Universidade de Brasília
Faculdade de Tecnologia

Reconhecimento de Emoções a Partir de Expressões Faciais pelo Robô NAO

Eric do Vale de Castro

Gabriel Guimarães Almeida de Castro

TRABALHO DE GRADUAÇÃO

ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília

2023

Universidade de Brasília
Faculdade de Tecnologia

Reconhecimento de Emoções a Partir de Expressões Faciais pelo Robô NAO

Eric do Vale de Castro
Gabriel Guimarães Almeida de Castro

Trabalho de Graduação submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Orientador: Prof. Dr. Roberto de Souza Baptista

Brasília
2023

Castro, Eric do Vale de.
C355r Reconhecimento de Emoções a Partir de Expressões Faciais pelo Robô NAO / Eric do Vale de Castro; Gabriel Guimarães Almeida de Castro; orientador Roberto de Souza Baptista. –– Brasília, 2023.
86 p.

Trabalho de Graduação em Engenharia de Controle e Automação –– Universidade de Brasília, 2023.

1. Expressão Facial. 2. Robótica. 3. Rede Neural Convolucional. 4. Processamento de Imagem. I. Castro, Gabriel Guimarães Almeida de. II. Baptista, Roberto de Souza, orient. III. Título

Universidade de Brasília
Faculdade de Tecnologia

Reconhecimento de Emoções a Partir de Expressões Faciais pelo Robô NAO

Eric do Vale de Castro
Gabriel Guimarães Almeida de Castro

Trabalho de Graduação submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Trabalho aprovado. Brasília, 15 de Fevereiro de 2023:

Prof. Dr. Roberto de Souza Baptista, UnB/FGA
Orientador

Prof. Dr. Carla Maria Chagas E. Cavalcante Koike, UnB/FT/CIC
Examinador interno

Prof. Dr. Wânia Cristina De Souza, UnB/IP
Examinador interno

Brasília
2023

Este trabalho é dedicado a todas as pessoas que sofrem de depressão, que precisam de apoio, auxilio e de uma mão amiga para poder se levantarem desta doença tão terrível.

Eric do Vale de Castro

Este trabalho é dedicado as pessoas que sonham em alcançar objetivos que parecem impossíveis devido à distância entre o sonhado e a realidade. Lembre-se, tudo é possível, basta ter coragem e perseverança para alcançar o que deseja.

Gabriel Guimarães Almeida de Castro

Agradecimentos

Agradeço primeiramente aos meus pais Vicente e Maria Santa que sempre me ensinaram a ser forte e persistente, pois na vida sempre iria aparecer muitas dificuldades, mas ao se manter forte elas seriam ultrapassadas. Com todas as complicações eles sempre estiveram lá para me apoiar e me ajudar com o quer que fosse. Ao meu irmão Evandro que apesar de todas as chaturas me ensinou que com dedicação tudo é possível, mesmo com todos os problemas e dificuldades.

Em seguida, agradeço ao maior presente que a Universidade de Brasília(UnB) poderia me dar, o meu grande amor Ayalla. Você é minha a melhor amiga, parceira, companheira, namorada, noiva e futura esposa, sempre que precisei você estava lá meu amor. Você me apoiou, me tirou muitas risadas e até me fez perder o meu cabelo, mas você meu amor sempre esteve lá. Você, meu amor, sua mãe e sua irmã são grandes tesouros que poderiam me proporcionar.

Após isso, aos meu amigos da faculdade Gabriel, Luiza e a Stephanie, pois a UnB se mostrou ser extremamente árdua, mas nós três juntos estávamos lá para um cuidar do outro nos incontáveis trabalhos noturnos, horas e horas de estudos para não irmos tão bem assim no final, mas mesmo depois disso tudo lutamos e vencemos juntos. E sim vencemos todas as barreiras pela frente.

O próximo é o laboratório LARA e as pessoas que conheci dele: Marina, Raphael Braccialli, Lucas Guilherme, Bruno, Miguel e as outras pessoas que estavam para me ajudar, me dar conselhos e mesmo com o mundo pegando fogo, tínhamos bolo e muita procrastinação para retirarmos muitas risadas e mostrar que o mundo é um lugar melhor.

E, ao melhor projeto da UnB, a UnBeatables e as pessoas maravilhosas como a Débora, Nátalia, Lívia, Paulo, Guilherme e outras pessoas que tivemos excelentes viajes juntos, muitas noites em claro para colocar o robô para funcionar e jogar futebol, passeios na praia e as saídas para comer com muitas risadas e fofocas.

A seguir, no final da graduação apareceu duas preciosidades, a ansiedade em pessoa Rosana, que quando começamos a trabalhar juntos logo vimos que dávamos certinho, não só do trabalho, mas também de reclamar do trabalho, e conversar sobre os presentes que iriamos dar aos nossos parceiros. E, da mega tranquilidade do Igor Beduin, que mesmo com ódio no olhar, mostrava toda aquela tranquilidade maravilhosa. Aí juntou nós três, o trio maravilha, fazendo o casamento perfeito, que adoraria juntar novamente no futuro.

E, por fim, ao meu orientador Roberto Baptista, que não só me deu conselhos para ser melhor como estudante, mas também conselhos para a vida. E, irei sentir saudades de

você me odiar.

Eric do Vale de Castro

Eu gostaria de começar agradecendo a Deus, pela força que me deu para continuar nesta jornada apesar das dificuldades. Em seguida, quero agradecer aos meus pais, Edson e Sueidy, por terem sempre me incentivado a buscar o conhecimento e a fazer tudo com dedicação. Também gostaria de agradecer aos meus avós paternos e maternos pelo apoio incondicional.

Gostaria de expressar minha gratidão aos meus amigos, Denise, Reinaldo, João Pedro, Arthur e Lucas, que se tornaram como família para mim ao chegar em Brasília para começar esta jornada.

E aos meus amigos de faculdade, Luíza, Eric e Stephanie, que caminharam junto comigo nesta jornada da Engenharia Mecatrônica. Pois este apoio, amizade e incentivo foram muito importantes para que eu pudesse continuar e vencer todas as barreiras.

Agradeço ao Laboratório de Robótica e Automação (LARA), foi nele que conheci pessoas incríveis como os professores que me orientaram em projetos e os alunos que estavam sempre dispostos a ajudar.

Agradeço também ao meu orientador Roberto Baptista, por ter aceitado este desafio e por ter sido uma grande fonte de apoio e orientação.

Enfim, gostaria de agradecer a todos vocês que estiveram presentes em minha vida e que contribuíram para a realização deste trabalho, vocês ajudaram a me tornar a pessoa que sou hoje. Obrigado a todos!

Gabriel Guimarães Almeida de Castro

“Leva muito tempo para perceber que
a felicidade e a infelicidade dependem de você,
porque é muito confortável para o ego achar
que os outros estão fazendo você infeliz.”
(Osho)

“Os sorrisos são provavelmente
as expressões faciais mais subestimadas,
muito mais complicadas do que a maioria das pessoas pensam.
Há dezenas de sorrisos, cada um diferente do outro
em aparência e na mensagem que expressam.”
(Paul Ekman)

Resumo

As expressões faciais são muito importantes para a comunicação não verbal, e assim, como o ditado popular: "Um gesto fala mais do que mil palavras". Baseado neste dito seria possível criar uma rede neural convolucional capaz de de classificar algumas emoções a partir da face dos usuários, sendo elas: Raiva, Nojo, Medo, Felicidade, Neutralidade, tristeza e Surpresa. Com isso, o classificador sendo aplicado na plataforma robótica *NAO* adiciona uma nova capacidade para realizar outras aplicações com o mesmo. O trabalho apresentado neste manuscrito terá uma metodologia de criar um classificador de emoções com a algumas estratégias desde uma rede convolucional de 4 camadas e do *MobileNet*, e todas elas serão testadas de forma funcional na plataforma robótica. A melhor estratégia se apresentou ser o do *MobileNet*, que com o uma média 66% de precisão baseado no banco de imagens de validação. E, esta rede aplicada no próprio robô teve resultados iguais a do treinamento, como a neutralidade, tristeza e alegria teve em média 81% de acerto. Apesar de não apresentar uma taxa nada satisfatória em relação a emoção de nojo, porém, para o restante das emoções são possíveis de serem trabalhadas e analisadas para um relatório de emoções numa sessão de terapia.

Palavras-chave: Expressão Facial. Robótica. Rede Neural Convolucional. Processamento de Imagem.

Abstract

Facial expressions are very important for nonverbal communication, and as the popular saying goes: "A gesture speaks louder than a thousand words". Based on this saying, it would be possible to create a convolutional neural network capable of classifying some emotions from the users' faces, namely: Anger, Disgust, Fear, Happiness, Neutrality, Sadness, and Surprise. With this, the classifier applied to the *NAO* robotic platform adds a new ability to perform other applications with it. The work presented in this manuscript will have a methodology of creating an emotion classifier with some strategies including a 4 layers convolutional network and the *MobileNet*, and all functionally tested on the robotic platform. The best strategy was using the *MobileNet*, with an average of 66% accuracy based on the validation image database. And, this network applied to the robot itself showed results equal to those of the training, such as neutrality, sadness, and joy, with an average accuracy of 81%. Despite not presenting a satisfactory rate regarding the emotion of disgust, the rest of the emotions can be worked on and analyzed for an emotion report in a therapy session.

Keywords: Facial Expression. Robotic. Convolution neural network. Image Processing.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Com o avanço da tecnologia de inteligência artificial e o aumento da capacidade computacional, os robôs estão adquirindo habilidades antes exclusivas aos seres humanos. Atualmente, robôs humanóides são capazes de caminhar em ambientes diversos, detectar objetos, transportar cargas, responder a comandos de voz e muito mais. Além disso, a robótica também tem aplicações para integrar crianças Transtorno do Espectro Autista(TEA) na sociedade como pode ser visto no trabalho (SANTOS, 2019).

Neste contexto, o presente trabalho tem como objetivo treinar uma rede neural para reconhecer emoções baseadas em expressões faciais e integrar essa habilidade a um robô, expandindo ainda mais as suas capacidades.

1.1 Contextualização

Ao longo dos anos, a pesquisa em reconhecimento de emoções tem se tornado cada vez mais relevante devido à sua aplicação em diversas áreas, como robótica, marketing e saúde. Sendo assim, a padronização das expressões faciais e a criação de bancos de dados permitem com que robôs consigam classificar as emoções através de inteligência artificial.

Portanto, será necessário entender que a partir de uma expressão de uma emoção, o ser humano é capaz de se comunicar, ou seja, a expressão facial tem a capacidade de transmitir uma informação. Desta forma, a comunicação é separada em duas modalidades: a comunicação verbal e não verbal.

O processo de comunicação verbal refere-se transmissão de informação entre duas ou mais pessoas, que se incluí desde a palavras ditas, mensagens escritas e linguagem de sinais.

A comunicação não verbal refere-se à transmissão de significado ou sentimentos sem o uso de palavras escritas ou faladas. Ou seja, é a comunicação entre pessoas que se utiliza de:

- Movimento das mãos;
- Linguagem corporal;
- Posturas;
- Gestos;
- **Expressões Faciais**.

Sendo assim, o principal recurso que será avaliado neste trabalho serão as expressões faciais, pois elas tem a capacidade de apresentar padrões que podem ser aprendidos por inteligência artificial, que por sua vez será capaz de prever rapidamente uma emoção o usuário expressa.

E, como demonstrado na figura 1, a bancada de trabalho é composta pela plataforma robótica *NAO*, que será manipulado remotamente a partir de um computador externo. A imagem mostra a execução do projeto final em que o robô a partir das câmeras, captura as imagens do usuário, envia a um classificador treinado, e informa através de fala a emoção detectada. E, por fim, é há uma luminária com o propósito de melhorar a nitidez das imagens.

Figura 1 – Banca de Trabalho

Este projeto poderia ser útil em processos de diagnósticos de problemas mentais mais rapidamente, no auxílio de crianças com Transtorno do Espectro Autista(TEA), no apoio de profissionais de psicologia e psiquiátricos para tratamento de pacientes e entre outras finalidades.

A utilização da robótica como abordagem terapêutica para crianças com Transtorno do Espectro Autista (TEA) não apenas proporciona maior acessibilidade educacional e social, como também favorece o desenvolvimento da socialização dessas crianças desde o momento do diagnóstico. Por meio da implementação de ferramentas de reconhecimento de emoções, é possível empregar uma estratégia de avaliação dos momentos em que a criança se expressa, permitindo a identificação das experiências que lhe proporcionam maior satisfação, bem como aquelas que geram desconforto, como raiva ou aversão. Além disto, há projetos onde as crianças são treinadas a se expressarem melhor.

1.2 Objetivo

O objetivo deste trabalho é investigar, de maneira didática, a possibilidade de utilizar técnicas de inteligência artificial para reconhecer emoções através de expressões faciais. E, serão exploradas técnicas de processamento de imagens e modelos de rede neural para tentar alcançar um bom nível de precisão no reconhecimento de emoções expressas pelo usuário em frente da plataforma robótica *NAO*.

2 Fundamentação Teórica

Neste capítulo, será exposta alguns tópicos teóricos imprescindíveis para a compreensão global da metodologia a ser apresentada no capítulo 3. Para sua elaboração, foram utilizadas referências bibliográficas provenientes do meio acadêmico.

2.1 Expressão Facial

2.1.1 Comunicação Não Verbal

Em todo processo de comunicação cerca de 93% das interações sociais se tratam de comunicação não verbal (RAMOS; BORTAGARAI, 2012), que pode acontecer por meio de uma piscada de olho, ao manter as sobrancelhas cerradas, através de uma abertura dos lábios por uma curva suave em seus cantos e etc; todas estas expressões são exemplos de comunicação não verbal, que se trata exclusivamente de uma transmissão de significado sem o uso da linguagem falada ou escrita.

Como diz o ditado popular: "um Gesto fala mais do que mil palavras". Pressupõe-se que, é demasiado exagero que tal expressão possa se referir a um simples e único gesto; contudo, esta área de conhecimento vem se tornando um vasto campo de estudos. Pois, dentro de gestos, posturas, expressões, contato visual, proximidade física, entre outros gestos não-verbais, pode-se apresentar uma mensagem cultural ou emocional, e ainda pode-se reconhecer problemas como deficiências, doenças ou transtornos mentais.

2.1.2 Emoções Humanas

Antes mesmos de entrar no conceito de expressão facial é necessário levar em consideração a seguinte pergunta: como as emoções são comunicadas e interpretadas? A comunicação humana não se limita apenas a palavras, ou seja, envolve gestos e expressões faciais e ao longo dos anos os seres humanos aprenderam a identificar padrões que os ajudem a interpretar o que cada expressão está comunicando.

Ao olhar o dicionário do português, entende-se que o significado de emoção é uma reação moral, psíquica ou física, geralmente causada por uma confusão de sentimentos (DICIO, DICIONÁRIO ONLINE DE PORTUGUÊS, 2023). Apesar de apresentar uma definição específica, o termo vai além do seu significado, pois a emoção também dependerá de uma série de fatores como meios culturais e características pessoais.

Na área da psicologia dentro da neurociência, emoção é um universo com várias teorias de como elas surgem, e há muitas formas diferentes de serem interpretadas a variar

de cada cultura, como por exemplo na cultura brasileira onde as pessoas tem intimidade entre outras pessoas por meio de toques interpessoais, e em contraste, a cultura japonesa é completamente oposta, com uma certa contenção emocional. Algumas culturas também podem diferenciar em termos de normas sociais para quais são aceitáveis de se expressar em diferentes contextos.

A face é, de fato, a parte do corpo humano mais visível no contato social, e é incrível como ela é capaz de exibir mais de dez mil expressões (FREITAS-MAGALHÃES, 2013). Desta forma, a face pode expor emoções muito contundentes, e as principais delas são alegria, tristeza, raiva, nojo e medo. Assim, a face é extremamente importante como canal de comunicação.

Enfim, as análises de expressões faciais e de linguagem corporal vem ganhando bastante força, tanto no ambiente acadêmico quanto no ambiente externo, pois é notório canais como o Metaforando (SANTOS, V., 2016) no *Youtube* que tem cerca de 5.59 milhões(dados verificados no dia 31/01/2023) de seguidores buscando conhecer esta área tão curiosa. E tudo isso partiu de *Charles Darwin* e foi confirmado pelo pesquisador americano *Paul Ekman* que também criou novas teorias da área.

Este trabalho analisará algumas expressões que remetem a emoções básicas e inatas ligadas ao instinto e sobrevivência. Elas são:

- **Raiva**
- **Nojo**
- **Medo**
- **Alegria**
- **Neutralidade**
- **Tristeza**
- **Surpresa**

Apesar de tudo, foi incluso a neutralidade para representar a falta de expressão, pois isto é importante em termos de estudos.

2.1.2.1 Raiva

A raiva é bem representada pela identificação das sobrancelhas cerrados, testa comprimida, lábios contraídos, a descoloração avermelhada da pele, ocasionada pelo aumento da pressão sanguínea, além de aumento do tom de voz, conversas muito aceleradas, respiração

rápida e transpiração (SANTOS, F. A. D., 2017). Estas características estão presentes na maioria das pessoas ao demonstrar esta emoção.

Figura 2 – Exemplo 1 da Expressão de Raiva

Fonte: *AffectNet* (MAHOOR, 2011)

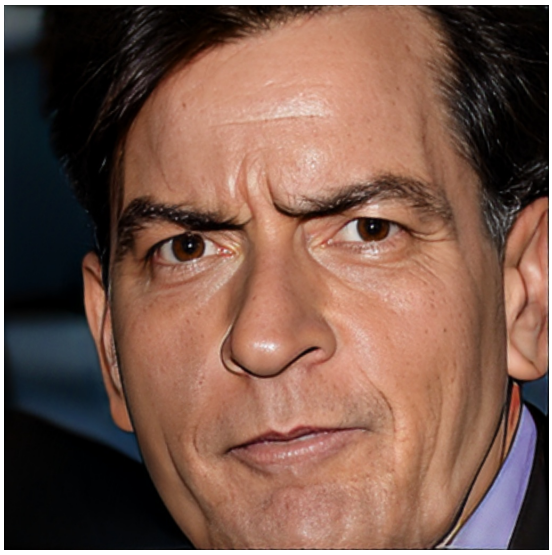

Figura 3 – Exemplo 2 da Expressão de Raiva

Fonte: *AffectNet* (MAHOOR, 2011)

2.1.2.2 Nojo

Esta é uma das emoções mais complexas de se identificar, e vários estudos mostram que para atuar de forma precisa e mostrar uma expressão de nojo, é necessário que a situação seja real e que haja realmente algo para causar nojo no ator. Então, devido a isto, os resultados obtidos ao aplicar o reconhecimento de expressões utilizando o robô podem não ser tão precisos, mas a finalidade destes testes é apenas mostrar uma aplicação. Sendo assim, os testes que realmente tem a finalidade de medir precisão são os resultados obtidos nos gráficos de acurácia e nas matrizes de confusão, pois nelas são utilizadas imagens do banco de validação.

Ao verificar a emoção de nojo percebe-se enrugamentos na base do nariz, fechamento dos olhos com enrugamento da parte inferior e do levantamento das narinas ocasionando a subida do lábio superior (SANTOS, F. A. D., 2017).

Figura 4 – Exemplo 1 da Expressão de Nojo

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 5 – Exemplo 2 da Expressão de Nojo

Fonte: *AffectNet* (MAHOOR, 2011)

2.1.2.3 Medo

As características que o medo têm é muito relacionado a perda ou dano, apresentando sobrancelhas bem levantadas, rugas de lado a lado na testa, pálpebra superior bem levantada com bastante tensão, boca em formato retangular e rugas bem presentes no queixo (SANTOS, F. A. D., 2017).

Figura 6 – Exemplo 1 da Expressão de Medo

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 7 – Exemplo 2 da Expressão de Medo

Fonte: *AffectNet* (MAHOOR, 2011)

2.1.2.4 Alegria

Agora ao observar a demonstração de alegria, percebe-se que há uma leve contração dos músculos ao redor dos olhos, enchimento das bochechas devido ao levantamento dos

lábios superiores e uma abertura bem prolongada da boca com o aparecimento dos dentes muitas vezes (SANTOS, F. A. D., 2017).

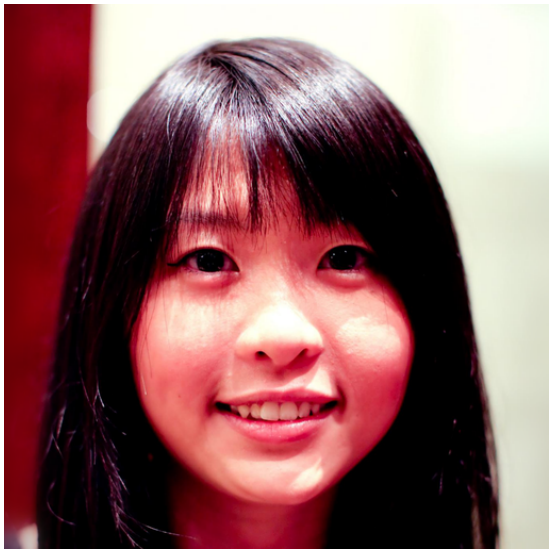

Figura 8 – Exemplo 1 da Expressão de Alegria

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 9 – Exemplo 2 da Expressão de Alegria

Fonte: *AffectNet* (MAHOOR, 2011)

2.1.2.5 Neutralidade

A neutralidade não é um tipo de emoção, mas há momentos onde não há expressões na face de uma pessoa, então é necessário que haja esta categoria para fins de estudo. Sendo assim, a neutralidade é quando a face não apresenta quase nenhuma contração muscular, boca fechada, olhos e pálpebras sem contração e apenas um olhar fixo a frente.

Figura 10 – Exemplo 1 da Expressão de Neutro

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 11 – Exemplo 2 da Expressão de Neutro

Fonte: *AffectNet* (MAHOOR, 2011)

2.1.2.6 Tristeza

Tem-se então a expressão de tristeza, ela é uma marca essencial do ser humano, e é demonstrada com rugas no centro da testa, sobrancelhas levantadas na parte mais interna e com muita tensão e encurvamento dos lábios para baixo (SANTOS, F. A. D., 2017).

Figura 12 – Exemplo 1 da Expressão de Triste

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 13 – Exemplo 2 da Expressão de Triste

Fonte: *AffectNet* (MAHOOR, 2011)

2.1.2.7 Surpresa

Por último, a emoção de surpresa é bem característica pela testa contraída com rugas ou sem, pálpebras superior e inferior bem abertos abrindo os olhos com simetria e com a boca aberta de forma oval (SANTOS, F. A. D., 2017).

