UNIVERSIDADE FEDERAL DE ITAJUBÁ

TinyML - Sistema de Reconhecimento Facial para Autenticação

Felipe Faustino Brito - 2021007366
Felipe Queiroz Flores Quintão Bachetti - 2021005147
Jorge Christino dos Santos Ferreira - 2021014825

IEST01 - 01/07/2025

1. Introdução

O reconhecimento facial é uma das aplicações mais proeminentes da visão computacional, com vasto potencial em áreas como segurança, personalização de interfaces e automação residencial. No entanto, a execução de modelos de reconhecimento facial, tradicionalmente pesados e exigentes em poder de processamento, representa um desafio considerável para plataformas embarcadas de baixo custo.

Este projeto aborda diretamente esse desafio, detalhando o desenvolvimento, treinamento e teste de um modelo de Inteligência Artificial para reconhecimento facial, projetado para operar em um ambiente de hardware restrito. O objetivo central foi criar um sistema capaz de identificar, com precisão e em tempo real, as faces dos três integrantes do grupo de trabalho.

Para alcançar este objetivo, foi utilizada a plataforma *Edge Impulse Studio*, que simplifica o fluxo de trabalho de desenvolvimento de aplicações de *Machine Learning* embarcado. O processo envolveu a coleta e o pré-processamento de um conjunto de dados de imagens, o treinamento de um modelo de rede neural convolucional utilizando a técnica de *Transfer Learning* com a arquitetura *MobileNetV2* — otimizada para performance em dispositivos móveis e embarcados — e, por fim, a validação e o *deploy* do modelo finalizado na placa microcontroladora Xiao ESP32S3 Sense.

2. Dataset

A captura das imagens foi efetuada diretamente pela câmera acoplada à placa Xiao ESP32S3, conectada ao Edge Impulse Studio através da ferramenta "Data Acquisition". Essa abordagem permitiu a coleta de dados em um ambiente muito próximo ao da aplicação final, assegurando que as características das imagens, como resolução e iluminação, fossem consistentes com as que o modelo encontraria durante a fase de inferência.

O dataset foi estruturado em quatro classes distintas: três classes nominais, cada uma correspondendo a um integrante do grupo ("brito", "felipe", "jorge"), e uma quarta classe denominada "vazio". A classe "vazio" foi populada com imagens do ambiente que não continham o rosto de nenhum dos membros, sendo crucial para treinar o modelo a reconhecer e ignorar entradas irrelevantes, evitando falsos positivos.

