso em: 2021-09-10.
- [10] OpenCV: Histograms - 2: Histogram Equalization. Disponível em: <https://docs.opencv.org/master/d5/daf/tutorial_py_histogram_equalization.html>. Acesso em: 2021-07-21.
- [11] OPENCV. *About - OpenCV*. 2020. Disponível em: <<https://opencv.org/about/>>. Acesso em: 2022-06-25.
- [12] DLIB.NET. *dlib C++ Library - Introduction*. 2015. Disponível em: <<http://dlib.net/intro.html>>. Acesso em: 2022-06-25.
- [13] PHAM, L.; VU, T. H.; TRAN, T. A. Facial expression recognition using residual masking network. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2021. p. 4513–4519. ISSN 1051-4651.
- [14] CHEONG, J. H. et al. Py-feat: Python facial expression analysis toolbox. *arXiv preprint arXiv:2104.03509*, 2021.
- [15] SAMBARE, M. *FER-2013*. 2013. Disponível em: <<https://www.kaggle.com/msambare/fer2013>>. Acesso em: 2022-05-18.

- [16] MATPLOTLIB.ORG. *Plot types — Matplotlib 3.5.2 documentation*. 2012. Disponível em: <https://matplotlib.org/stable/plot_types/index.html>. Acesso em: 2022-06-26.
- [17] KNOWM. *knowm/XChart: XChart is a light-weight Java library for plotting data*. 2022. Disponível em: <<https://github.com/knowm/XChart>>. Acesso em: 2022-06-26.