Figura 14 – Exemplo 1 da Expressão de Surpresa

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 15 – Exemplo 2 da Expressão de Surpresa

Fonte: *AffectNet* (MAHOOR, 2011)

Com todos esses atributos, apesar do grande treinamento dos profissionais necessários em uma sessão de terapia, o fluxo humano de emoções é imenso, então ter uma ferramenta que os auxilie criando um relatório que mostra as emoções do paciente em relação ao tempo é algo muito interessante. E assim, será mais fácil fornecer um diagnóstico e tratamentos mais adequados. Enfim, é completamente possível criar um robô capacidade de oferecer um bom relatório de emoções, e de forma impessoal, sem interferência das emoções do profissional e livre de qualquer tipo de preconceito ou julgamentos.

2.2 O Robô NAO

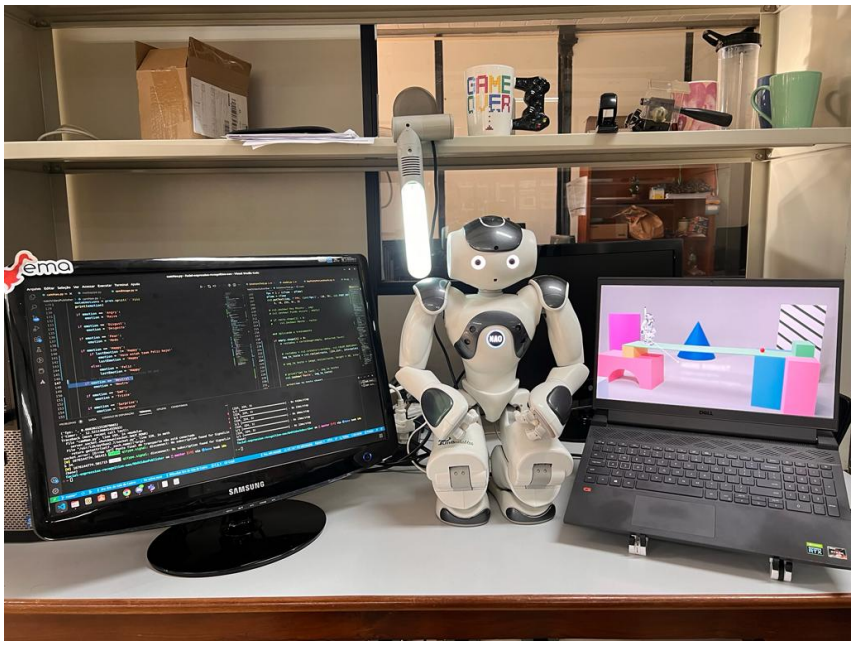

Figura 16 – O Robô NAO.

O modelo que será trabalhado neste projeto é a plataforma robótica *NAO*, e robô foi criado pela empresa *Aldebaran - Softbank* com o objetivo de auxiliar crianças para tratamentos em hospitais, autista e entre outros, por isso, ele apresenta uma aparência mais infantil singela, pois tem uma possibilidade retirar toda e qualquer tensão em que o paciente possa apresentar, assim, sendo possível aplicar em uma sessão de terapia.

A plataforma possui alguns recursos que possibilitam diversas aplicações, como (*SOFBANK GROUP - ALDEBARAN*, 2023):

- 25 graus de liberdade: são a quantidade de variáveis independentes que determina o comportamento de um sistema ou modelo. Na mecânica, por exemplo, o grau de liberdade de um corpo rígido pode ser determinado pela contagem de dimensões de movimento independentes;

- 7 sensores de toque: Botões capacitivos dando liberdade a diversas funções, localizados: 2 nas parte externa das mãos, 2 nas pontas dos pés e 3 no topo da cabeça;
- 4 microfones direcionais: possibilitando reconhecimento de voz ou análises de conversas;
- 2 Auto falantes;
- 2 Câmeras;
- 2 Sonares: sensores de detecção obstáculos a frente;
- Giroscópio, acelerômetros
- Aberto e completamente programável.

Existem diversas aplicações superinteressantes para esta plataforma, como por exemplo, existe uma copa do mundo de futebol de robôs em que algumas equipes programam suas plataformas para simular uma partida de futebol. A equipe organizadora acredita que se é possível programar os robôs para jogar futebol, então é possível atingir muitas outras realizações a partir da plataforma (*HTWK ROBOTS*, 2019). Existem programas educacionais para crianças autistas que utilizam dos robôs, palestras em hospitais universitários e entre outros âmbitos. Então, por que não seria possível fazê-lo auxiliar profissionais terapêuticos?

Por sua vez, a interação humana com o auxílio do uso da robótica vem sendo cada vez mais estudada, e apresentando-se uma área multidisciplinar (TAKAHASHI, 2018). A robótica pode ter uma relação com a pedagogia, medicina, industrial, construção civil e muitas outras áreas. Sendo assim, pode-se mencionar o tratamento de crianças com autismo, como por exemplo uma pesquisa realizada onde teve uma participação de terapeuta ocupacional, psicóloga e fonoaudióloga e com auxílio de uma plataforma robótica, todos tiveram uma grande surpresa com relação a aceitação por maior parte dos usuários da pesquisa. Pode-se observar o lado exclusivo da psicóloga da pesquisa, e o uso do modelo robótico que conseguiu obter bastante atenção das crianças e muita curiosidade em interagir com o robô(SANTOS, 2019). Deste modelo, se faz uma excelente adaptação de interação humano e robô dentro de uma sessão de terapia para auxiliar o profissional para obter ainda mais precisão em diagnósticos ou auxiliar seus pacientes.

2.3 Processamento/Normalização das Imagens

2.3.1 O que é uma imagem

Antes mesmo de falar sobre robótica, emoções e sessões de terapia, é importante aprender um pouco sobre **imagem**. Mas, de fato, o que seria uma imagem?

A olhar o conceito físico, imagens são formadas pelo reflexo de ondas eletromagnéticas geradas por um corpo capaz de gerar luminescências(Sol), e são capitadas por receptores sensíveis a luz capazes de interpretação daquele objeto que sofreu sua reflexão. Como, por exemplo, o sol emite uma onda eletromagnética, que são colididas em uma peça e refletidas aos nossos olhos, que por fim é interpretada e moldada a imagem daquele objeto. Contudo, estas ondas não possuem cores fundamentalmente, é apenas uma interpretação do nosso cérebro aos estímulos captados pelos nossos olhos (HENRIQUE et al., 2019). Porém, os computadores, obviamente não possuem olhos e nem um cérebro, por isso, é necessário que interpretem as informações de maneira diferente.

A interpretação do computador da luz é feita basicamente por números. Mas, as cores se diferem uma da outra devido ao seu comprimento de onda, e a luz branca é o conjunto de cores. E, as cores são formadas a partir das cores primárias, sendo elas o vermelho, verde e azul, que também vem do termo inglês *Red, Green* e *Blue* (RGB), e delas serão a base de todas as cores observáveis.

Uma imagem digital é separada em pequenas frações dito como *pixels*. Eles são criados a partir de uma única cor, que em um todo irão formar uma imagem (GIBSON; JOHNSON; PADGETT1, 2020). Ou seja, uma imagem em *Video Graph Array*(VGA) apresentam uma resolução/matriz de 640 *pixels* de largura por 480 de altura(640 x 480) calculando um total de 307200 *pixels*. Já, caso formos para imagens ditas 4K ultrapassamos um total de 4096 x 2160 *pixels*. E, nisto na figura 17 é observável a grande diferença entre essas duas características.

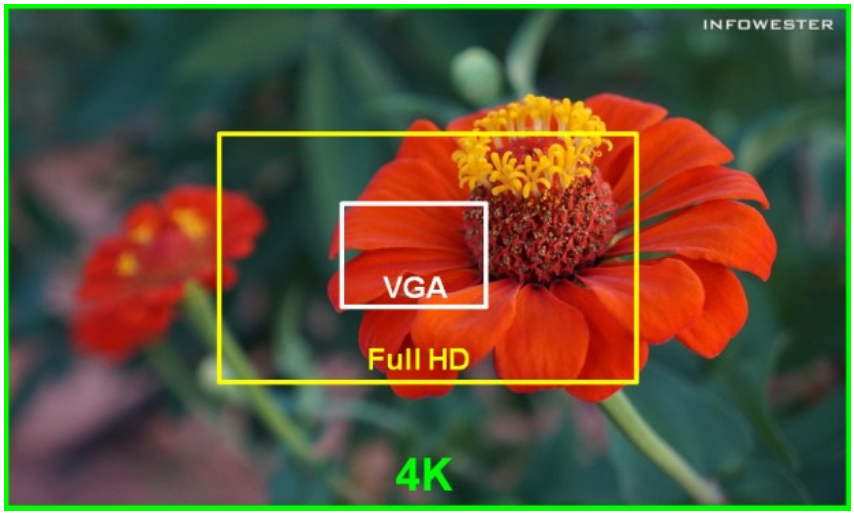

Figura 17 – Imagem VGA vs 4k.

Fonte: *Infowester* (*INFOWESTER*, 2020)

Com isso, numericamente, o *pixel* é composto por um vetor de 3 elementos para representar exatamente todas as cores primárias: [R, G, B]. A partir disso, cada elemento deste vetor será descrito de 0 a 255 reproduzindo exatamente a intensidade daquela respectiva cor, como por exemplo temos a seguinte figura 18 com sua, respectiva, matriz numérico de

cores:

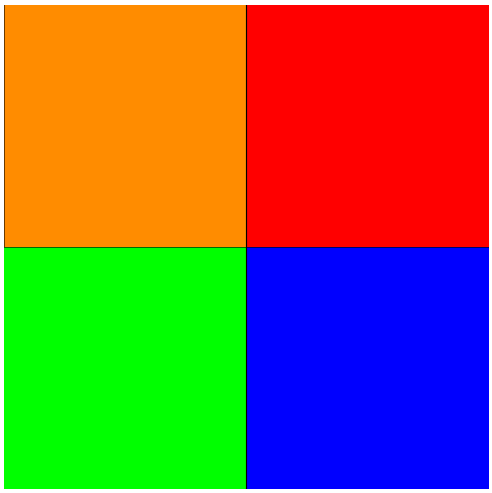

Figura 18 – Configurações de cores

$$\begin{bmatrix} [255, 140, 0] & [255, 0, 0] \\ [0, 255, 0] & [0, 0, 255] \end{bmatrix} \tag{2.1}$$

Portanto, assim, o computador tem a capacidade de a partir desses dados numéricos ler, modificar, apagar e reconhecer padrões de dados que forem necessários para aplicações diversos até talvez para reconhecimento de emoções pela face de seres humanos.

2.3.2 Rede de Computadores

Para ter a possibilidade da trocar dados entre o robô *NAO* ou até mesmo entre mais de um computador de forma remota é preciso ter uma comunicação de rede e estarem conectados nela. Por isso, será necessário o uso do protocolo *Transmission Control Protocol / Internet Protocol*(TCP/IP) que será utilizado neste projeto.

O TCP/IP é organizado por meio de camadas a partir do conjunto de protocolos de comunicação entre computadores (DALOSTO, 2019). Estas camadas têm como responsabilidade a transmissão de dados e/ou dar serviço ao próprio as camadas mais superiores.

O funcionamento do protocolo TCP/IP é bem simples, pois temos a camada da aplicação que realiza uma requisição para a camada de transporte, que por sua vez irá unificar estes dados em pacotes e encaminhar para a camada de internet, e assim serão entrelaçados e transportados para o endereço de destino.

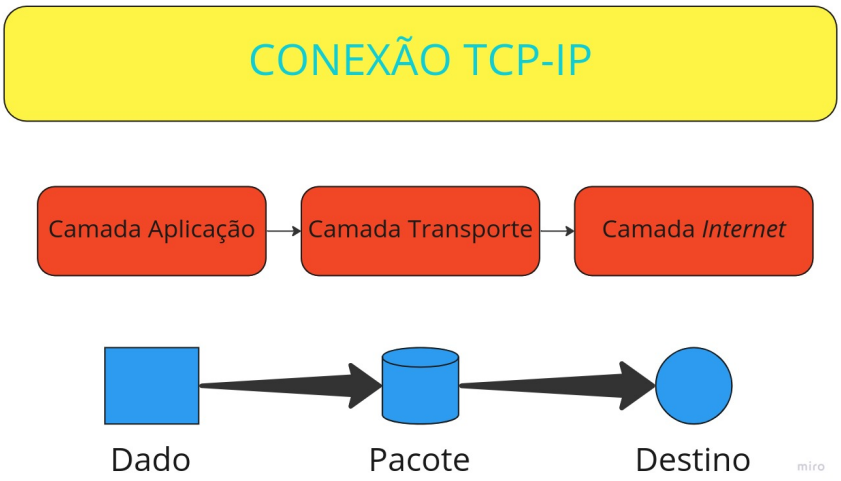

Figura 19 – Protocolo TCP/IP

2.3.3 *Keypoints*

Como visto na sessão de imagens 2.3.1, uma imagem pode demonstrar muitas informações, pois além do usuário estar em um ambiente completamente movimentado que causará uma distorção da pessoa estudada, como pode também ter diferentes tipos de imagens como pinturas de faces e outros. Desta forma, a plataforma tem que ser capaz de identificar o paciente e tentar aplicar extrair dados específicos para facilitar e aumentar a precisão do reconhecimento de emoções.

Com isso, usando técnicas de processamento de imagens pode-se aplicar um reconhecimento de face para filtrar somente a face do paciente, e além disso, extrair os *Keypoints*, que são pontos chaves para especificar uma indicação das partes da face do paciente. Por exemplo, seria capaz de informar o formato dos olhos se estão fechados ou abertos, o desenho da boca pode se apresentar com um sorriso ou a boca fechada, as sobrancelhas se estão tensas ou bem abertas

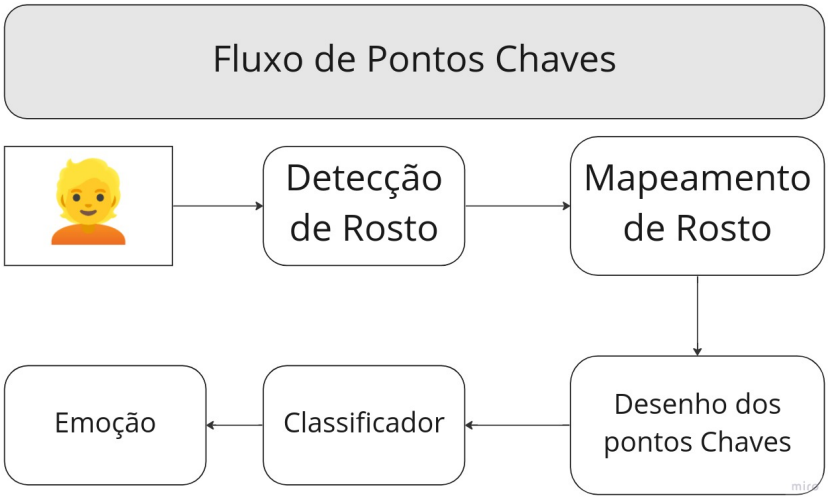

Figura 20 – Metodologia de execução por Pontos Chaves

Esta metodologia (PALESTRA et al., 2015) apresentou uma ótima precisão com cerca de 95% de acurácia (base de dados de treinamento), por isso, esta pesquisa seria perfeita para trabalho com a plataforma, pois, além de ter um ótimo classificador para reconhecimento de emoção, aumentaria bastante a performasse do robô, pois teria que tratar somente os pontos chaves, e assim, ter uma taxa de reconhecimento muito maior, reconhecendo talvez de 8 a 10 imagens por segundos.

2.4 Rede Neural Convolucional

Para realizar a classificação das emoções neste trabalho foi escolhido o uso de redes neurais convolucionais que é um tipo de aprendizado de máquina supervisionado. Elas foram propostas por Yann LeCun e foram inspiradas na percepção visual dos humanos e tem alta precisão em aplicações relacionadas a visão computacional.

2.4.1 Aprendizado de máquina e termos básicos

As redes neurais são uma subclasse de aprendizado de máquina, que por sua vez geralmente é categorizado em **aprendizado supervisionado**, **aprendizado não supervisionado** e **aprendizado por reforço**. No aprendizado supervisionado o algoritmo é treinado com um banco de dados contendo várias entradas e suas supostas saídas ou categorias. Já no aprendizado não supervisionado o algoritmo recebe um banco de dados contendo várias características e ele tenta aprender propriedades desejadas deste banco. E enfim, no aprendizado por reforço é fornecido um *feedback* durante o uso do programa e estes podem ser premiações ou punições.

Um dos termos mais usados em aprendizado de máquina é a **classificação** que segundo (GOODFELLOW; BENGIO; COURVILLE, 2016), neste tipo de problema, o programa de computador deve especificar em quais das k categorias uma entrada pertence, sendo k o número de categorias. Neste caso, as entradas serão imagens recebidas pela câmera do robô *NAO* e as categorias são as 7 ou 8 emoções dos bancos de imagem *FER2013* ou *AffectNet*.

Outro termo muito importante é a *acurácia* que é a proporção de exemplos para os quais o modelo retorna uma saída correta. O termo oposto seria a *taxa de erro*, ou seja, a proporção de exemplos para os quais o modelo retorna uma saída incorreta.

Ao referir-se ao treinamento redes neurais, muito se usa os termos ***overfitting*** e ***underfitting*** que se referem a capacidade de generalização da rede. *Underfitting* ocorre quando o modelo não consegue obter um erro suficientemente baixo para entradas do banco de dados de treinamento. Já o termo *overfitting* se refere a casos em que há uma diferença muito grande entre a taxa de erro de treinamento e a taxa de erro de validação (GOODFELLOW; BENGIO; COURVILLE, 2016). Sendo assim quando há *underfitting* a rede neural não consegue uma boa taxa de acertos e ainda pode ter melhorias ao aumentar o

número de épocas de treinamento e quando há *overfitting* a rede neural começou a decorar os dados do banco de dados ao invés de aprender suas características para ter uma boa generalização.

Outra técnica muito útil e que foi utilizada neste trabalho foi validação cruzada (*cross-validation*) onde os bancos de dados são divididos em duas partes, por exemplo, 75% para treinamento e os outros 25% para validação, desta forma é possível entender melhor a capacidade de generalização da rede neural (DUDA; HART; STORK, 2001).

Além de todas as técnicas e termos explicados acima é importante mencionar também a **matriz de confusão**. Elas são utilizadas para avaliar a performance de um modelo de classificação comparando o que foi previsto com os valores reais das classes. Os valores na diagonal principal são valores onde a classificação foi correta e todos os outros valores são representações de erros de classificação.

As matrizes de confusão geradas neste projeto são feitas com amostragem reduzida (down-sampling) do banco de testes.

2.4.2 As redes neurais convolucionais (CNNs)

Uma rede neural é um modelo computacional que tenta imitar a forma como o cérebro humano funciona, aprende e processa informações. Elas são compostas por vários nós interconectados em que cada um deles realiza cálculos matemáticos que definem com as entradas são processadas.

A primeira camada de uma rede neural é a entrada, que no caso de imagens tem seu tamanho definido pela quantidade de *pixels* da mesma. Sendo assim, com uma imagem de 48x48 e em *rgb*, a entrada será de $48x48x3 = 6912$. Já as camadas que vem em seguida são chamadas de camadas ocultas e a última de camada de saída.

Cada um dos nós de uma rede neural é conectado a próxima camada a partir pesos, que são apenas valores que auxiliam na produção de uma saída desejada como em classificações. Estes pesos são iniciados de maneira aleatória na definição da rede, e então ocorre o processo de treinamento para defini-los.

No processo de treinamento imagens são fornecidas a rede neural e observa-se o resultado produzido na camada de saída, este valor em conjunto com a saída desejada é usado em funções de custo, que definem basicamente uma taxa de erro. Sendo assim este erro produzido, que na verdade é uma função, é derivado de forma a obter a direção de um mínimo local do erro, e assim este valor é multiplicado por uma taxa de aprendizado α que é subtraído dos valores dos pesos de forma a produzir a cada passo, valores estatisticamente mais corretos.

As redes neurais convolucionais (*CNNs*) são um tipo específico de rede neural ar-

tificial que são eficazes para processamento de dados de imagem e vídeo como foi citado anteriormente. São baseadas em uma estrutura hierárquica de camadas onde as operações de convoluções serão capazes de separar informações básicas da imagem, como por exemplo as bordas. Assim, após várias camadas espera-se ter as várias características que formam o dado completo separadas.

Seu funcionamento básico consiste na aplicação de filtros 2D sobras entradas da imagem, gerando as representações das características as imagens. Estas operações são formadas pela operação conhecida como convolução que por sua vez, são seguidas de operações de *max-pooling* que reduzem a dimensionalidade dos dados. A seguir, a informação é passada por camadas densas ou totalmente conectadas que enfim fazem a classificação. A seção 3.5 explica melhor o funcionamento de cada uma das camadas usadas em uma *CNNs*.

Em termos matemáticos, as redes neurais convolucionais utilizam a operação de convolução ao invés da multiplicação matricial em pelo menos uma das suas camadas internas. Segue abaixo na equação 2.2 a representação matemática da convolução (GOODFELLOW; BENGIO; COURVILLE, 2016).

$$s(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)da = (x * w)(t) \tag{2.2}$$

3 Materiais e Métodos

3.1 *NAO* - Ambiente de desenvolvimento

3.1.1 O motivo do uso de um computador externo

Para o início do processo de desenvolvimento do projeto de reconhecimento de emoção por meio de expressões faciais, será necessário entender em primeiro lugar o uso dos recursos da plataforma robótica *NAO*.

Com o uso da documentação da *NAOqi* (*DOCUMETAÇÃO NAOQI*, 2023), percebe-se que existem dois modelos de elaboração do uso dos recursos, e eles são aplicados de duas maneiras, interna e a externa, a grande diferença entre ambas as metodologias é o uso de processos de comunicação, que consiste em usar qualquer tipo de motor, sensor ou criar qualquer tipo de comportamento a partir do modelo de comunicação *IP*(Internet protocol), ou realizada integralmente internamente dentro da plataforma robótica.

Desta maneira, para criações futuras de relatórios clínicos, avaliações em tempo real de terceiros, principalmente de profissionais de psicanalista, ou até mesmo realizar sessões de forma remota, sem a interferência humana ou extrema tensão de uma sessão de terapia, se faz necessário aplicar estas metodologias de maneira aplicando o sistema de comunicação IP, ou seja, usar todos os recursos da plataforma de maneira remota.

3.1.2 Comunicação entre o robô e o computador externo

Dentre a infinidade de recursos que a *NAOqi* oferece, um deles é a possibilidade de utilizar um computador externo para controlar o robô ou fazer o processamento de dados, e para isto se faz necessário entender que existem alguns processos iniciais para se desenvolver. Ao observarmos, o código apresentado no Apêndice B.1, nele se apresenta que precisamos de duas coisas essenciais:

1. **Endereço** de conexão e sua respectiva **porta** de comunicação;
2. **Sessão** de desenvolvimento com o robô.

Em primeiro lugar, é preciso se atentar ao endereço de *IP* (*NAOIP*) que é o rótulo numérico único para comunicação direta com o robô, mas por ser um *IP* dinâmico ele sempre apresentará o seu valor alterado, que no caso do *NAO* sempre que ele for desligado e ligado novamente terá um novo endereço de *IP*. É necessário também estar atento a porta, que no

caso do *NAO* ela é sempre fixa e de valor 9559(NAOPORT) (*CONEXÃO SESSÃO TCP-IP - NAOQI*, 2023).

Com as devidas configurações demonstradas anteriormente, o próximo passo de ligação entre o computador e o NAO é a criação de uma sessão utilizando uma conexão *TCP-IP*, verificado na sessão 2.3.2, pois é a partir dela que podemos realizar qualquer troca de informação necessária para o desenvolvimento de todo o tipo de projeto, como mostrado a seguir:

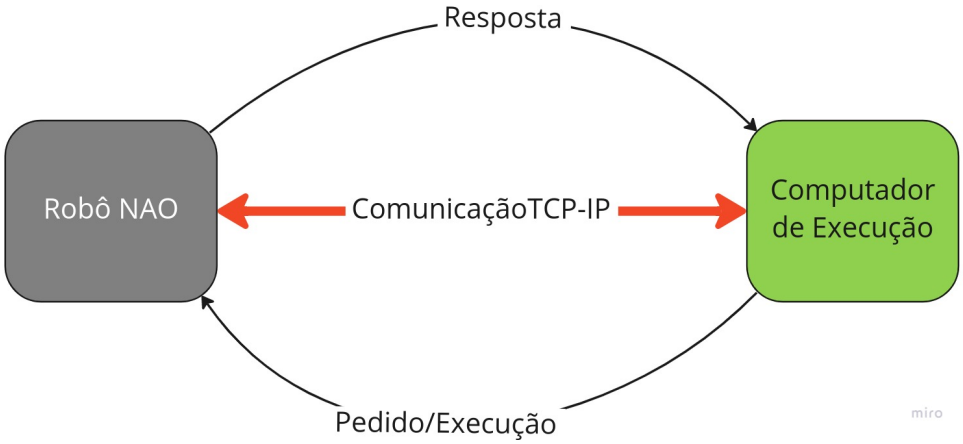

Figura 21 – Troca de informações entre o computador e o Robô NAO.

Com base na figura 21 é notório que todo pedido realizado pelo usuário/desenvolvedor, como um pedido para o robô falar ou obter informação de qualquer sensor, precisa passar pela rede. E para este trabalho é necessário obter as imagens capturadas pela câmera e enviá-las ao computador externo para processar as emoções. Tudo isso através da comunicação *TCP-IP* que se apresenta como:

$$tcp://\textbf{\textit{NAOIP}} : \textbf{\textit{NAOPORT}} \tag{3.1}$$

3.2 Comunicação local para transferência de imagens

O primeiro problema encontra dentro do desenvolvimento do projeto foi que todos os processos de reconhecimento profundo de face, e até mesmo a extração dos *Keypoints* como pré processamento da entrada, estão desenvolvidos na versão 3.11 do *Python*, e que todo o *framework* do robô está na versão 2.7 do *Python*. E, por isso, foi necessário procurar algumas soluções, e a melhor delas é transferência das imagens em uma rede local que possibilitasse enviar do *Python 2.7* que é a da *NAOqi* para um programa de execução no *Python 3.11*.

Desta maneira, a solução encontrada para haver a conversa entre as duas metodologias de programação é abrir uma conexão *socket* entre ambas as versões do *Python*. Com isto, os códigos apresentados tanto no apêndice B.2 (DOCUMENTAÇÃO *SOCKET* - PYHTON 2.7, 2020) e no apêndice B.3 (DOCUMENTAÇÃO *SOCKET* - PYHTON 3.11, 2023) é possível

encontrar a seguinte maneira de execução. Primeiramente, com as configurações de conexão já pré-estabelecido, tanto o *IP* (*host*) quanto a porta de conexão (*porthost*), foi criado um serviço *socket* com as configurações básicas, e principalmente com a quantidade de troca de dados, pois este foi o próximo problema encontrado.

Para a manipulação de imagem, tanto com o uso de processamento de imagens ou o uso da visão computacional, era necessário que as imagens obtidas do *NAO* fossem de uma resolução mais alta, por isso, na linha 23 do código B.2 tivemos uma resolução VGA(Video Graphics Array - Matriz Gráfica para Vídeo), ou seja, temos uma matriz de 640x480 *pixels*. Contudo, para a comunicação local *socket*, o tamanho máximo para transferência é de 65kB(kiloBytes), porém, os *frames* capturadas pelo *NAO* são por volta de 300kB. A partir disto, a solução encontrada para o problema anterior foi de dividir a imagem em pacotes.

Em B.2, a partir da linha 63, definimos um número de pacotes que a imagem atual deverá possuir, assim, pegamos o total da imagem(*tamImagem*) e dividimos pela quantidade máxima de transferência(*maxTransf*), e definimos a quantidade de pacotes(*numPacks*), por exemplo, caso a imagem possuir 300kB de tamanho e por definição do *socket* tem que no máximo transferir 65kB, desta forma, apresenta:

$$numPacks = \begin{cases} \frac{tamImagem}{maxTransf}, & \text{se } \frac{tamImagem}{maxTransf} \in \mathbb{Z} \\ \\ \mathbb{Z}\left\{\frac{tamImagem}{maxTransf}\right\} + 1, & \text{se } \frac{tamImagem}{maxTransf} \in \mathbb{Q} \text{ e } \neq \mathbb{Z} \end{cases} . \tag{3.2}$$

$$numPacks = \mathbb{Z}\left\{\frac{tamImagem}{maxTransf}\right\} + 1 = \mathbb{Z}\left\{\frac{300.000}{65.000}\right\} + 1 = \mathbb{Z}\left\{4{,}62\right\} + 1 = 5 \tag{3.3}$$

Neste modelo, o *socket* terá que enviar dos tipos de dados, a primeira será a quantidade de pacotes que aquela imagem apresenta, e por fim será enviado os pacotes que compõe a imagem.

E, por fim, o processo de captura dessa imagem realizada no *Python 3.11*, ele apresenta praticamente as mesmas características quanto ao dito anteriormente, entretanto, para este caso só receberemos os dados, tanto do número de pacotes da imagem quanto os próprios pacotes. E, assim unificamos todos os dados dentro de um *buffer* e decodificamos para ser interpretada com uma imagem. Permitindo, assim, realizar qualquer processamento de imagem ou recuperar qualquer dado com a visão computacional.

3.3 Desenho dos *Keypoints*

Esta etapa do projeto é manipulação da biblioteca *Mediapipe* que será a grande responsável de extrair os *Keypoints* nas imagens recuperadas diretamente ao *NAO*. E, então,

mais a frente, ter a capacidade de aplicar os pesos da rede neural treinada e tentar informar as emoções apresentadas pelo usuário.

3.3.1 Desbravando os Recursos do *Mediapipe*

Com as imagens sendo já recuperadas diretamente do robô, o próximo passo é de encontrar nas imagens a face das pessoas, e assim, ter a possibilidade de reconhecimento de emoções. A partir disso, a biblioteca do *Mediapipe* é extremamente interessante. Ela, além de ter a possibilidade de reconhecimento facial, tem a capacidade de monitorar qualquer ponto da face, o mapeamento da iris dos olhos, das mãos, pose e até mesmo de objeto. (DOCUMENTAÇÃO DO *MEDIAPIPE*, 2022).

No Apêndice B.4 temos o código do *Face Mesh*, que é o processo de reconhecimento e de mapeamento de todos os pontos da face da pessoa que seja possível. Assim, é possível ter toda a localização de mudança da face, ou até mesmo a permanência de neutralidade. Nós podemos observar nas seguintes figuras:

Figura 22 – Imagem de uma face da Base de Fotos

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 23 – Mapeamento com o *FaceMesh* com os seus pontos

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 24 – Imagem da face com expressão de raiva

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 25 – Mapeamento para a expressão de raiva

Fonte: *AffectNet* (MAHOOR, 2011)

Ao observar as figuras 22 e a 24 é notável a diferença de expressão facial apresentada de ambas as imagens, enquanto uma desenvolve uma aparência de abertura de boca e de sobrancelha, a outra temos uma pressão dos lábios superiores e no centro das sobrancelhas e fechamento da boca, ou seja, um identifica-se com alegria e a outra de raiva. E, nisto, os pontos apresentados no mapeamento do *Mediapipe* nas imagens 23 e 25 conseguiríamos realizar um desenho de demonstração da expressão específica.

3.3.2 Aplicações de Novas Características

Com o mapeamento já pré estabelecido, as imagens permitem criar novas características para estabelecer maior precisão de mudanças da face do usuário e facilitar o processamento da rede neural, tanto de treino quanto na avaliação final da emoção.

Como dito no 2.3.1, para reduzir consideravelmente a quantidade de dados de processamento, primeiramente devemos aplicar uma conversão da estrutura da imagem de RGB(*Red, Green e Blue*) para estrutura em cinza e sua intensidade. Portanto, uma diminuição considerável da estrutura da imagem.

Figura 26 – Imagem da face com expressão de surpresa

Fonte: *AffectNet* (MAHOOR, 2011)

Figura 27 – Mapeamento para a expressão de surpresa em cinza

Fonte: *AffectNet* (MAHOOR, 2011)

A segunda etapa é ter o máximo possível de dados e menos possível ainda de informações, e desta maneira foi aplicado duas metodologias, somente os *keypoints* e filtrar aqueles pontos que apresentarem maior influência de apresentação de uma emoção na face do usuário. Desta forma, a construção de uma tela que apresente somente os pontos será em zerar completamente a imagem e desenhar exclusivamente os pontos.

Figura 28 – Imagem da face com expressão de tristeza

Fonte: *AffectNet* (MAHOOR, 2011)

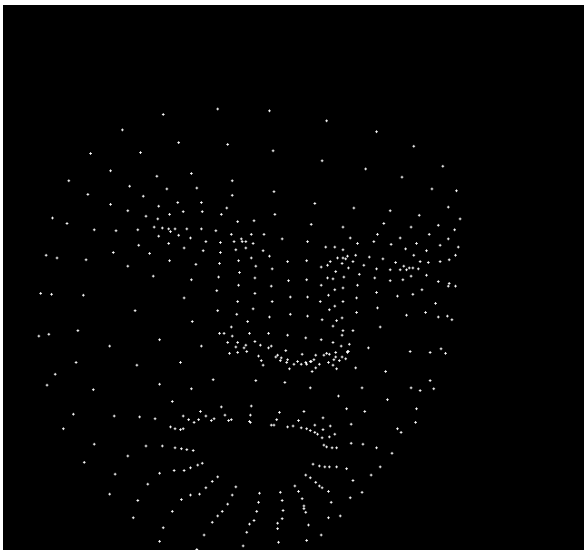

Figura 29 – Mapeamento para a expressão de tristeza somente com *keypoints*

E, por fim, a partir dos testes aplicados é possível ser notado que alguns pontos não apresentam mudança em diferentes expressões, e, com isso, para em termos estratégicos

e para melhorar a performance do projeto, ocorreu uma filtragem destes pontos. Assim, ao verificar (MAPEAMENTO DOS *KEYPOINTS*, 2022) todos os pontos produzidos pela biblioteca têm a capacidade de selecionar apenas ao interesse os pontos desejados.

Figura 30 – Imagem da face com expressão de neutralidade

Fonte: *AffectNet* (MAHOOR, 2011)

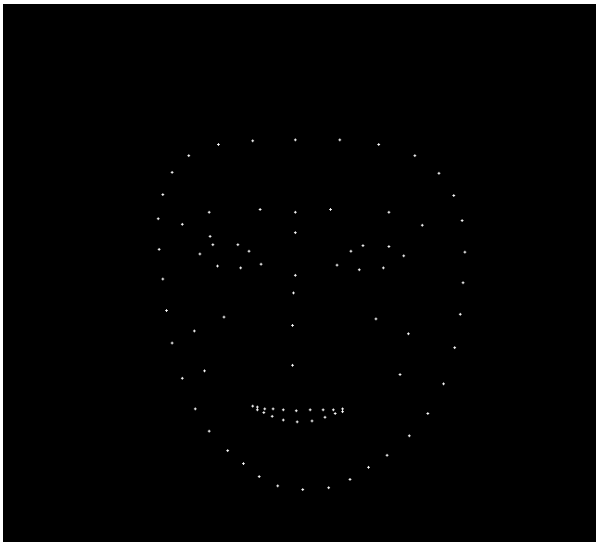

Figura 31 – Mapeamento para a expressão de neutralidade somente com *keypoints*

Portanto, com este todo processo de desenvolvimento, foi toda a estratégia aplicada a todo o projeto em relação ao processamento de imagem que será aplicada em conjunto ao treinamento quanto ao reconhecimento para obter o melhor rendimento de reconhecimento de emoções.

3.4 Os Bancos de Dados

Com o uso da metodologia aprendida anteriormente, e antes mesmo de entrar em todo o processo de treinamento da rede neural, foi necessária a manipulação das imagens do banco utilizando os filtros dos *keypoints* e da procura da face do usuário dentro de todas as imagens do banco de dados para que elas estivessem completamente preparadas para o treinamento da rede.

O processo de recuperação das bases de dados apresenta algumas propriedades distintas.

3.4.1 FER 2013

O banco *FER-2013* é bem estruturado com a separação de treinamento e validação 75%/25%(Treino/Testes) e com uma estrutura onde o nome das pastas correspondem as

emoções. O banco de imagens pode ser facilmente baixado na plataforma *Kaggle*. Segue abaixo a quantidade de imagens de cada classe:

- *Train*/Treino
  - *Angry*/Raiva (3995 imagens)
  - *Disgust*/Nojo (436 imagens)
  - *Fear*/Medo (4097 imagens)
  - *Happy*/Alegria (7215 imagens)
  - *Neutral*/Neutro (4965 imagens)
  - *Sad*/Triste (4830 imagens)
  - *Surprise*/Surpresa (3171 imagens)
- *Test*/Teste
  - *Angry*/Raiva (958 imagens)
  - *Disgust*/Nojo (111 imagens)
  - *Fear*/Medo (1024 imagens)
  - *Happy*/Alegria (1774 imagens)
  - *Neutral*/Neutro (1233 imagens)
  - *Sad*/Triste (1247 imagens)
  - *Surprise*/Surpresa (831 imagens)

Como pode ser visto a quantidade de imagens para emoções como o nojo é muito menor, o que tem efeitos sobre a rede neural além do fator que a quantidade de imagem é menor exatamente em emoções que são complexar de se analisar.

O artigo (KHANZADA; BAI; CELEPCIKAY, 2020) cita que o nível de acurácia humana no artigo é de apenas $65 \pm 5\%$ e que os melhores artigos publicados mostram uma acurácia de 75.2%.

3.4.2 AffectNet

A estrutura da base de dados *AffectNet* é um pouco diferente da anterior, pois as imagens são classificadas apenas por emoção, sem separação para treino e teste. Desta maneira, é preciso separar o banco de imagens para que seja possível fazer a validação cruzada, sendo assim, foi escolhido o valor recomendado pela maioria dos artigos de 75%/25%(Treino/Testes). Com isto, foi criado um código B.5 de separação decorrente ao necessário para teste é de treino. Segue abaixo a quantidade de imagens de cada classe:

- *Neutral*/Neutro (75374 imagens)
- *Happy*/Alegria (134915 imagens)
- *Sad*/Triste (25959 imagens)
- *Surprise*/Surpresa (14590 imagens)
- *Fear*/Medo (6878 imagens)
- *Disgust*/Nojo (4303 imagens)
- *Angry*/Raiva (25382 imagens)
- *Contempt*/Desprezo (4250 imagens)

Há artigos que conseguem resultados similares aos do *FER2013* como valores entre 60% e 70% de acurácia.

Após separar o banco em treinamento e validação (no caso do *AffectNet*), é necessário também realizar o processamento da base de dados, e extrair os *keypoints* em todas as imagens. E, por isso, todas as imagens passaram por toda o processamento dito na sessão 3.3. E, assim, todas as imagens estão separadas e completas para o uso da da rede neural.

3.5 Rede Neural Convolucional com 4 camadas

O modelo de camadas utilizado neste trabalho foi baseado no artigo (DEBNATH et al., 2021) que possui apenas 4 camadas de convolução e consegue uma boa precisão em poucas rodadas de treinamento.

A grande vantagem deste modelo é que por ter apenas 4 camadas e ser rápido para convergir, temos um treinamento mais rápido principalmente no caso do banco de imagens (**AffectNet**) que possui imagens grandes. O modelo pode ser visto na figura 32

Nas camadas da *CNN* utilizada há 4 outros blocos utilizados: convolução, *pooling*, normalização e ativação.

O principal objetivo da camada de convolução é extrair as *features* ou características da imagem. As características são extraídas através da operação de convolução utilizando um *kernel* que percorre a imagem como se fosse um filtro e cria uma imagem de tamanho reduzido com as *features* de forma mais evidente para a rede neural.

A camada de *pooling* tem como objetivo simplificar a informação proveniente da camada de convolução. Esta simplificação é feita com base no *kernel* utilizado na camada de convolução.

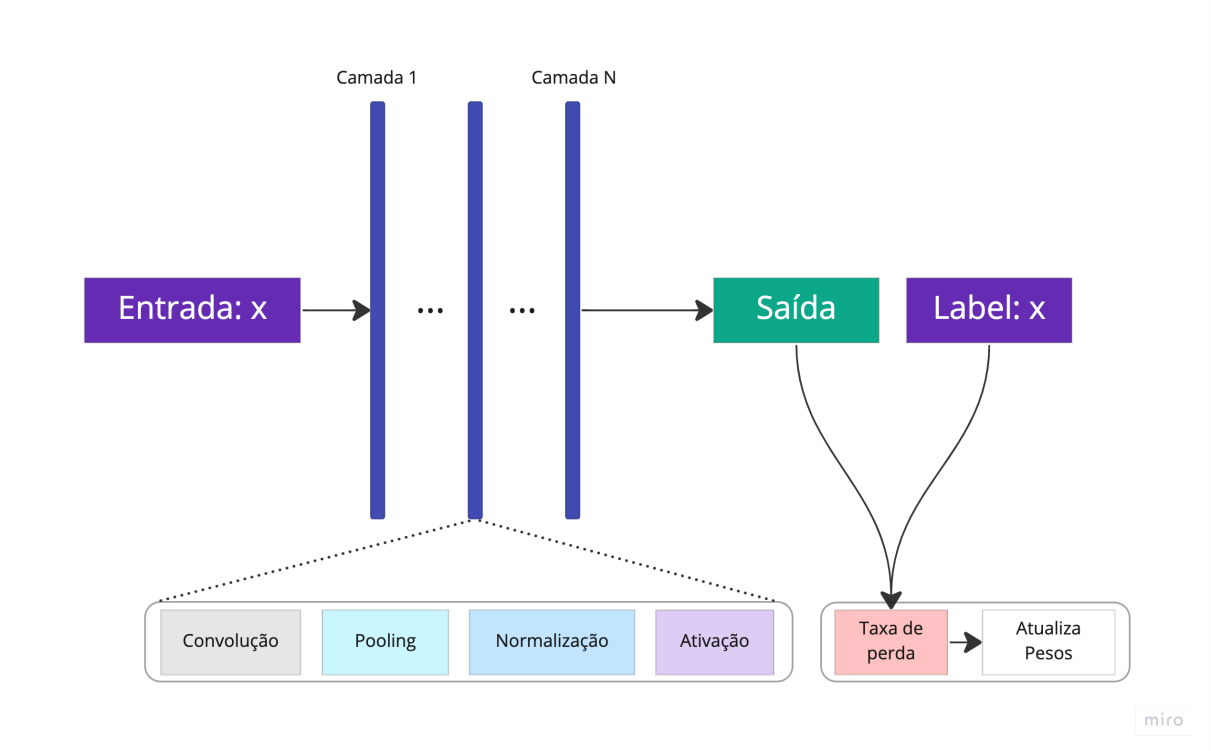

Figura 32 – Camadas da rede neural convolucional baseada no artigo (DEBNATH et al., 2021)

A camada de normalização é utilizada para acelerar o processo de treinamento ao manter os valores da ativação média próximo de 0 e o valor do desvio padrão da ativação próximo de 1.

Na camada de ativação utilizamos a função *ReLu*. Esta camada tem como objetivo trazer a não-linearidade ao sistema de forma que o sistema consiga aprender qualquer tipo de funcionalidade. A função*ReLu* é uma das mais utilizadas neste tipo de aplicação por ser mais eficiente computacionalmente quando comparada a funções como *sigmoid*, *tanh* e *Softmax*.

Apenas na última camada, conhecida como a camada totalmente conectada (*fully connected layer*), será utilizado a função *Softmax* e o número de nós deve ser igual ao de saídas desejadas, ou o número de emoções no caso deste projeto.

Este modelo de 4 camadas foi utilizado também no teste dos *keypoints* devido a quantidade de camadas ser menor e possibilitar que a rede neural seja treinada de forma mais rápida.

3.6 Transferência de conhecimento com *MobileNet*

Após obter os resultados e verificar acurácia de validação utilizando os dois modelos de sistema, com a extração de *keypoints* e sem ela, foram feitos vários testes com o objetivo de conseguir melhores resultados.

Sendo assim, foram realizados treinamentos utilizando um dos modelos mais famosos usados para *transfer learning* ou transferência de conhecimento, a *MobileNet*.

Os pesos utilizados nas camadas com a *MobileNet* são pesos treinados com a *ImageNet* e foram congelados, então apenas as últimas camadas adicionadas foram treinadas.

Neste modelo utilizou-se transferência de conhecimento com a *MobileNetV2*, foram utilizados os pesos da *ImageNet* e treinados apenas camadas totalmente conectadas que foram adicionadas ao fim da rede neural. A diferença entre este modelo e o primeiro modelo de *MobileNet* é que a sua arquitetura é baseada em uma estrutura residual invertida onde a entrada e a saída do bloco de resíduo são camadas finas de gargalo (SANDLER et al., 2018).

3.7 Treinamento dos modelos de Rede Neural

Para realizar o treinamento das redes neurais utilizou-se a plataforma *Google Colabs* e *Kaggle*, pois elas permitem o uso de *GPU's* com grande capacidade de processamento como a *Tesla T4*, possibilitando que o treinamento seja feito em uma quantidade de tempo significativamente menor e assim permitindo que mais testes sejam feitos.

O uso de duas plataformas se deve ao fato de que a quantidade tempo utilizando as *GPU's* é limitado.

Para chegar a melhores resultados evitando *overfitting* foi utilizado a técnica de parada antecipada, onde são checadas algumas condições nos valores de acurácia e taxa de erro (*loss*). Para aplicar está técnica utilizou-se o *callback EarlyStopping* (*KERAS*, 2023a). Na maioria dos treinamentos foi monitorada a taxa de erro de validação, pois a partir do ponto onde ela decresceu muito e começa a crescer novamente tem-se o *overfitting*. E além disto, para evitar que crescimentos momentâneos do erro de validação faça o treinamento para, utilizou-se o parâmetro *patience* com um limite de 15 a 20, sendo este valor a quantidade de épocas a checar antes de para o treinamento. E ao final executar esta função, o treinamento para e o valor de melhor desempenho é restaurado.

Para que o treinamento tivesse a menor duração possível, foi utilizado outra técnica onde os treinamentos são iniciados com uma taxa de aprendizado relativamente grande e depois tem seu valor decrescido quando certas condições são atingidas. Para isso foi empregue o uso da função de *callback ReduceLROnPlateau* (*KERAS*, 2023b). Assim, ao passar algumas épocas de treinamento sem nenhuma melhora no erro de validação, a taxa de aprendizado era diminuída para 10%, assim conseguindo o melhor valor do ponto de mínima em relação ao erro obtido pela rede neural.

3.8 Integração da rede neural convolucional ao *NAO*

Para finalizar, com o treinamento da rede neural realizado para o modelo final, foi necessário integrar os módulos criados e desenvolver algumas demonstrações. Deste modo, foram conectados os módulos que enviar imagens do *NAO* para o computador externo, o módulo que recebe as imagens e o modulo que faz as predições utilizando os pesos salvos durante o treinamento.

O sistema foi integrado e testado também com os *keypoints* e sem os mesmos para distinguir em qual sistema tem o melhor desempenho.

Com os pesos já gerados pela rede neural devido ao treinamento mostrado anteriormente, e para isto, a partir do código B.7, ao inicializar a classe *FacialExpressionModel* receberá dois argumentos diferentes, o primeiro será o caminho do arquivo de modelo da rede neural caracterizada por ser um arquivo *JavaScript Object Notation*(JSON), e o outro o caminho do arquivo que terá os pesos de cada camada e de cada nó caracterizada por ser um arquivo *.h5*.

Após isso, fazendo os devidos processamentos de imagens recebidas do robô, como transforma as imagens somente com os *Keypoints* ou tornar o tamanho das imagens para 224 por 224, nós as aplicamos para dentro do método *predictEmotion*, e seu retorno será exatamente aquela emoção que apresentar maior probabilidade daquela imagem. Será aplicado o seguinte modelo de execusão no treinamento:

Figura 33 – Imagem da recurada pelo Robô *NAO*

Figura 34 – Imagem aplicada no classificador

Desta forma, o classificador terá a capacidade de realizar uma predição da emoção expressa diretamente da câmera.

E, para finalizar, e aproveitando a rede local de comunicação entre os *Pythons*, pegar a emoção recebida da rede neural e enviamos para o *NAO* para que possa o próprio robô informar a emoção e criar um relatório de acordo com o tempo de observação.

Assim, o projeto como todo fica desta forma:

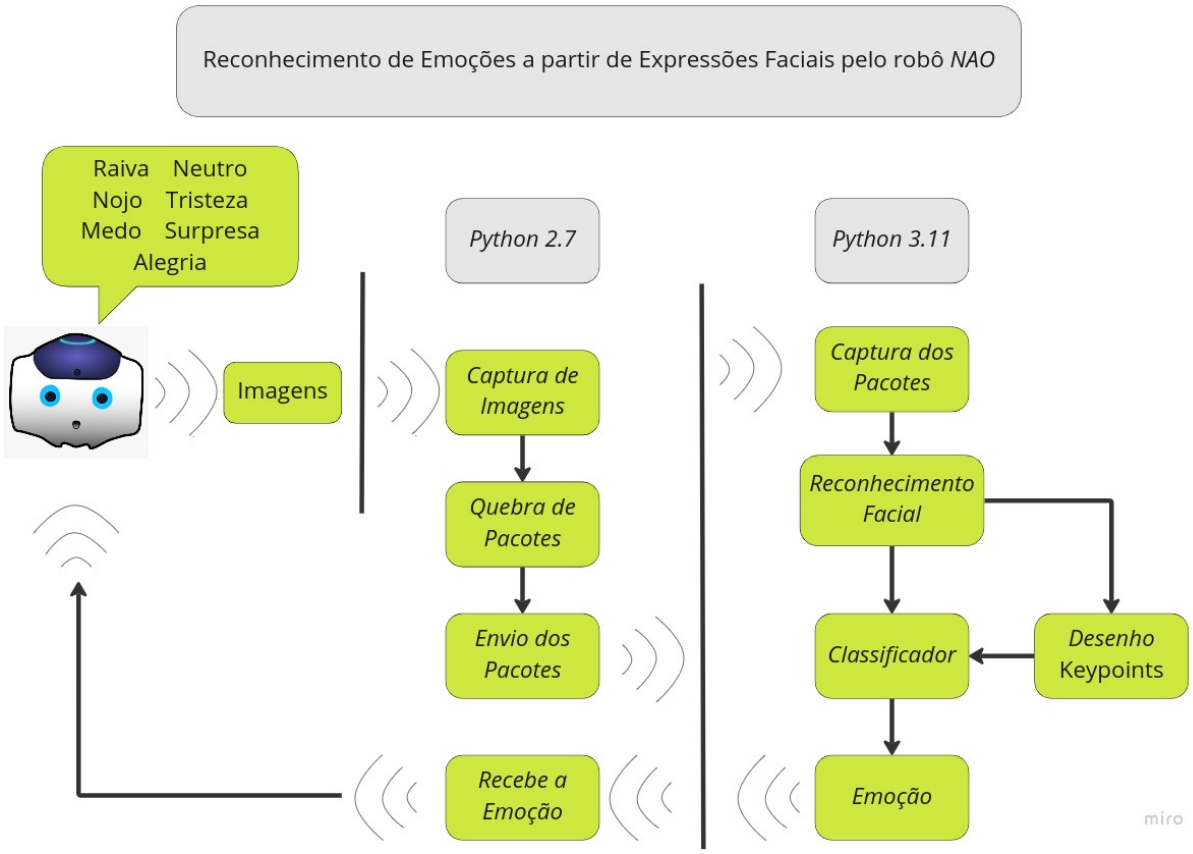

Figura 35 – Fluxo Completo do Projeto

4 Resultados

Neste capítulo apresentaremos os resultados obtidos a partir do desenvolvimento que foi apresentado no capítulo anterior incluindo gráficos, análise de precisão e análise dos erros.

Os dados avaliados foram obtidos a partir dos treinamentos da rede neural com as bases de dados *FER2013* e *AffectNet*. Além disso, será abordado o reconhecimento de emoções a partir de imagens adquiridas pela visão direta da plataforma robótica *NAO* e as possíveis abordagens aplicada a mesma. Finalmente, discutiremos os dados obtidos e apresentaremos diferentes conceitos e estudos para o seu desenvolvimento.

Pode-se observar que nos dois *datasets* utilizados neste trabalho, temos resultados bastante similares atingindo entre 60% a 65% de precisão nos melhores resultados. A razão deste valor não ser muito alto se deve a fato de que não é totalmente possível determinar a emoção de uma pessoa apenas considerando sua expressão facial, apesar de ser uma característica muito relevante.

4.1 Análise da extração de *keypoints* - FER2013

Como já foi mencionado anteriormente, para analisar o desempenho da rede neural com as *features* extraídas antes do treinamento, foram realizados alguns testes utilizando o banco de imagens *FER2013* e o modelo com apenas 4 camadas por ser mais rápido para treinar.

As duas próximas subseções mostram os resultados do treinamento das duas formas, com o módulo de extração de *keypoints* e sem o mesmo.

4.1.1 Resultados do treinamento sem extração de keypoints

Ao treinar redes neurais em aplicações de Machine Learning costuma-se fazer várias rodadas de treinamento para que seja possível encontrar múltiplos pontos de mínima em relação ao erro, e assim pode-se escolher o melhor resultado. Entretanto, ao utilizar redes neurais convolucionais pode ser mais custoso devido a quantidade de tempo para realizar o treinamento, mas apesar disso a *cnn* foi treinada 4 vezes com o banco *FER2013* e os resultados não foram significativamente discrepantes.

Entende-se que para que não haja *overfitting* é necessário utilizar os pesos da rede neural em ponto antes que o *loss* de validação comece a decrescer enquanto o *loss* de treinamento decresça também. Levando isto em consideração, ao realizar o treinamento utilizando

o banco de imagens *FER2013* foram obtidos aproximadamente 63% de precisão como pode ser visto nas figuras 36, 37, 38 e 39.

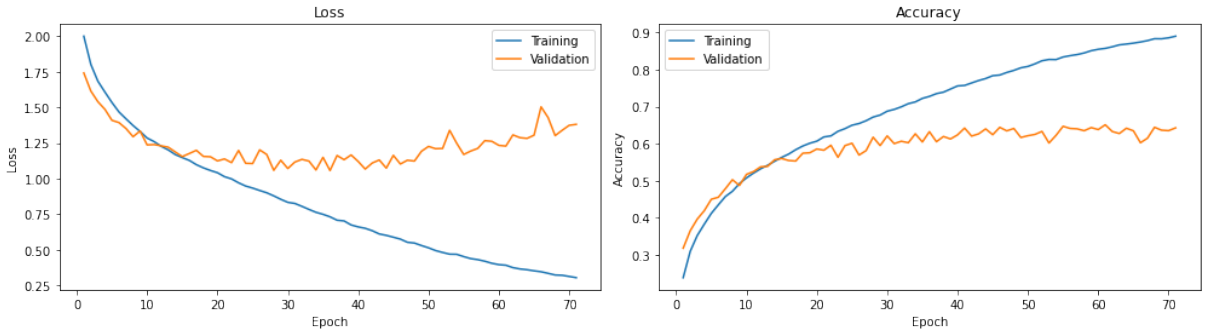

Figura 36 – Gráfico de acurácia e perda da CNN com 4 camadas - 1ª rodada

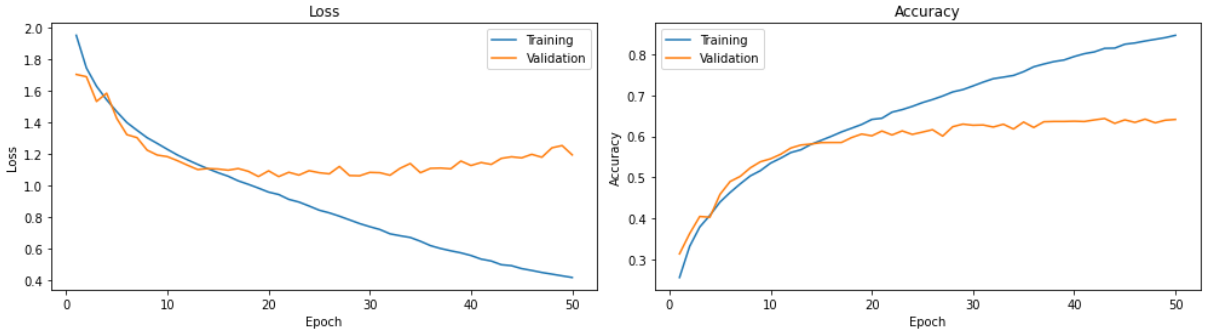

Figura 37 – Gráfico de acurácia e perda da CNN com 4 camadas - 2ª rodada

Figura 38 – Gráfico de acurácia e perda da CNN com 4 camadas - 3ª rodada

Figura 39 – Gráfico de acurácia e perda da CNN com 4 camadas - 4ª rodada

Como já mencionado anteriormente, não é possível ter resultados muito superiores, pois apenas a expressão facial seria muito pouco para detectar o real sentimento da pessoa naquele momento. Mas levando isto em consideração, este é um resultado foi bom se comparado aos resultados obtidos por outros artigos x, y, z.

4.1.2 Resultados extraindo os *Keypoints*

Ao extrair as características faciais e ao remover todo o restante da imagem, não foram obtidos resultados satisfatórios. Foram obtidos valores próximos de 48% na acurácia de validação como pode ser visto na figura 40.

Figura 40 – Gráfico de acurácia e perda da CNN com 4 camadas e extração de keypoints

A primeira característica discutida que pode ter causado este desempenho tão baixo utilizando este *dataset* seria devido as imagens serem muito pequenas, de apenas 48x48. Sendo assim, ao extrair vários *keypoints* em uma imagem tão pequena, não há muita precisão.

Por este motivo foram incluídos testes com outro *dataset* que contém imagens maiores, o *AffectNet* que será abordado nas próximas seções.

4.2 Treinamento dos dados *AffectNet*

4.2.1 Resultados do treinamento sem extração de *keypoints*

Ao realizar o treinamento da rede neural com 4 camadas utilizando as imagens do banco *AffectNet* foram obtidos aproximadamente 63% de precisão. Este resultado é similar ao obtido utilizando o banco *FER2013* apesar de ter 8 classes e ter imagens em qualidades bem superiores.

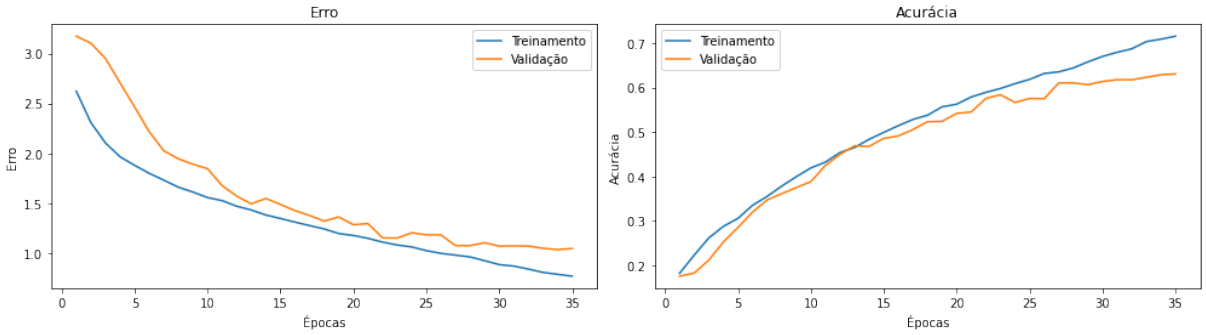

Figura 41 – Gráfico de acurácia e erro com 4 camadas e banco de imagens *AffectNet*

4.2.2 Resultados extraindo os Keypoints

Ao treinar a rede neural com 4 camadas utilizando o banco de imagens *AffectNet* foram obtidos 50% de acurácia de validação como pode ser visto na figura 42, o que é um resultado baixo em comparação ao treinamento com imagens completas, que obteve uma acurácia de 63%.

A principal diferença entre as metodologias é que a extração de keypoints resulta em uma significativa redução na quantidade de imagens úteis, já que muitas imagens não podem ser processadas devido às suas condições. Além disso, os gráficos de acurácia e erro são muito mais voláteis com a extração de keypoints.

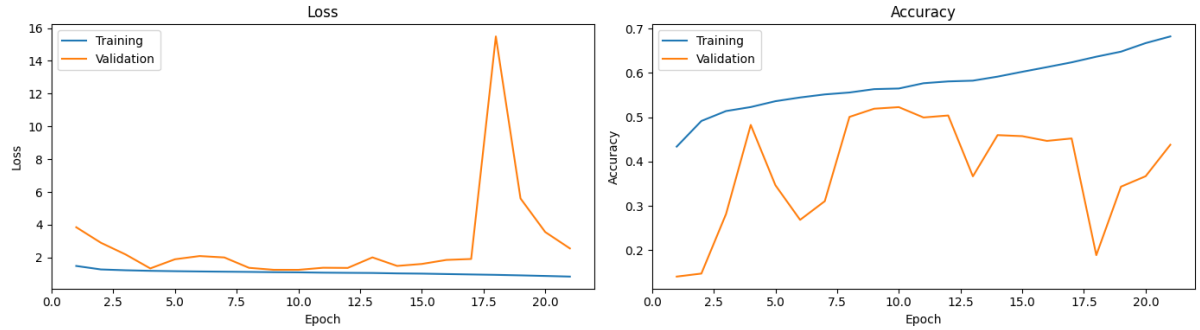

Figura 42 – Gráfico de acurácia e erro com 4 camadas, extração de *keypoints* e banco de imagens *AffectNet*

Como este método utilizando a extração de *keypoints* não obteve resultado satisfatório, o módulo foi descartado dos experimentos posteriores.

A extração de *keypoints* poderia ser melhorada utilizando uma quantidade maior de *keypoints*, fazendo verificações de quando a extração funcionou e ainda utilizando o desenho de figuras geométricas para entregar um resultado melhor a rede neural. Mas como o objetivo principal deste trabalho é desenvolver uma versão inicial do projeto em que o robô consiga reconhecer emoções, as melhorias do módulo de extração de *keypoints* ficarão para um futuro projeto.

4.3 Comparação dos resultados usando os diferentes modelos de rede neural

Como pode ser visto anteriormente a extração dos *keypoints* não favoreceram os resultados devido a perca de muita informação e devido ao fato que o *dataset* FER2013 é desafiador por ser desbalanceado, com imagens em baixa resolução, por ter uma precisão real por volta de 65%.

Deste modo, há outras maneiras de aumentar a acurácia da rede neural e um destes métodos é testar diferentes arquiteturas para ver qual se adéqua melhor ao problema. Neste trabalho foram testadas 2 arquiteturas: *MobileNetV2* e o modelo de 4 camadas de convolução que foi baseado no artigo (DEBNATH et al., 2021).

4.3.1 MobileNet

Como citado na metodologia, nesta etapa foi utilizada a versão *MobileNetV2* do *Keras*. Esta versão mostrou melhor desempenho nesta aplicação.

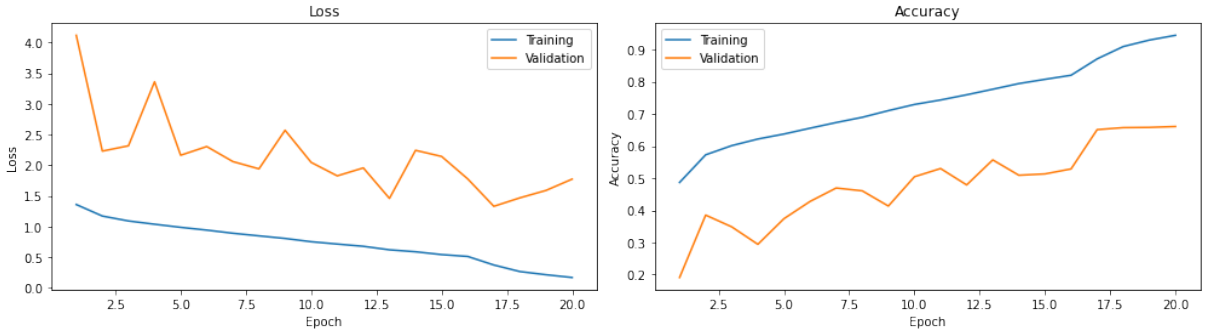

Figura 43 – Gráfico de acurácia e perda da CNN utilizando as camadas da *MobileNetV2*

Como pode ser visto no gráfico 43 foram obtidos 66% de acurácia de validação utilizando este modelo.

Ao observar a matriz de confusão tem-se uma visão mais realista dos resultados devido ao fato que ela mostra a precisão para cada uma das classes. Os melhores resultados estão nas classes *alegria*, *neutro*, *triste* e *surpresa*. Se o problema fosse reduzido para a predição de apenas estas classes os resultados seriam bem melhores. A classe *nojo (disgust)* é difícil de prever como foi mencionado anteriormente, até mesmo pela dificuldade de criar imagens, além do fato que algumas destas classes possuem alguma intersecção.

Pode-se observar também que na matriz de confusão quando a rede neural deveria prever surpresa, ela prevê quase a mesma quantidade de medo. E é notório a intersecção entre estas classes em estudos sobre faces, sendo assim, apesar de que a classificação teve uma grande taxa de erro ela tem sentido.

Algo peculiar que pode ser observado também é que a maioria dos erros das classes de raiva, nojo e medo foram para a classe triste.

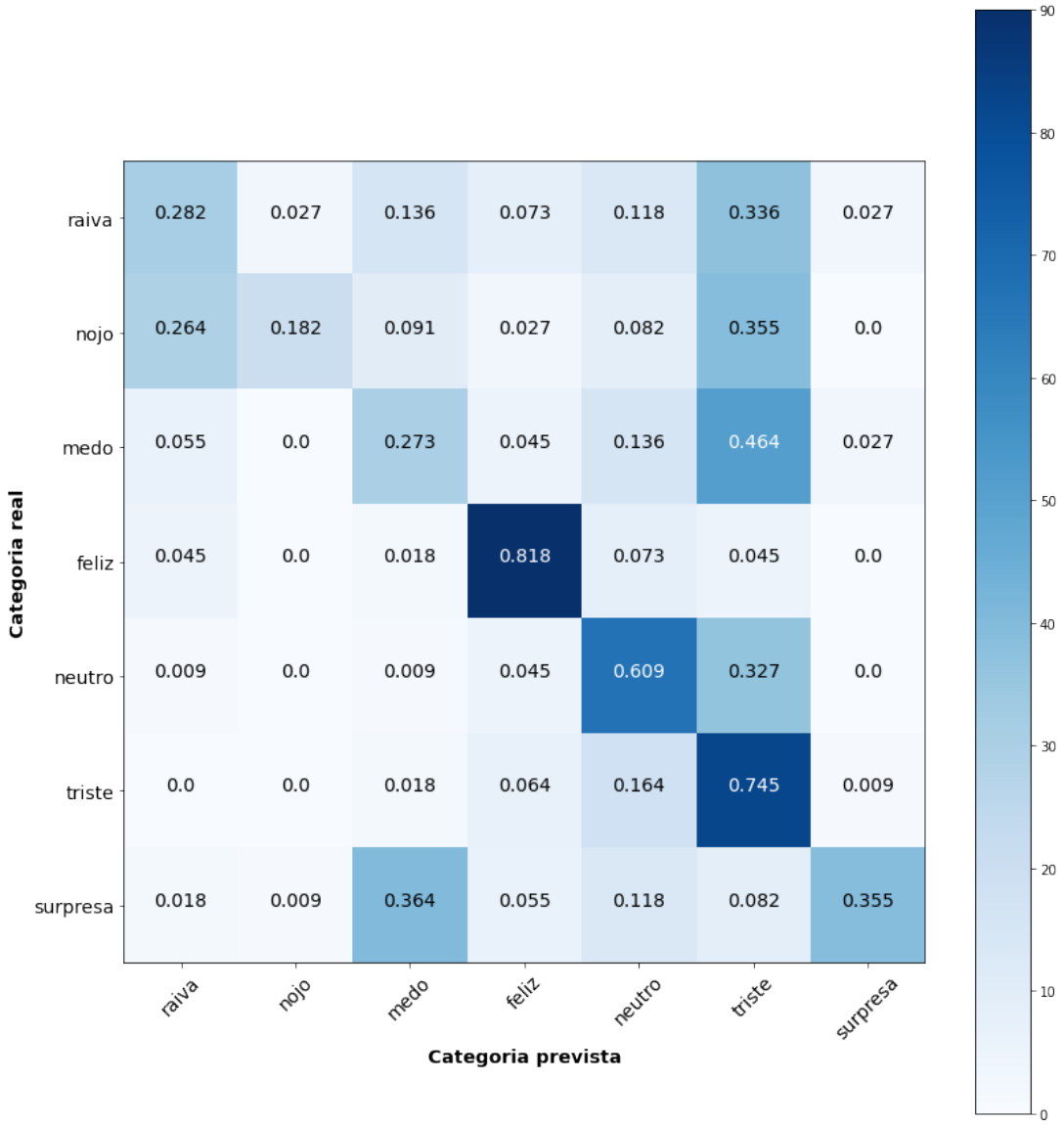

Figura 44 – Matriz de confusão da CNN utilizando as camadas da *MobileNetV2*

Validando este modelo na prática foram obtidos os melhores resultados, e por isso, este foi usado na implementação final no robô *NAO*.

4.3.2 Modelo com 4 camadas

Neste modelo os resultados obtidos foram de aproximadamente 62% de precisão utilizando o banco de imagens de validação.

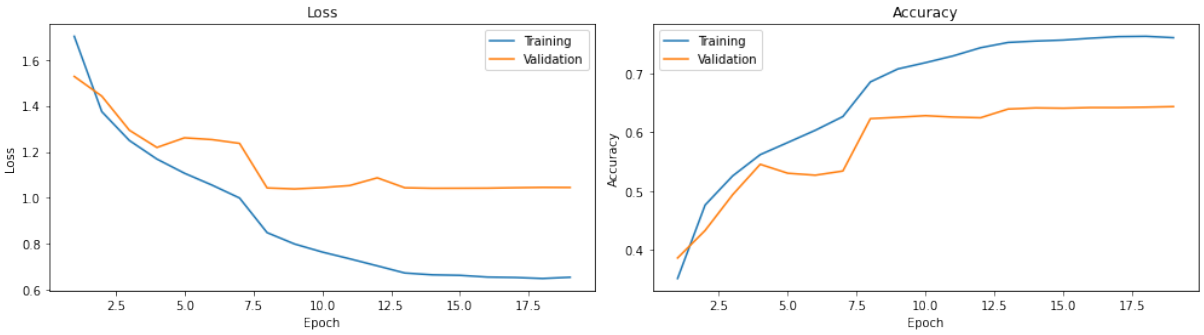

Figura 45 – Gráfico de acurácia e perda da CNN utilizando a arquitetura com 4 camadas

Ao observar os resultados na matriz de confusão na figura 46 percebe-se que os resultados foram melhores utilizando o banco de imagens para fazer a validação. Mas, ao utilizar a rede para gerar relatórios na prática os resultados não foram melhores do que utilizando transferência de conhecimento com a *MobileNetV2*. Não foi estudado o motivo, mas a várias diferenças que pode causar isto como por exemplo a extração das características dentro *MobileNetV2*, que por sua vez foi treinada com outro banco de imagens, o *ImageNet*.

Pode-se observar que também vários erros de classificação de medo e raiva, classificaram como triste. Além disso, mais de 20% dos erros classificação de nojo foram para a classe raiva.

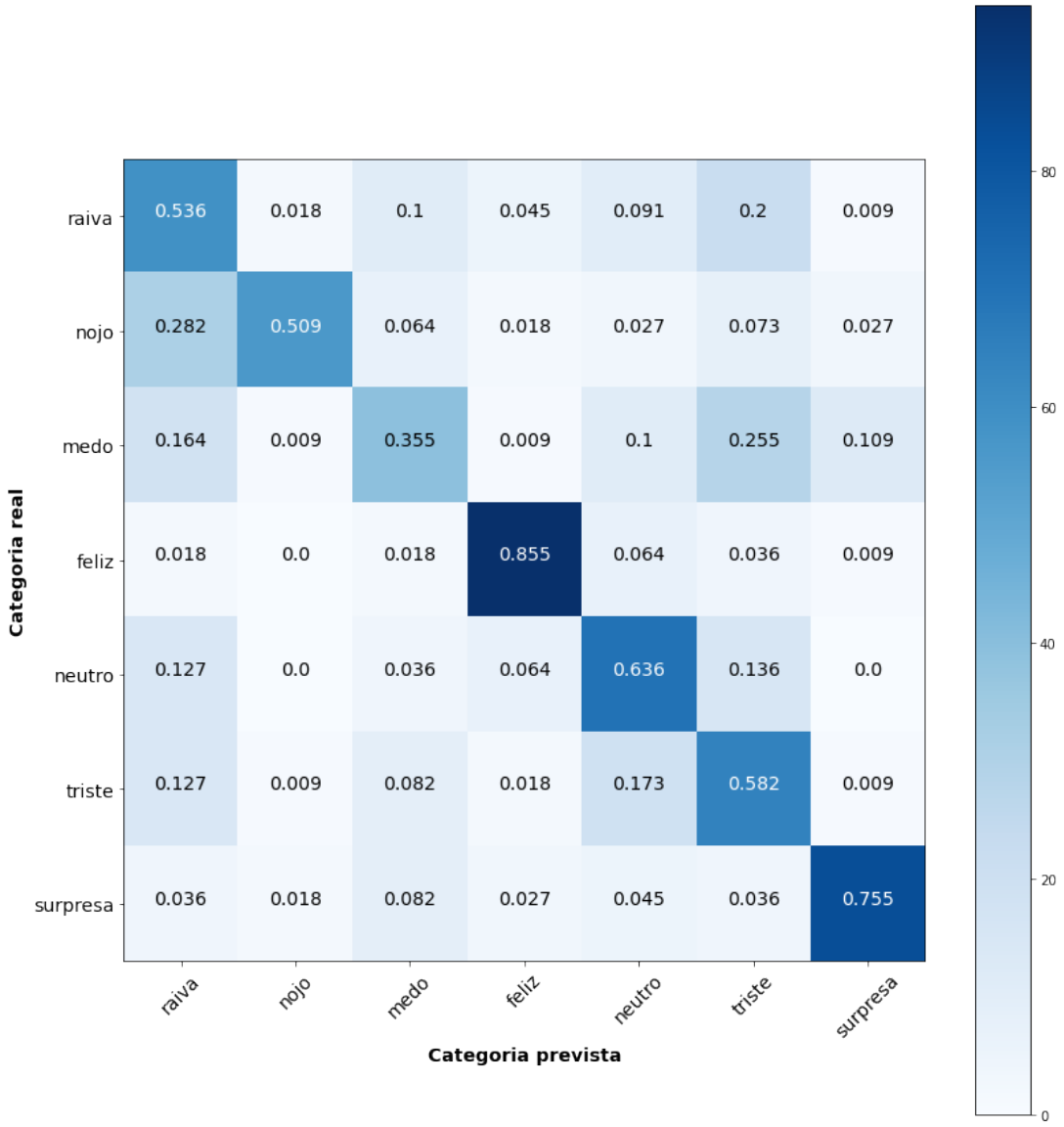

Figura 46 – Matriz de confusão da CNN utilizando a arquitetura com 4 camadas

4.4 Aplicação da rede neural no robô para gerar relatórios

Nesta sessão será verificado todos os treinamentos realizados, e aplicados diretamente ao robô *NAO* para analisar aquele que melhor se adaptou à câmera do robô.

Este procedimento terá a capacidade de realizar o reconhecimento de emoções a partir de vídeos curtos em torno de 20 a 30 segundos de vídeo, retiradas diretamente do robô, e em cada um deles o usuário estará realizando uma determinada emoção durante todo ele. Com isso, será apresentado ao classificador uma frequência de 3 imagens a cada segundo, e nisto, será apresentado 3 emoções por segundo.

Com a imagem de entrada no classificador informará a porcentagem de ser aquela

emoção variando de 0 a 100%, ou seja, se realizar uma entrada de uma imagem com uma pessoa alegre, o classificador informará uma certa porcentagem de ser aquela emoção, por exemplo, podendo apresentar 95% de alegria, 2% de tristeza e 3% de neutralidade.

A partir disso, é criado gráficos de demonstração das emoções recebidas do classificador de acordo com o tempo do vídeo, e assim, podemos verificar a emoção segundo a segundo, e gerando o relatório.

E, ao final, terá um vídeo mais longo com 2 minutos e 30 segundos com todas as expressões desenvolvida pelo usuário para gerar o relatório de tempo e analisar precisão do classificador.

E, todos os vídeos que foram utilizados para os testes e capturadas diretamente do *NAO*, podem ser vistos na íntegra pelo site do *Youtube* por este *LINK*(CASTRO, 2023).

4.4.1 Aplicação da Metodologia de 4 Camadas

Os primeiros resultados que aplicados com a metodologia de 4 camadas (DEBNATH et al., 2021), e com os vídeos capturadas do *NAO* usando somente os recurso de expressar uma única emoção, apesar de promissores nos valores teóricos, podemos avaliar diretamente ao *NAO*, e nisto, observemos:

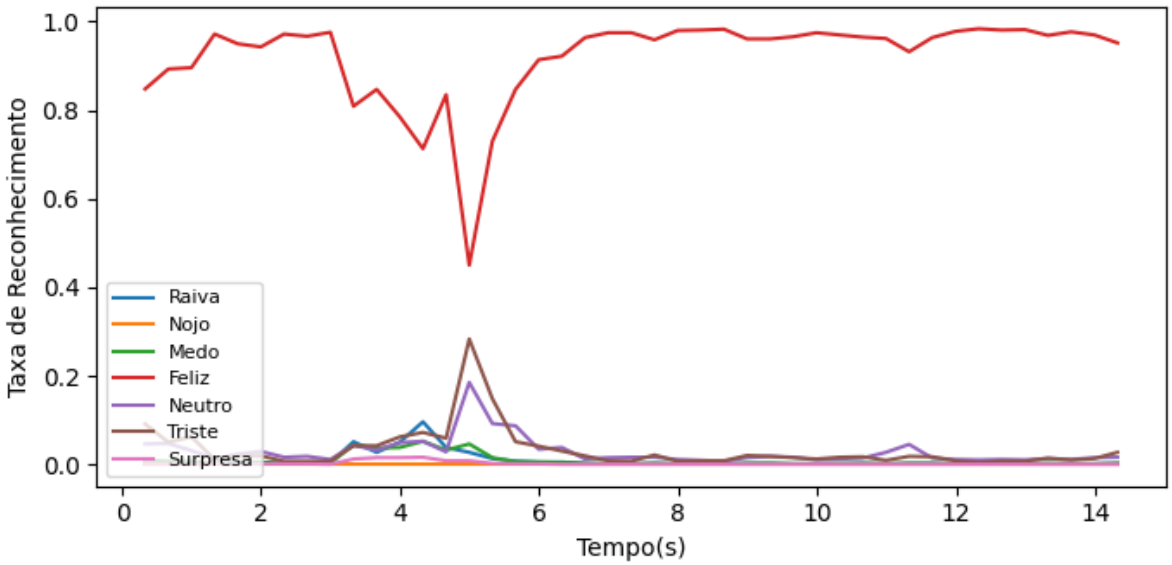

Figura 47 – Gráfico para Reconhecimento de Alegria com 4 camadas

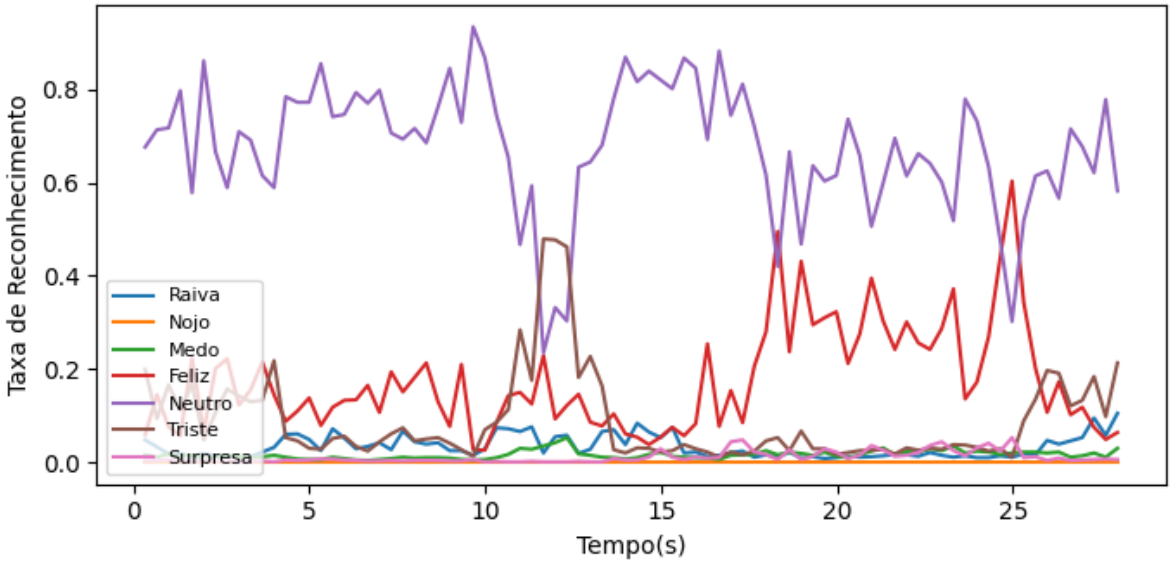

Figura 48 – Gráfico para Reconhecimento de Neutro com 4 camadas

Ambas as figuras 47 e 48 são, respectivamente, sobre o reconhecimento das emoções de alegria e de neutro, obtiveram um excelente reconhecimento. Por exemplo, a alegria não teve qualquer alteração de emoção no vídeo de teste, ou seja, 100% de precisão. Já para a emoção neutro obteve alguns momentos que recebeu com tristeza e com alegria.

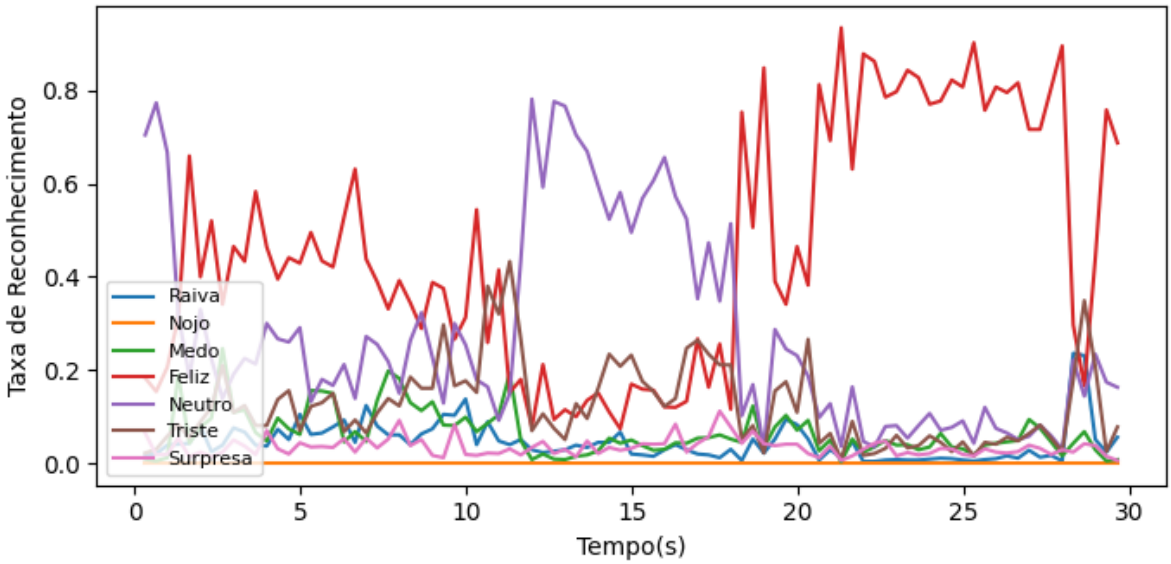

Figura 49 – Gráfico para Reconhecimento de Medo com 4 camadas

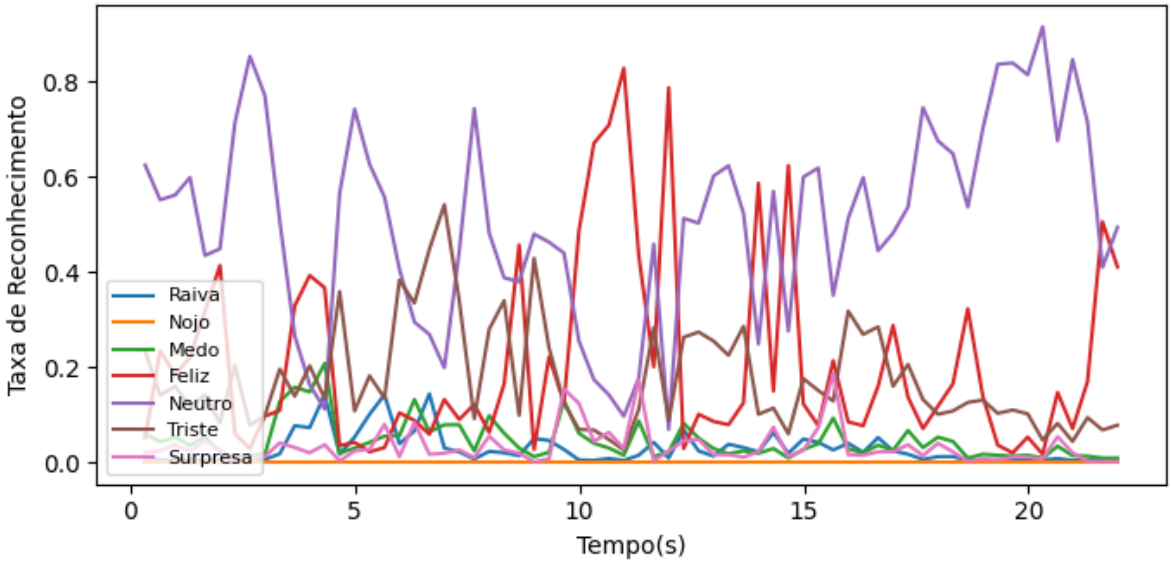

Figura 50 – Gráfico para Reconhecimento de Surpresa com 4 camadas

Porém, para as figuras 49 e 50 as emoções de medo e de surpresa não obtiveram nenhum êxito no reconhecimento, reduzindo, e muito, a precisão da rede neural, e por isso, seria necessária uma metodologia ainda melhor para maior precisão.

Os dados restantes das outras emoções desta metodologia estão no anexo A.

4.4.2 Aplicação da Metodologia de 4 Camadas Com filtro *Keypoints*

Ao aplicar os resultados da rede treinada com a metodologia do filtro de *Keypoints* não foram muito proveitosos. Se avaliarmos vídeos retirados do *NAO* as melhores percepções temos com raiva e alegria, como mostrado a seguir:

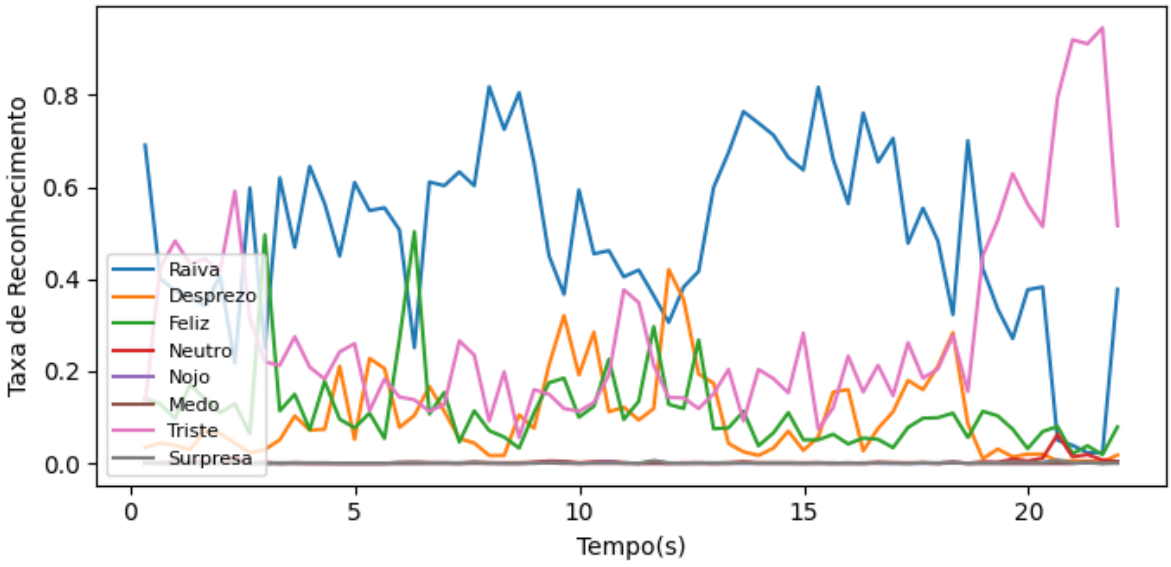

Figura 51 – Gráfico para Reconhecimento de Raiva com *Keypoints*

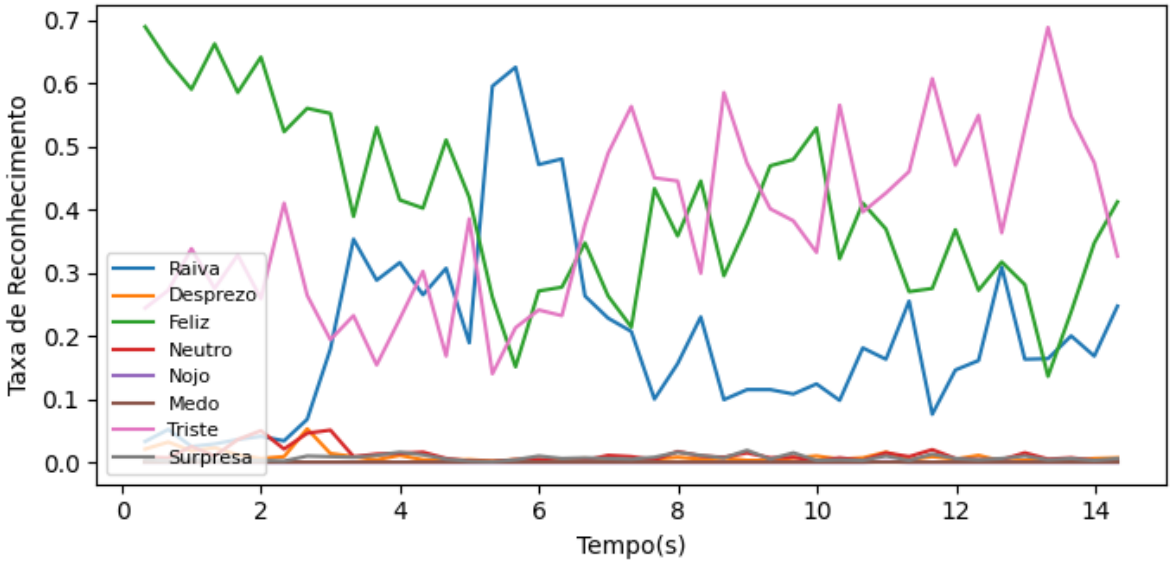

Figura 52 – Gráfico para Reconhecimento de Alegria com *Keypoints*

Estas duas emoções são possíveis avaliar que as figuras 51 e 52 são bem expressivos os reconhecimentos. Mas, ao avaliar a emoção de alegria, se realizar uma expressão com o sorriso com os lábios bem aberto e deixando os dentes bem amostra tem uma excelente precisão, mas se captar o oposto cai bruscamente a precisão.

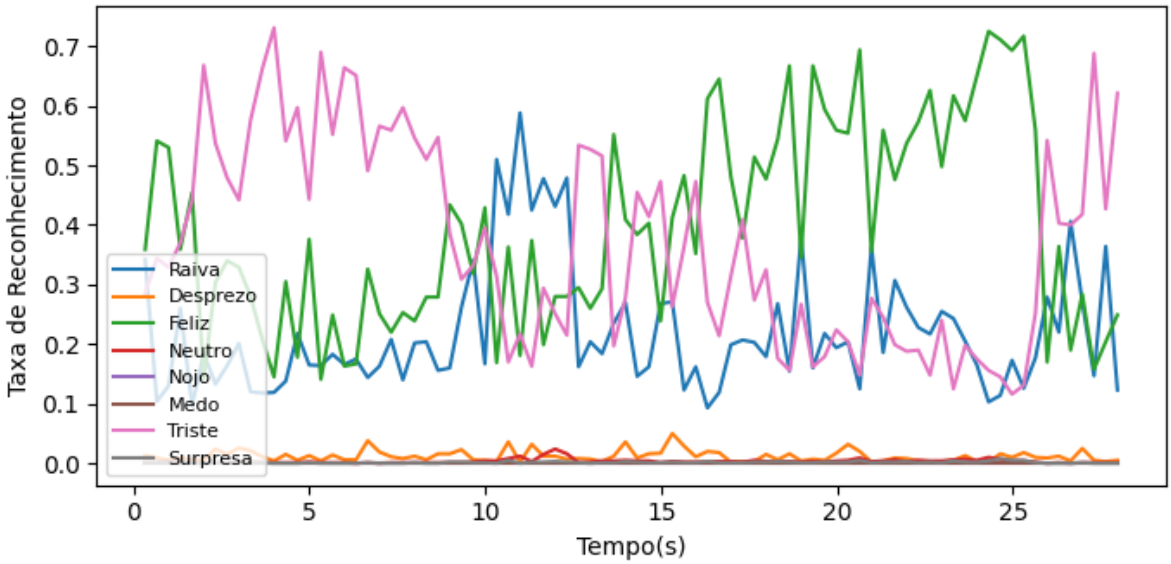

Figura 53 – Gráfico para Reconhecimento de Neutro com *Keypoints*

Já, ao observar, a figura 53 é notável que não houve em nenhum momento a captura da própria emoção testada, mostrando ainda mais que o uso do deste filtro não foi a melhor estratégia de uso para análise.

Para o restante das emoções é encontrada no anexo A.

4.4.3 Aplicação da Metodologia *MobileNet*

As aplicações com a metodologia do *MobileNet* diretamente no *NAO* foi a melhor que se adaptou à plataforma. Por isso analisaremos mais a fundo cada emoção treinada. Deste momento, será mostrado o resultado de cada uma das emoções individualmente para que possamos verificar a capacidade da rede para cada expressão do usuário.

Nesta etapa também fornecerá tabelas de reconhecimento da detecção da emoção. Ao aplicar o vídeo de expressão de somente alegria serão em torno de 50 imagens, e assim, verificar o quantitativo de imagens que reconheceu de alegria e quanto das outras emoções para termos um paralelo de precisão de acerto para aquela emoção em específico.

4.4.3.1 Raiva

Primeiramente, o vídeo que faremos a analise terá cerca de 21 segundos com faces de apresentando raiva em diferentes posições, e obtemos o seguinte fluxo:

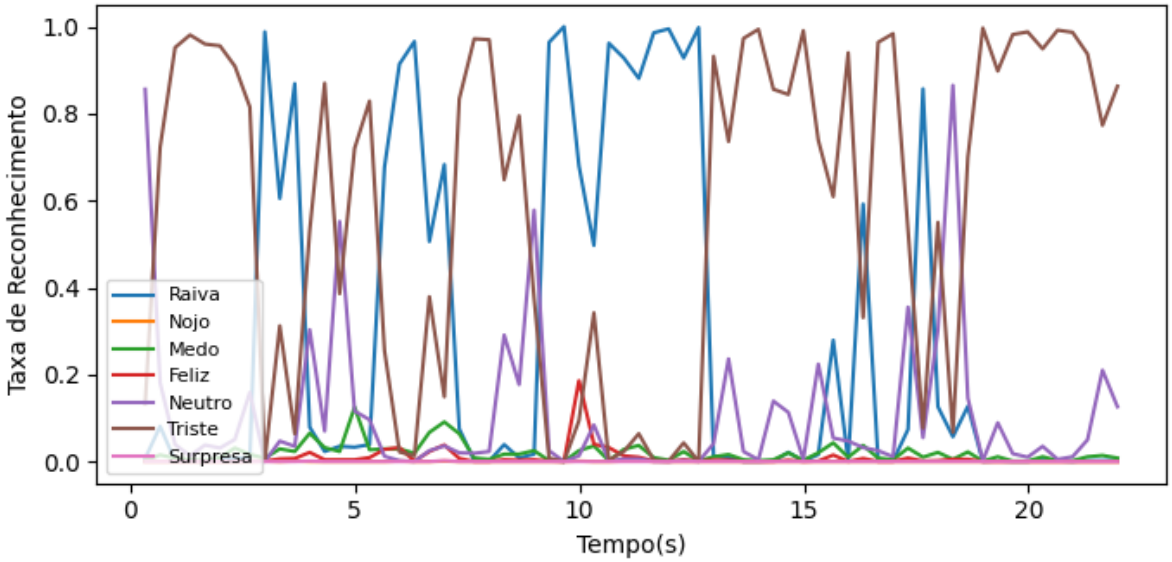

Figura 54 – Fluxo de reconhecimento para Raiva

Como é observável na figura 54 no período em questão são identificados a emoção de raiva algumas vezes, porém quando é identificado a porcentagem muito grande de acerto, enquanto que para o restante os que apresentaram diferente de raiva com uma porcentagem bem baixa.

Tabela 1 – Quantidade de Imagens por Emoção - Raiva

Emoções	Quant. Imagens
Raiva	21
Nojo	0
Medo	0
Alegria	0
Neutro	4
Triste	41
Surpresa	0
Total	66

Já ao vermos fria mente todos as imagens recuperadas a rede apresenta cerca de 31,82% de acerto, mas levando em consideração a figura 54 podemos avaliar que nesta porcentagem oscila bastante entre as emoções de tristeza e de raiva, desta forma, ao aplicar um filtro ou um *buffer* das probabilidades, podemos melhorar ainda mais este reconhecimento tornando-o ainda mais preciso.

4.4.3.2 Nojo

Neste ponto será analisado para a emoção nojo, a partir de um vídeo com cerca de 18 segundos de expressões de nojo, que apesar de todos os problemas apontados anteriormente, esta expressão seria muito complicada para que a rede recuperasse esta emoção, mesmo usando o próprio banco de dados. Desta forma, temos:

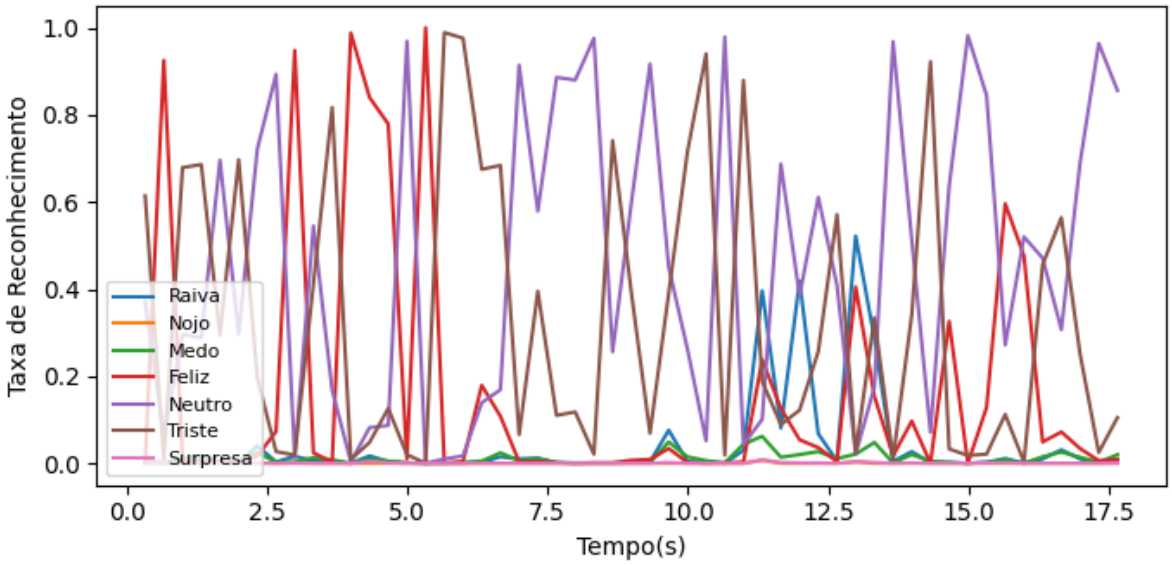

Figura 55 – Fluxo de reconhecimento para Nojo

Como já previsto, a emoção de nojo não apresentou nenhum reconhecimento, e isto, foi testado com alguns minutos de testes não ocorrendo nenhuma ocorrência de tal emoção, além de, apresentar muitas oscilações com relação às outras emoções variando entre neutro e triste.

Tabela 2 – Quantidade de Imagens por Emoção - Nojo

Emoções	Quant. Imagens
Raiva	3
Nojo	0
Medo	0
Alegria	7
Neutro	26
Triste	17
Surpresa	0
Total	53

E, sobre isto, como foi previsto é muito complexo reproduzir a emoção no robô para que a rede neural fosse ser interpretada. Por isso, é a única emoção que não dá para ser avaliada direto na plataforma.

4.4.3.3 Medo

Nesta sub sessão será avaliada a emoção de Medo direto da plataforma, com um total de 30 segundo de vídeo desenvolvendo esta expressão, conquistando os seguintes resultados:

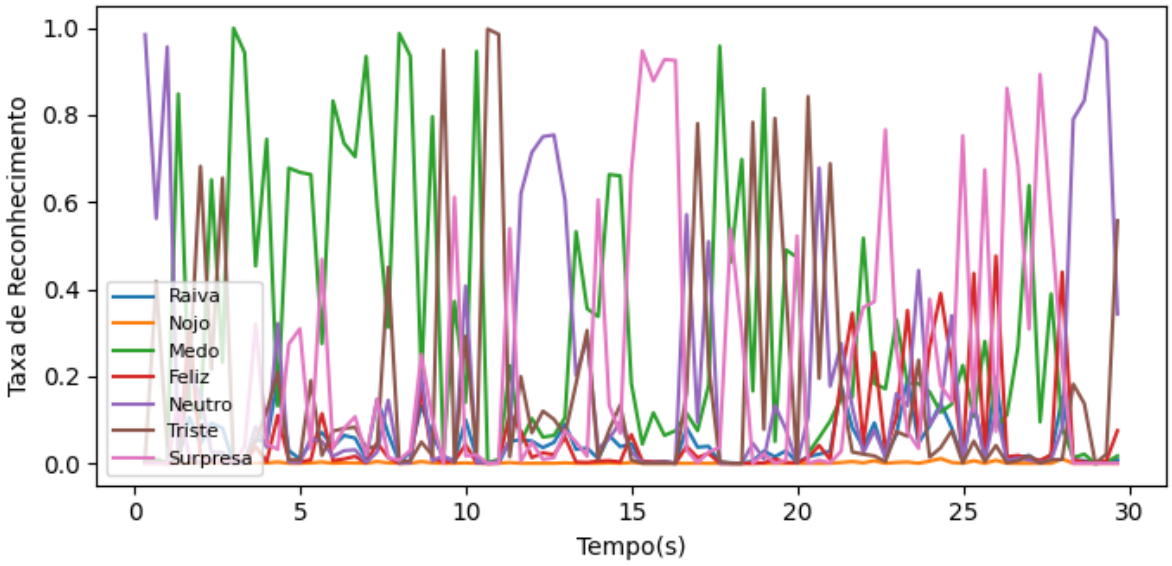

Figura 56 – Fluxo de reconhecimento para Medo

Para análise do medo ocorreu muita oscilação no decorrer de vídeo ficando, principalmente, entre medo e surpresa, e nisto para a rede tem uma dificuldade para diferenciar entre elas. Mesmo para nós, seres humanos, ambas as expressões ainda são complexas de serem analisadas, pois ambas ainda apresentam uma característica de olhos bem abertos e lábios também abertos.

Tabela 3 – Quantidade de Imagens por Emoção - Medo

Emoções	Quant. Imagens
Raiva	0
Nojo	0
Medo	29
Alegria	7
Neutro	20
Triste	12
Surpresa	21
Total	89

Apesar do medo e surpresa se apresentarem em boa quantidade, o neutro também se encontra logo em seguida, em termos de imagens, ficando principalmente, entre as 3 emoções no vídeo demonstrado. E, para a devida emoção, temos uma precisão de 32,58% de acerto.

4.4.3.4 Alegria

Em seguida, a emoção alegria será analisada, apresentando cerca de 15 segundo de vídeo com este sentimento, será nela que terá algumas expressões com sorriso com lábios bem abertos, e com lábios fechados e com posições da face diferentes, e temos:

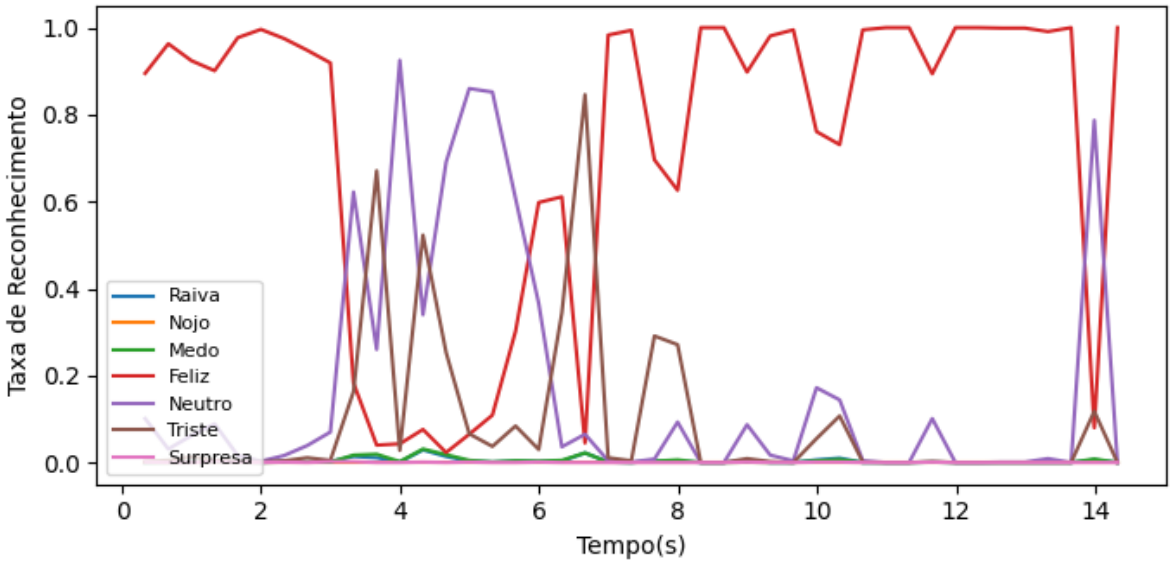

Figura 57 – Fluxo de reconhecimento para Alegria

Ao analisar o gráfico 57 é uma das emoções mais precisas de serem reconhecidas pela rede neural, principalmente, que é extremamente marcante o uso dos dentes no sorriso, e nisto, a maioria dos modelos testados anteriormente apresentou, o mínimo que seja dos dentes na imagem, já era automaticamente reconhecido como alegria.

Tabela 4 – Quantidade de Imagens por Emoção - Alegria

Emoções	Quant. Imagens
Raiva	0
Nojo	0
Medo	0
Alegria	33
Neutro	7
Triste	3
Surpresa	0
Total	43

Ao verificar a tabela 4 é possível ser notado que quase não há erros na análise de emoção com quase 76,75% de acerto.

4.4.3.5 Neutro

O próximo da lista, com um total de 28 segundos de vídeo, verificaremos a expressão de neutro, conquistando os seguintes dados:

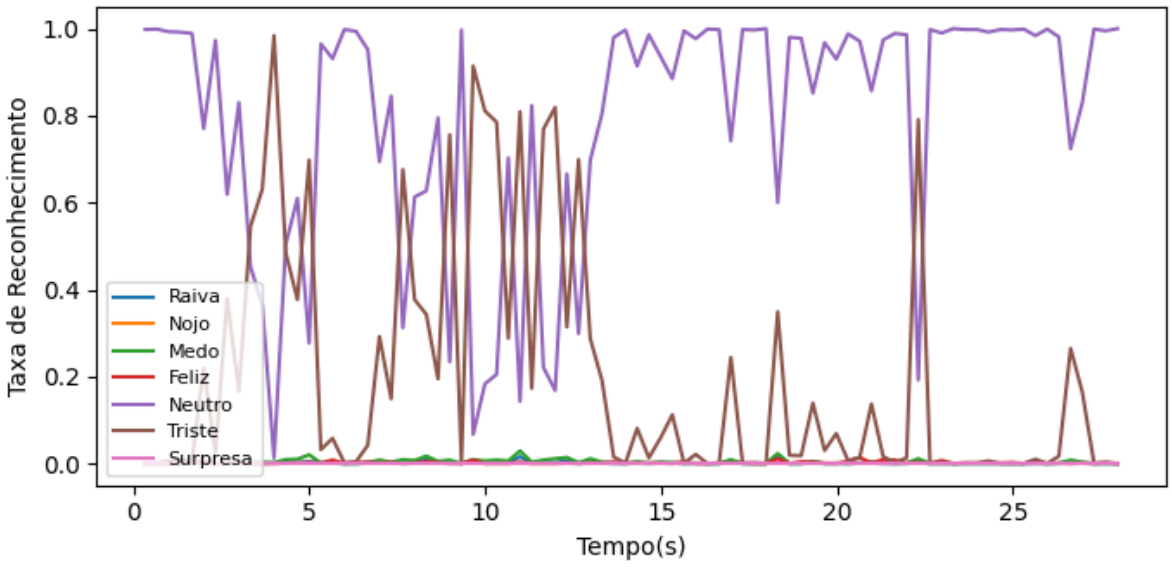

Figura 58 – Fluxo de reconhecimento para Neutro

Assim, como na análise da alegria, é possível notar na figura 58 é incrível o reconhecimento desta emoção em relação ao que o usuário apresenta a plataforma, havendo pequenas alterações principalmente, na alteração dos olhos apontados para baixo.

Tabela 5 – Quantidade de Imagens por Emoção - Neutro

Emoções	Quant. Imagens
Raiva	0
Nojo	0
Medo	0
Alegria	0
Neutro	70
Triste	14
Surpresa	0
Total	84

Com os dados acima, é notável, o reconhecimento para esta expressão apresenta por volta de 83,33% de precisão, ou seja, uma acurácia excelente para estes dados.

4.4.3.6 Tristeza

Após isto, seguimos para a expressão de tristeza, com 27 segundo de vídeo apresentando esta emoção, é possível verificar o seguinte:

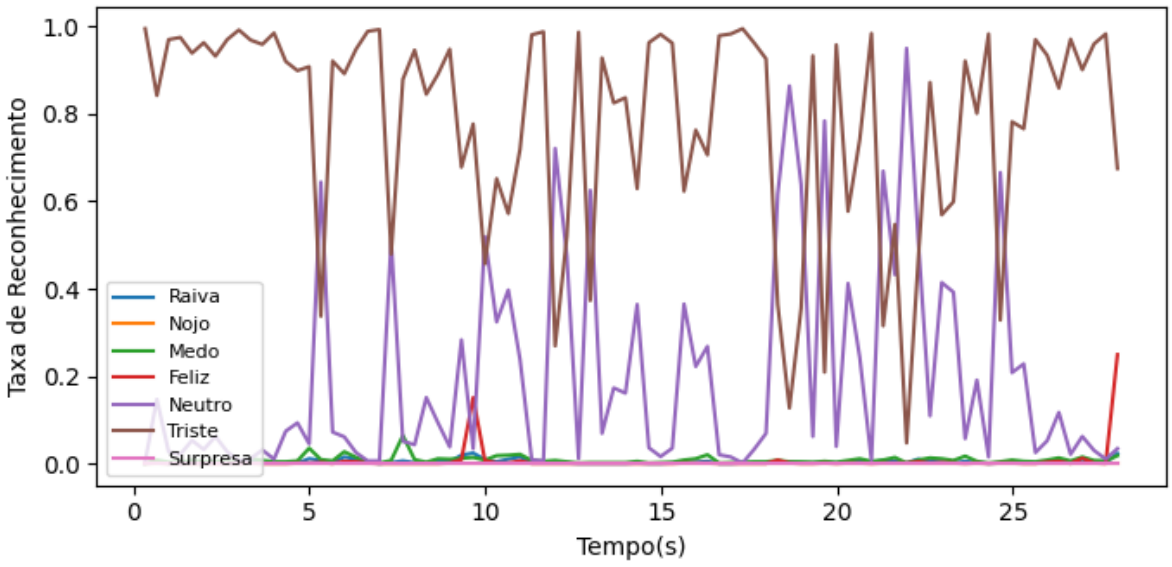

Figura 59 – Fluxo de reconhecimento para Tristeza

Ocorrendo em suma maioria, a rede reconheceu a expressão de tristeza, salva as ressalvas em que apresenta alguns dados com neutro, surgindo nos momentos em que o usuário muda o olhar para frente.

Tabela 6 – Quantidade de Imagens por Emoção - Tristeza

Emoções	Quant. Imagens
Raiva	0
Nojo	0
Medo	0
Alegria	0
Neutro	13
Triste	71
Surpresa	0
Total	84

Com uma precisão de 84,52% das imagens capturadas, é notável perceber o quão preciso temos para esta emoção.

4.4.3.7 Surpresa

Para a última emoção, seguimos para análise da expressão surpresa, verificando:

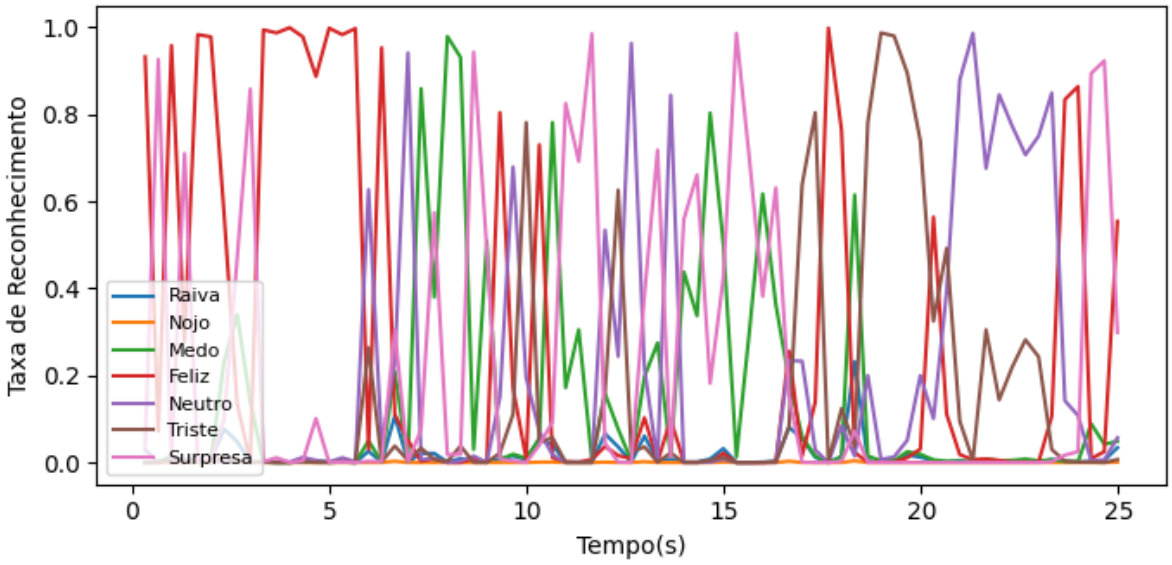

Figura 60 – Fluxo de reconhecimento para Surpresa

Com baste oscilação, encontramos bastantes pontos rosas(surpresa) com pico, mas também está misturada com alegria, neutro e medo (como visto anteriormente este especificamente). Desta forma, para ter uma avaliação mais precisa olhemos a tabela a seguir:

Tabela 7 – Quantidade de Imagens por Emoção - Surpresa

Emoções	Quant. Imagens
Raiva	0
Nojo	0
Medo	9
Alegria	23
Neutro	14
Triste	10
Surpresa	19
Total	75

Com um total de 25,33%, se provou uma expressão bastante imprecisa de se reconhecer.

4.4.3.8 Linha do Tempo Com Todas as Expressões

Por fim, faremos uma análise de todas as emoções sendo expressas em um único vídeo de 2 minuto de 26 segundos, assim, podemos dar uma avaliada em tempo real de alterações:

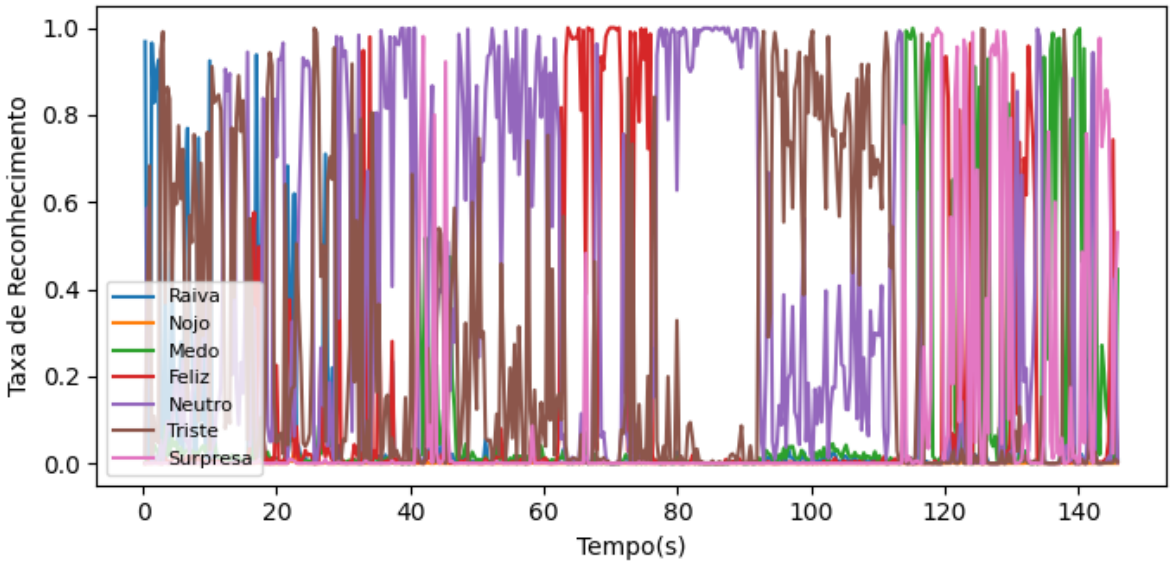

Figura 61 – Fluxo de reconhecimento com todas as emoções

Tabela 8 – Quantidade de Imagens por Emoção - Surpresa

Emoções	Quant. Imagens
Raiva	16
Nojo	0
Medo	33
Alegria	58
Neutro	167
Triste	124
Surpresa	40
Total	438

Ao verificar a figura 61 podemos perceber que com o decorrer do tempo, o processo de reconhecimento tem suas definições de acordo exato com cada emoção, observamos as emoções bem marcantes de neutro, alegria e tristeza variando principalmente, nas regiões entre 50 a 110 segundos. A confusão estabelecida de surpresa com medo nas ultimas faixas. Apesar de, não muito presente no início, a raiva apresenta suas marcas de presença, ou seja, aplicando algum tipo de filtro ou verificação das probabilidade da desta emoção tem a maior assertividade. E, por fim, o nojo não observado em nenhum momento durante o vídeo.

4.5 Discussão

Foi obtido resultados satisfatórios na identificação de emoções pelo robô através dos recursos disponíveis. O melhor resultado apresentou 66% de acerto utilizando a *MobileNet*, o que é um resultado dentro do esperado, pois em competições profissionais no *Kaggle* e em artigos bem conceituados os resultados são de 65% $\pm$ 5%. Entretanto algumas emoções apresentaram taxas de acerto baixas devido à dificuldade do problema, mas isso pode ser melhorado combinando a análise de outras fontes, como a voz.

A adição de fontes como a voz, que reflete as emoções através da sua frequência, e ainda por visão computacional, a análise do contexto e postura corporal, pode resultar em resultados mais precisos e taxas de acerto melhores.

É possível ainda melhorar o reconhecimento de expressões com técnicas de processamento de imagens como *image augmentation*, que geraria outras imagens através de rotações, mudança de cores e de escala. Outros bancos de imagens também poderiam ser usados e mesclados de forma a ter imagens de diferentes padrões possibilitando a criação de um classificador mais robusto.

Além disto, é possível fazer uma análise de gráficos como 47 que mostra a taxa de reconhecimento que seria uma pseudo probabilidade de cada uma daquelas emoções naquele momento. Isso permite usar recursos como a análise de qual das emoções possui a maior área em determinado espaço de tempo, ou mesmo de analisar quando é possível surgir uma intersecção entre duas diferentes emoções.

5 Conclusões

Este projeto apresentou uma construção de rede neural utilizando várias técnicas de aprendizado de máquina com objetivo de reconhecer emoções a partir de expressões faciais. E, ao final de desenvolvimento, foi integrada à plataforma robótica *NAO* para acrescentar mais uma nova habilidade ao robô.

Durante o desenvolvimento, testou-se à extração das características principais da imagem antes de fornecer à entrada da rede neural, mas, no processo de treinamento é possível perceber que os resultados não foram satisfatórios, e ao aplicarmos à plataforma se apresentou o previsto anteriormente, uma baixa precisão. Desta forma, se fez necessário aplicar uma nova estratégia.

Uma das estratégias aplicadas foi a transferência de conhecimento a partir de pesos da *ImageNet*. Esta foi a técnica que mostrou os melhores resultados com uma precisão de 66% no banco de imagens de validação. E, enquanto que no robô, apontou bons resultados.

Os melhores resultados vieram principalmente das emoções da alegria, da neutralidade e da tristeza, pois são elas que melhor se desenvolveram tanto do banco de imagens de validação quanto aplicadas ao robô chegando em torno de 80% de acerto.

Já, com a emoção de raiva, mesmo com as trocas de reconhecimento entre raiva e tristeza, é possível perceber que no momento desta emoção se fez presente no reconhecimento, ou seja, se aplicar um filtro a partir de sua taxa obtêm uma precisão ainda maior.

Mesmo com emoções tão distintas é notável que algumas delas ainda são difíceis de serem diferenciadas pelo aprendizado de máquina, pois, alguns deles se mostram muito semelhantes, e as diferenças delas não são tão nítidas pelo banco de imagens, por exemplo, as emoções de medo e surpresa tem uma equivalência na região da boca, mas as sobrancelhas e a testa são pouco nítidas, dificultando o reconhecimento.

Porém, a emoção que não se obteve progresso no reconhecimento foi a de nojo, pois, com baixo número de dados tanto de treino quanto de validação tornando muito difícil para a rede neural ter uma acurácia de validação muita boa para este caso, tornando ainda pior na plataforma.

Por fim, os estudos de rede neural em conjunto das plataformas robótica vêm se tornando cada vez mais difundido e melhorando ainda mais as técnicas para tornar mais eficiente no classificação e mais assertivo em seus resultados.

5.1 Trabalhos Futuros

Para rede neural ter maiores taxas se faz necessário unificar os bancos ter maiores dados de validação, por exemplo pegar todos os dados do *FER2013*, do *AffectNet* e outros banco de imagens, tornar em uma só para que a rede possa melhor ainda mais os pesos.

Como as taxas não são 100% precisas é possível utilizar outros recursos de validação, por exemplo unificar estudos que trabalham com reconhecimento de emoções que usam infravermelho, para avaliar pulsação sanguínea ou trabalho que fazem analise de timbre vocal, pela voz é possível validar algumas emoções.

E, por fim, um trabalho futuro, com este projeto, é de direcionar uma sessão de acordo com a necessidade. Como criar uma ordem de perguntas, que possam direcionar diferentes emoções, feitas pelo próprio *NAO* e fazer uma análise emocional a partir delas. Ou, até mesmo, realizar sessões terapêuticas, que com o decorrer do tempo, validar diferentes tipos de emoções para auxiliar diagnósticos de transtornos *déficit* de atenção, hiperatividade, ou até mesmo para doenças mentais como ansiedade ou depressão.

Referências

CASTRO, E. d. V. d. **Trabalho de Graduação - Reconhecimento de Emoções a partir de Expressões Faciais pelo Robô NAO**. https://youtube.com/playlist?list=PLbvI1Fkd3fTP5C5SdxrCUchJS6S2AZzPv – acesso em 29/01. 2023. Citado na p. 54.

CONEXÃO SESSÃO TCP-IP - NAOQI. **NAOqi - ALTabletService**. http://doc.aldebaran.com/2-8/naoqi/core/altabletservice-api.html – acesso em 19. 2023. Citado na p. 34.

DALOSTO, F. d. M. IMPLEMENTAÇÃO DE INTERAÇÃO HUMANO-ROBÔ POR MEIO DE VOZ NA PLATAFORMA NAO., p. 56, 2019. DOI: tps://bdm.unb.br/handle/10483/28465. Citado na p. 28.

DEBNATH, T.; REZA, M.; RAHMAN, A.; S. BAND, S.; ALINEJAD-ROKNY, H. Four-layer Convnet to Facial Emotion Recognition With Minimal Epochs and the Significance of Data Diversity, mai. 2021. DOI: https://doi.org/10.1038/s41598-022-11173-0. Citado nas pp. 41, 42, 50, 54.

DICIO, DICIONÁRIO ONLINE DE PORTUGUÊS. **Emoção**. https://www.dicio.com.br/emocao/ – acesso em 28/01. 2023. Citado na p. 19.

DOCUMENTAÇÃO DO *MEDIAPIPE*. ***Mediapipe***. https://pypi.org/project/mediapipe/ – acesso em 19. 2022. Citado na p. 36.

DOCUMENTAÇÃO *SOCKET* - PYHTON 2.7. ***Socket* - Python 2.7**. https://docs.python.org/2/library/socket.html – acesso em 19. 2020. Citado na p. 34.

DOCUMENTAÇÃO *SOCKET* - PYHTON 3.11. ***Socket* - Python 3.11**. https://docs.python.org/3/library/socket.html – acesso em 19. 2023. Citado na p. 34.

DOCUMETAÇÃO NAOQI. **NAOqi - Developer guide**. http://doc.aldebaran.com/2-8/index_dev_guide.html – acesso em 16. 2023. Citado na p. 33.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2. ed. New York: Wiley, 2001. ISBN 978-0-471-05669-0. Citado na p. 31.

FREITAS-MAGALHÃES, A. **A Psicologia das Emoções - O Fascínio do Rosto Humano**. 1. ed.: Escrytos|Ed. Autor, set. 2013. v. 1. Citado na p. 20.

GIBSON, G. M.; JOHNSON, S. D.; PADGETT1, M. J. Single-pixel imaging 12 years on: a review, p. 19, 2020. DOI: https://doi.org/10.1364/OE.403195. Citado na p. 27.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2016. http://www.deeplearningbook.org. Citado nas pp. 30, 32.

HENRIQUE, F. R.; TOMAZIO, N. B.; ROSA, R. G. T.; SOUZA, A. M. d.; D'ALMEIDA, C. d. P.; SCIUTI, L. F.; GARCIA, M. R.; BONI, L. D. Luz à primeira vista: um programa de atividades para o ensino de óptica a partir de cores, p. 7, 2019. DOI: http://dx.doi.org/10.1590/1806-9126-RBEF-2018-0223. Citado na p. 27.

HTWK ROBOTS. ***Robocup 2019 SPL Final - HTWK vs. B-Human***. https://www.youtube.com/watch?v=4_BWQl91p9Y&t=921s&ab_channel=HTWKRobots – acesso em 29/01. 2019. Citado na p. 26.

INFOWESTER. **Resoluções HD, full HD, 4K, 8K e mais**. https://www.infowester.com/resolucoes.php – acesso em 29/01. 2020. Citado na p. 27.

KERAS. **EarlyStopping**. https://keras.io/api/callbacks/early_stopping/ – acesso em 01/02. 2023a. Citado na p. 43.

KERAS. **ReduceLROnPlateau**. https://keras.io/api/callbacks/reduce_lr_on_plateau/ – acesso em 01/02. 2023b. Citado na p. 43.

KHANZADA, A.; BAI, C.; CELEPCIKAY, F. T. **Facial Expression Recognition with Deep Learning**. arXiv, 2020. DOI: 10.48550/ARXIV.2004.11823. Disponível em: <https://arxiv.org/abs/2004.11823>. Citado na p. 40.

MAHOOR, M. H. **AffectNet**. http://mohammadmahoor.com/affectnet/ – acesso em 29/01. 2011. Citado nas pp. 21–24, 36–39.

MAPEAMENTO DOS *KEYPOINTS*. ***Mediapipe - Keypoints***. https://github.com/google/mediapipe/blob/master/mediapipe/modules/face_geometry/data/canonical_face_model_uv_visualization.png – acesso em 26. 2022. Citado na p. 39.

PALESTRA, G.; PETTINICCHIO, A.; DEL COCO, M.; CARCAGN, P.; LEO, M.; DISTANTE, C. Improved Performance in Facial Expression Recognition Using 32 Geometric Features, p. 11, 2015. DOI: http://dx.doi.org/10.1007/978-3-319-23234-848. Citado na p. 30.

RAMOS, A. P.; BORTAGARAI, F. M. A Comunicação Não-Verbal na Área da Saúde, p. 7, jan. 2012. DOI: http://dx.doi.org/10.1590/S1516-18462011005000067. Citado na p. 19.

SANDLER, M.; HOWARD, A. G.; ZHU, M.; ZHMOGINOV, A.; CHEN, L. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. **CoRR**, abs/1801.04381, 2018. arXiv: 1801.04381. Disponível em: <http://arxiv.org/abs/1801.04381>. Citado na p. 43.

SANTOS, F. A. D. COMUNICAÇÃO NÃO VERBAL: IDENTIFICAÇÃO DE EMOÇÕES ATRAVÉS DE EXPRESSÕES FACIAIS NA PRÁTICA DA PSICOLOGIA CLÍNICA, p. 83, 2017. DOI: http://repositorio.unesc.net/handle/1/5692. Citado nas pp. 21–24.

SANTOS, J. A. P. TEABOT – Robô para treinamento de expressões faciais emocionais para pessoas com Transtorno do Espectro do Autismo, p. 88, 2019. DOI: https://repository.ufrpe.br/handle/123456789/1439. Citado nas pp. 16, 26.

SANTOS, V. **Metaforando**. https://www.youtube.com/@Metaforando/featured – acesso em 29/01. 2016. Citado na p. 20.

SOFBANK GROUP - ALDEBARAN. **Robot NAO**. https://www.aldebaran.com/en/nao – acesso em 29/01. 2023. Citado na p. 25.

TAKAHASHI, N. M. ESTRUTURA DE PADRÕES DE INTERAÇÃO HUMANO-ROBÔ PARA APLICAÇÕES AUTÔNOMAS E INTELIGENTES, p. 160, 2018. DOI: https://doi.org/10.31414/EE.2018.D.129665. Citado na p. 26.

Apêndices

APÊNDICE A – Glossário

- Classes - Categorias
- *Convolutional Neural Network (CNN)* - Rede Neural Convolucional (RNC)
- *Dataset* - Banco de dados ou banco de imagens.
- *Learning rate* - taxa de aprendizagem
- *Machine learning* - Aprendizado de máquina.

APÊNDICE B – Códigos de programação

B.1 Código de conexão com a Plataforma *NAO* de forma externa

Código B.1 – Código *Python 2.7 NAOqi*

```
import qi
import argparse
import sys
from naoqi import ALProxy


NAOIP = '192.168.1.71'
NAOPORT = 9559

def main():

    session = qi.Session()

    session.connect("tcp://" + NAOIP + ":" + str(NAOPORT))

    alVideo = ALProxy("ALVideoDevice", NAOIP, NAOPORT)

    video_service = session.service("ALVideoDevice")
    resolution = vision_definitions.kVGA
    colorSpace = vision_definitions.kRGBColorSpace
    fps = 30
    SUBSCRIBE_NAME = "NAO_CAM"

    nameId = video_service.subscribe(SUBSCRIBE_NAME, resolution,
       colorSpace, fps)
```

B.2 Código de conexão de rede local para enviar imagens - *Python 2.7*

Código B.2 – Código Socket - *Python 2.7 NAOqi*

```
import socket, cv2, math
import numpy as np
import pickle
```

```
4  from PIL import Image
5  import qi
6  import vision_definitions
7
8
9  host = '127.0.0.1'
10 portHost = 8081
11 NAOIP = '192.168.1.71'
12 NAOPORT = 9559
13
14 BUFF_SIZE = 65536
15
16 server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
17 server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_RCVBUF,
       BUFF_SIZE)
18
19 session = qi.Session()
20 session.connect("tcp://" + NAOIP + ":" + str(NAOPORT))
21
22 video_service = session.service("ALVideoDevice")
23 resolution = vision_definitions.kVGA
24 colorSpace = vision_definitions.kRGBColorSpace
25 fps = 30
26 SUBSCRIBE_NAME = "NAO_CAM"
27
28 nameId = video_service.subscribe(SUBSCRIBE_NAME, resolution,
       colorSpace, fps)
29
30
31 server_socket.bind((host, portHost))
32 while True:
33     msg, addr = server_socket.recvfrom(BUFF_SIZE)
34     print('GOT connection from ', addr)
35
36     while True:
37         camNAO = video_service.getImageRemote(nameId)
38         if camNAO is None:
39             print('Erro: NAO is none!')
40             break
41
42         WIDTH_INDEX = 0
43         HEIGTH_INDEX = 1
44         IMAGE_ARRAY_INDEX = 6
45
46         imageWidth = camNAO[WIDTH_INDEX]
47         imageHeight = camNAO[HEIGTH_INDEX]
48         array = camNAO[IMAGE_ARRAY_INDEX]
49
50         frame = str(bytearray(array))
51
52         frame = Image.frombytes("RGB", (imageWidth, imageHeight),
               frame, 'raw', 'BGR', 0, -1)
```

```
53          frame = np.asarray(frame)
54
55          server_socket.sendto(pickle.dumps(frame_info), addr)
56
57          encode, buffer = cv2.imencode('.jpg', frame,
               [cv2.IMWRITE_JPEG_QUALITY, 80])
58
59          if encode:
60              buffer = buffer.tobytes()
61              buffer_size = len(buffer)
62
63          num_packs = 1
64          if buffer_size > BUFF_SIZE:
65              num_packs = math.ceil(float(buffer_size)/BUFF_SIZE)
66
67          frame_info = {"packs":int(num_packs)}
68
69          server_socket.sendto(pickle.dumps(frame_info), addr)
70
71          left = 0
72          right = BUFF_SIZE
73
74          for index in range(int(num_packs)):
75              data = buffer[left:right]
76              left = right
77              right += BUFF_SIZE
78
79              server_socket.sendto(data, addr)
```

B.3 Código de conexão de rede local para recebimento de imagens - *Python 3.11*

Código B.3 – Código Socket - *Python 3.11 NAOqi*

```
1  import socket, cv2
2  import numpy as np
3  import pickle
4
5  BUFF_SIZE = 65000
6  client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
7  client_socket.setsockopt(socket.SOL_SOCKET, socket.SO_RCVBUF,
      BUFF_SIZE)
8
9  HOST = '127.0.0.1'
10 portHost = 8081
11 message = 'TEST CONNECTION SOCKET'
12
13 client_socket.sendto(message.encode('utf-8'), (HOST, portHost))
14
```

```
15
16 while True:
17         packet,addr = client_socket.recvfrom(BUFF_SIZE)
18
19         if len(packet) < 100:
20             frame_info = pickle.loads(packet)
21
22             if frame_info:
23                 nums_of_packs =  frame_info["packs"]
24
25                 for index in range(nums_of_packs):
26                     data, address =
                           client_socket.recvfrom(BUFF_SIZE)
27
28                     if index == 0:
29                         buffer = data
30                     else:
31                         buffer += data
32                 frame = np.frombuffer(buffer, dtype=np.uint8)
33                 frame = frame.reshape(frame.shape[0], 1)
34                 frame = cv2.imdecode(frame, cv2.IMREAD_COLOR)
```

B.4 Código de Desenho do Keypoints - *Python 3.11*

Código B.4 – Código *Mediapipe - Python 3.11 NAOqi*

```
1  import mediapipe as mp
2  import cv2
3  import numpy as np
4  import mediapipe.python.solutions.face_detection_test
5
6  class FaceMeshDetector:
7      def __init__(self, staticMode=False, maxFaces=1,
          minDetectionCon=0.5, minTrackCon=0.5):
8          self.results = None
9          self.imgRGB = None
10         self.staticMode = staticMode
11         self.maxFaces = maxFaces
12         self.minDetectionCon = minDetectionCon
13         self.minTrackCon = minTrackCon
14
15         self.mpDraw = mediapipe.python.solutions.drawing_utils
16         self.mpFaceMash = mediapipe.python.solutions.face_mesh
17         self.faceMash = self.mpFaceMash.FaceMesh(max_num_faces=10)
18         self.drawSpec = self.mpDraw.DrawingSpec(thickness=1,
              circle_radius=1)
19
20         faceDetection =
              mp.solutions.mediapipe.python.solutions.face_detection
```

```
21          self.face = faceDetection.FaceDetection(model_selection=0,
               min_detection_confidence=0.7)
22          def findOnlyFaceMesh(self, img, draw=False):
23              self.imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
24              self.results = self.faceMash.process(self.imgRGB)
25
26              empty = np.zeros(img.shape, dtype='uint8')
27              img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
28
29              if self.results.multi_face_landmarks:
30                  for faceLms in self.results.multi_face_landmarks:
31                      if draw:
32                          self.mpDraw.draw_landmarks(img, faceLms,
                               self.mpFaceMash.FACEMESH_FACE_OVAL,
33                                                     self.drawSpec,
                                                          self.drawSpec)
34                          self.mpDraw.draw_landmarks(empty, faceLms,
                               self.mpFaceMash.FACEMESH_FACE_OVAL,
35                                                     self.drawSpec,
                                                          self.drawSpec)
36                  face = []
37                  for id, lm in enumerate(faceLms.landmark):
38                      ih, iw = img.shape
39                      x, y = int(lm.x*iw), int(lm.y*ih)
40                      img = cv2.circle(img, (x,y), 1, (255, 255,
                           255), -1)
41                      empty = cv2.circle(empty, (x,y), 1, (255, 255,
                           255), -1)
42                      face.append([x,y])
43              return empty, img
```

B.5 Código de separação da base AffectNet

Código B.5 – Código Separação de imagens - *Python NAOqi*

```
1  import shutil
2  import os
3  import numpy as np
4  import argparse
5  import sys
6
7  def get_files_from_folder(path):
8
9      files = os.listdir(path)
10     return np.asarray(files)
11
12 def split_dataset(path_to_data, path_to_test_data, train_ratio):
13     # get dirs
14     _, dirs, _ = next(os.walk(path_to_data))
15
```

```
    # calculates how many train data per class
    data_counter_per_class = np.zeros((len(dirs)))
    for i in range(len(dirs)):
        path = os.path.join(path_to_data, dirs[i])
        files = get_files_from_folder(path)
        data_counter_per_class[i] = len(files)
    test_counter = np.round(data_counter_per_class * (1 -
       train_ratio))

    # transfers files
    for i in range(len(dirs)):
        path_to_original = os.path.join(path_to_data, dirs[i])
        path_to_save = os.path.join(path_to_test_data, dirs[i])

        #creates dir
        if not os.path.exists(path_to_save):
            os.makedirs(path_to_save)
        files = get_files_from_folder(path_to_original)
        # moves data
        for j in range(int(test_counter[i])):
            dst = os.path.join(path_to_save, files[j])
            src = os.path.join(path_to_original, files[j])
            shutil.move(src, dst)
    next(os.walk(path_to_data))


data_directory = 'AffectNetData'
data_directory_train = "AffectNet/train"
data_directory_test = "AffectNet/test"

split_dataset(data_directory, data_directory_test, 0.7)
split_dataset(data_directory, data_directory_train, 0.0)
```

B.6 Código do tratamento do Classificador

Código B.6 – Código modelo do classificador - *Python NAOqi*

```
# Importing Libraries
import numpy as np
import tensorflow as tf

from tensorflow.keras.models import model_from_json
from tensorflow.python.keras.backend import set_session

# Initializing TF Session
config = tf.compat.v1.ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.15
session = tf.compat.v1.Session(config=config)
set_session(session)

```

```
14
15 class FacialExpressionModel(object):
16     # List of Emotions
17     EMOTIONS_LIST = ["Angry", "Disgust",
18                      "Fear", "Happy",
19                      "Neutral", "Sad",
20                      "Surprise"]
21
22     def __init__(self, model_json_file, model_weights_file):
23         # load model from JSON file
24         with open(model_json_file, 'r') as json_file:
25             loaded_model_json = json_file.read()
26         self.loaded_model = model_from_json(loaded_model_json)
27
28         # load weights into the new model
29         self.loaded_model.load_weights(model_weights_file)
30
31     # Function to predict emotion
32     def predict_emotion(self, img):
33         global session
34         set_session(session)
35         self.preds = self.loaded_model.predict(img)
36         return
              FacialExpressionModel.EMOTIONS_LIST[np.argmax(self.preds)],
              self.preds[0]
```

B.7 Código do Uso do Classificador

Código B.7 – Código de Uso do Classificador gerados pelo treino - *Python*

```
1  # Importing Libraries
2  import numpy as np
3  import tensorflow as tf
4
5  from tensorflow.keras.models import model_from_json
6  from tensorflow.python.keras.backend import set_session
7
8  # Initializing TF Session
9  config = tf.compat.v1.ConfigProto()
10 config.gpu_options.per_process_gpu_memory_fraction = 0.15
11 session = tf.compat.v1.Session(config=config)
12 set_session(session)
13
14
15 class FacialExpressionModel(object):
16     EMOTIONS_LIST = ["Angry", "Disgust",
17                      "Fear", "Happy",
18                      "Neutral", "Sad",
19                      "Surprise"]
20
```

```
21
22      def __init__(self, model_json_file, model_weights_file):
23          with open(model_json_file, 'r') as json_file:
24              loaded_model_json = json_file.read()
25          self.loaded_model = model_from_json(loaded_model_json)
26
27          self.loaded_model.load_weights(model_weights_file)
28
29      def predict_emotion(self, img):
30          global session
31          set_session(session)
32          self.preds = self.loaded_model.predict(img)
33          return
               FacialExpressionModel.EMOTIONS_LIST[np.argmax(self.preds)],
               self.preds[0]
```

Anexos

ANEXO A – Reconhecimento com 4 camadas

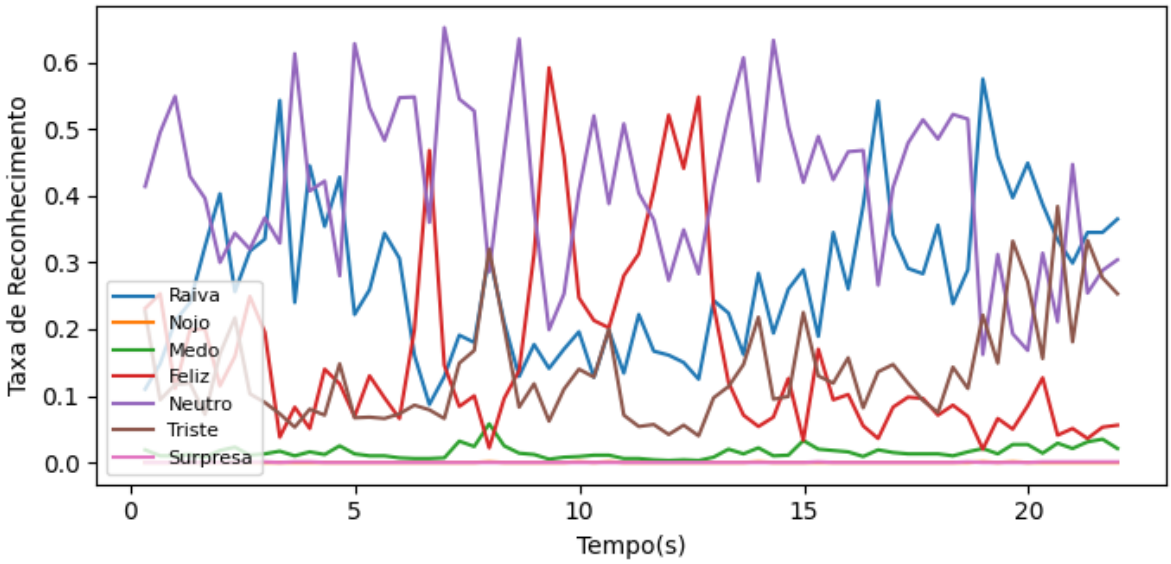

Figura 62 – Gráfico para Reconhecimento de Raiva com 4 camadas

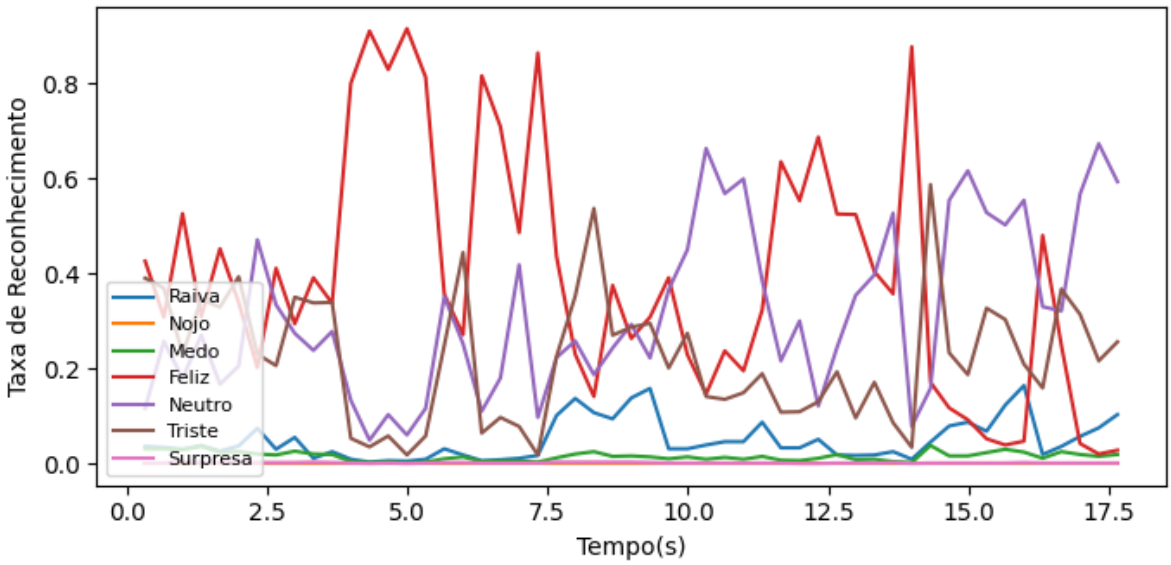

Figura 63 – Gráfico para Reconhecimento de Nojo com 4 camadas

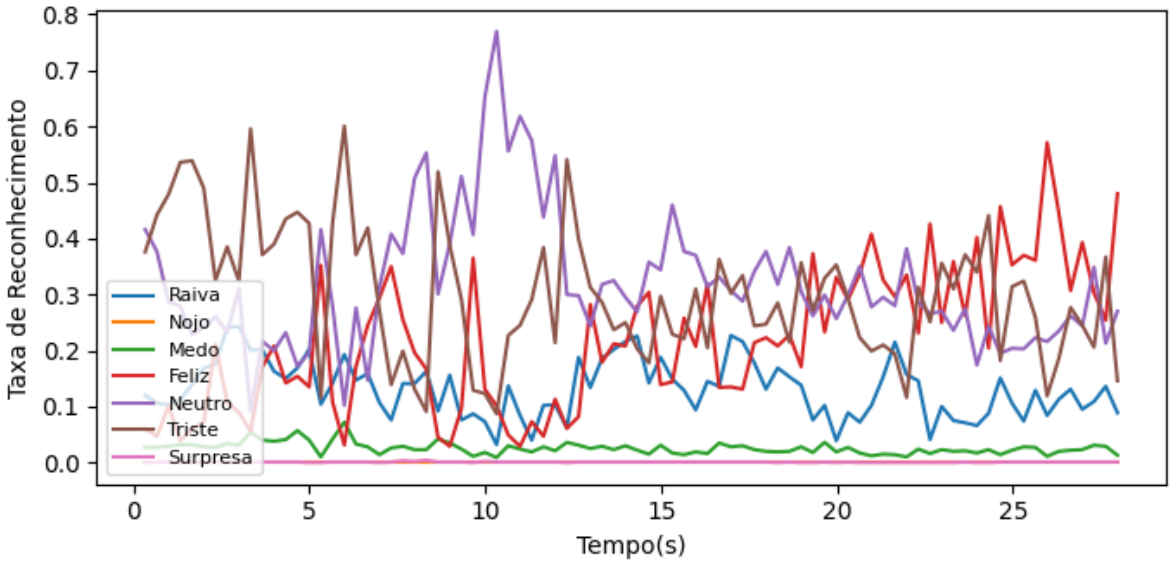

Figura 64 – Gráfico para Reconhecimento de Tristeza com 4 camadas

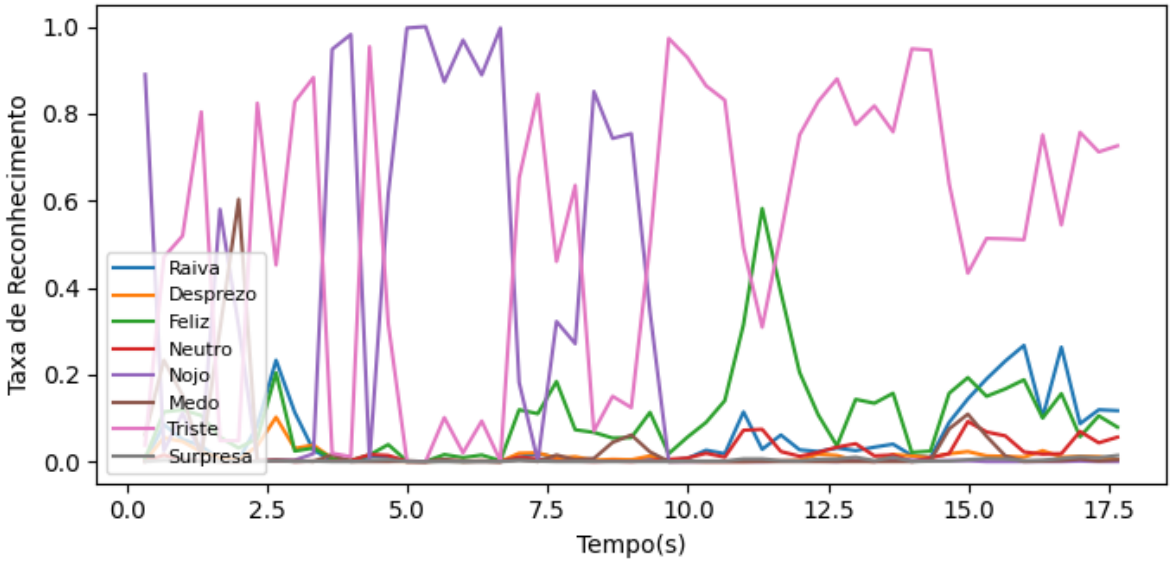

Figura 65 – Gráfico para Reconhecimento de Nojo com *Keypoints*

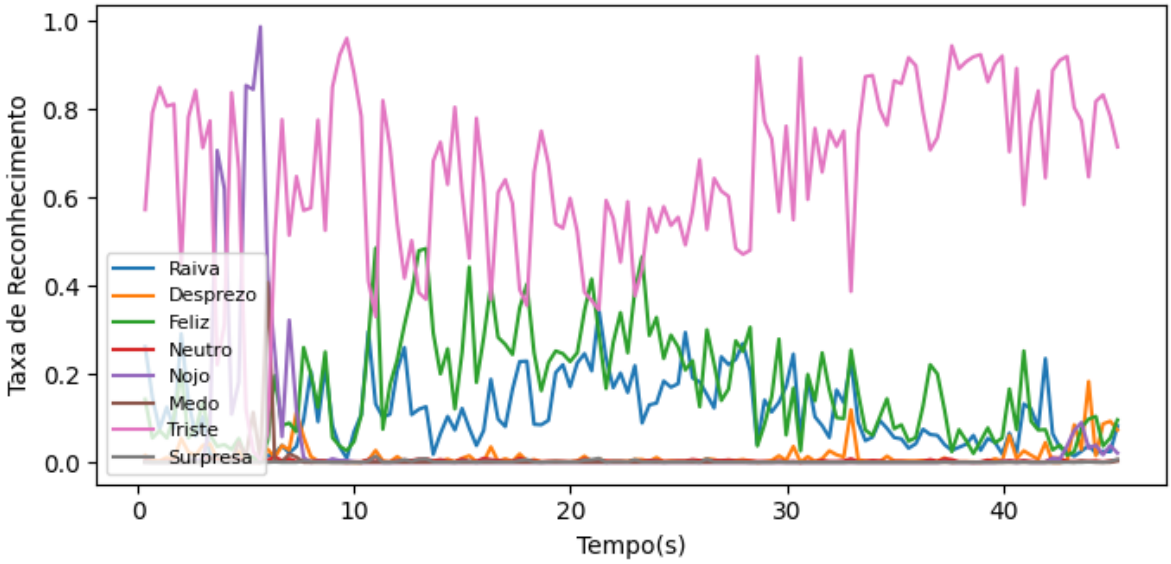

Figura 66 – Gráfico para Reconhecimento de Medo com *Keypoints*

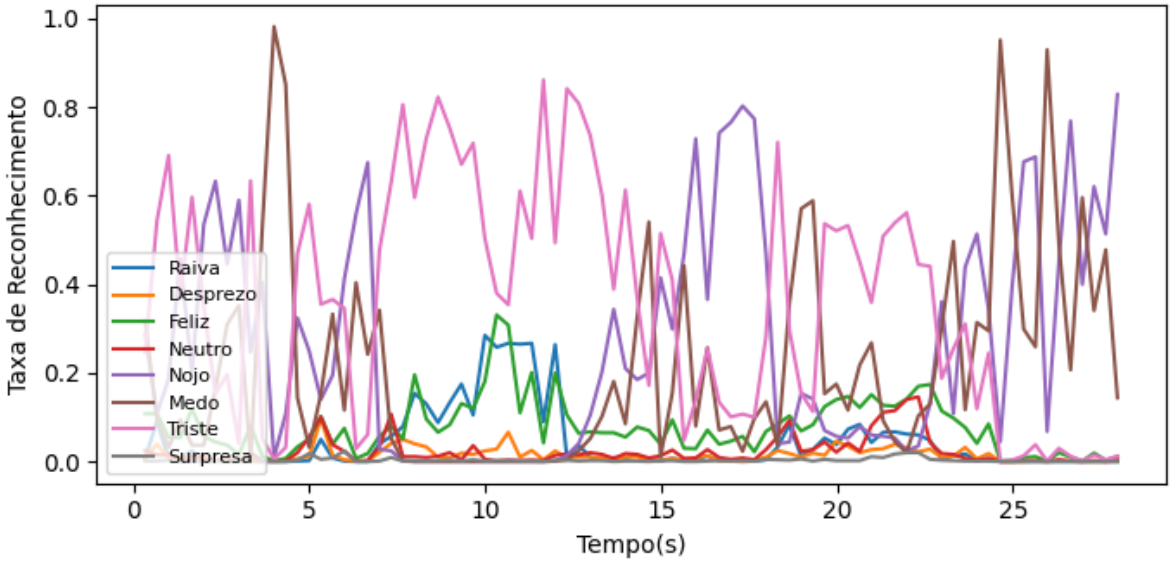

Figura 67 – Gráfico para Reconhecimento de Triste com *Keypoints*

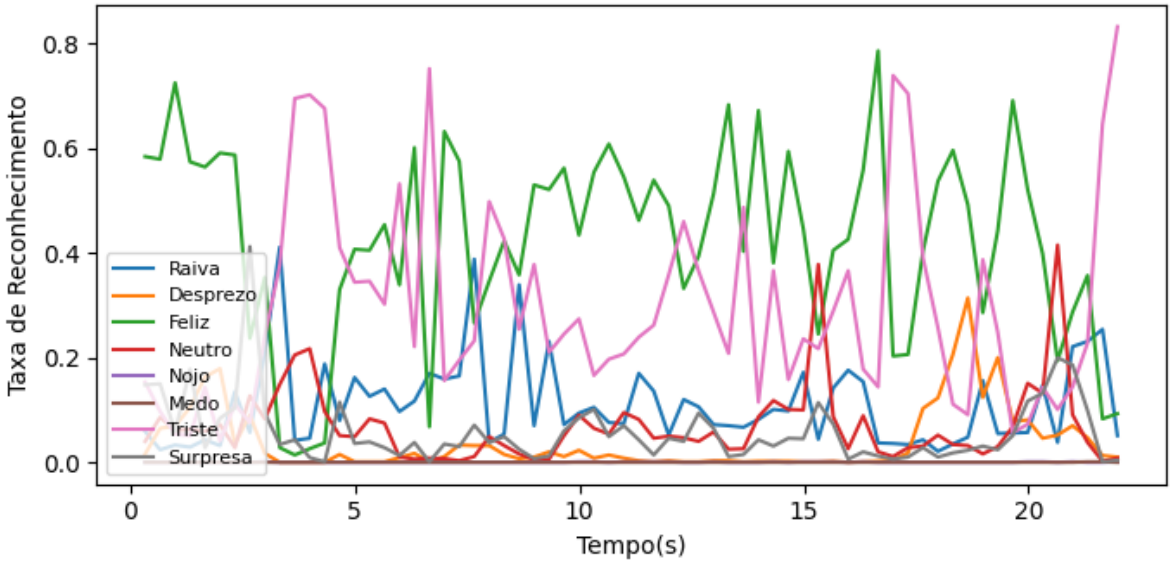

Figura 68 – Gráfico para Reconhecimento de Surpresa com *Keypoints*