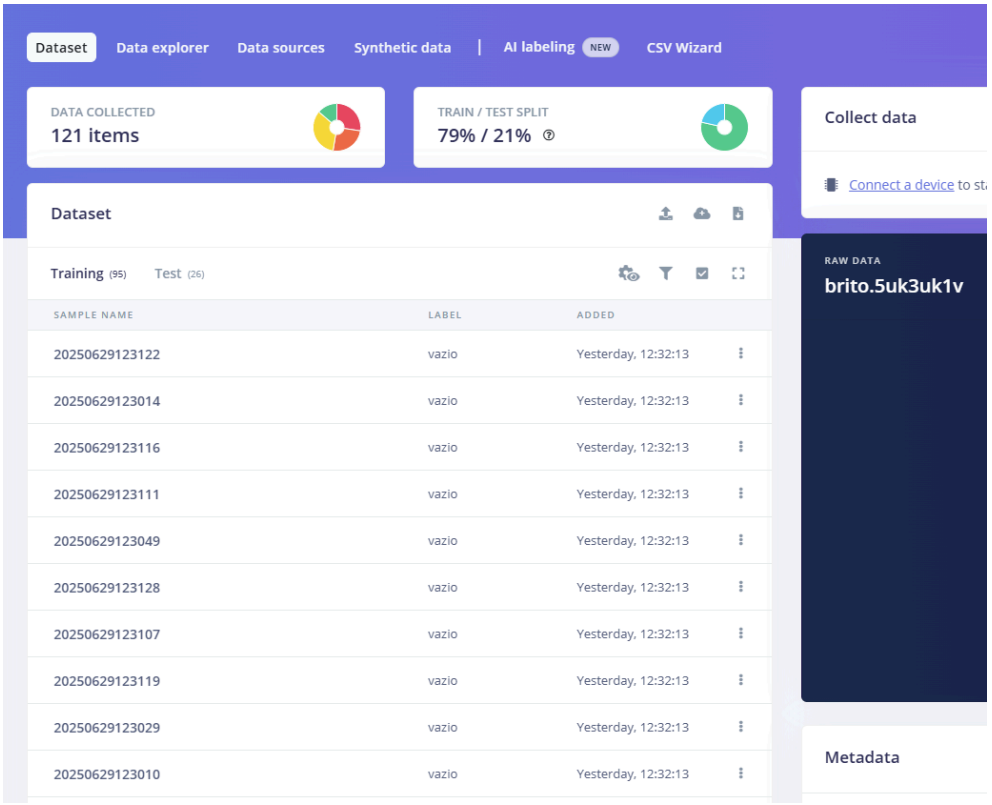
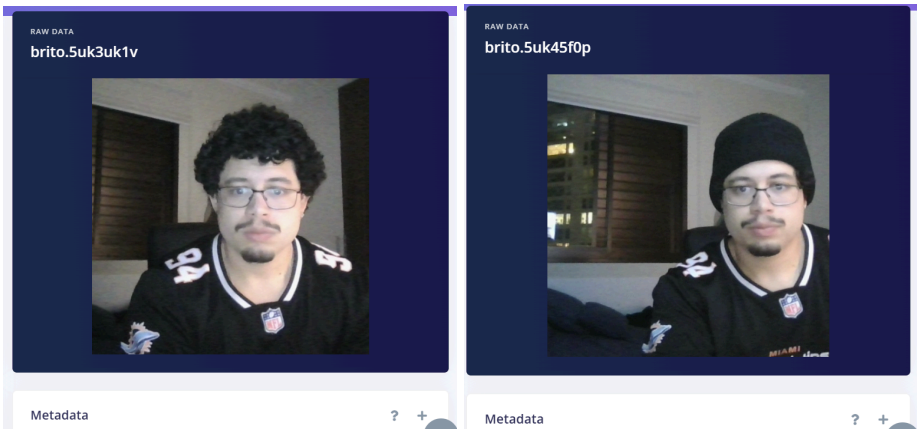


Imagem 1: Tela da ferramenta de “Data Acquisition”.

Com o objetivo de aumentar a robustez e a capacidade de generalização do modelo, houve uma preocupação em introduzir variabilidade durante a coleta. Para cada integrante, foram capturadas fotografias sob diferentes ângulos de face (frontal, perfilado), com expressões e iluminações variadas e com o uso de diferentes adereços, como óculos, boné e touca. Esta diversidade no treinamento é essencial para que o modelo seja capaz de realizar o reconhecimento mesmo sob condições variáveis.



Imagens 2 e 3: Exemplos de imagens capturadas para o dataset (label “brito”).

Ao todo, foram coletadas 121 imagens. O Edge Impulse Studio realizou automaticamente a divisão do conjunto de dados, alocando 95 imagens

(aproximadamente 79%) para o conjunto de treinamento e as 26 imagens restantes (aproximadamente 21%) para o conjunto de teste. Essa separação é vital para avaliar a performance do modelo em dados não vistos durante o treinamento, fornecendo uma métrica imparcial de sua acurácia.

Como etapa final de pré-processamento, foi definido que as imagens seriam convertidas para escala de cinza (preto e branco). A decisão de remover as informações de cor visa otimizar o processo, reduzindo a complexidade computacional e o tamanho do modelo. Além disso, essa técnica força o modelo a focar em características estruturais da face, como contornos, formas e texturas, o que pode levar a um aumento na precisão do reconhecimento em sistemas embarcados com recursos limitados.

3. Modelo

O modelo foi treinado utilizando Transfer Learning com uma rede leve, a MobileNetV2 96x96 0.35. Seu input recebe 9216 features e retorna 4 classes, sendo que sua última camada tem 16 neurônios e um dropout de 0,1.

As seguintes configurações de treinamento foram utilizadas:

- 20 épocas
- Taxa de aprendizado: 0.0005
- Treinamento em CPU
- Data Augmentation ativada
- Validação: 20%
- Batch Size: 32

Seguem seus resultados:

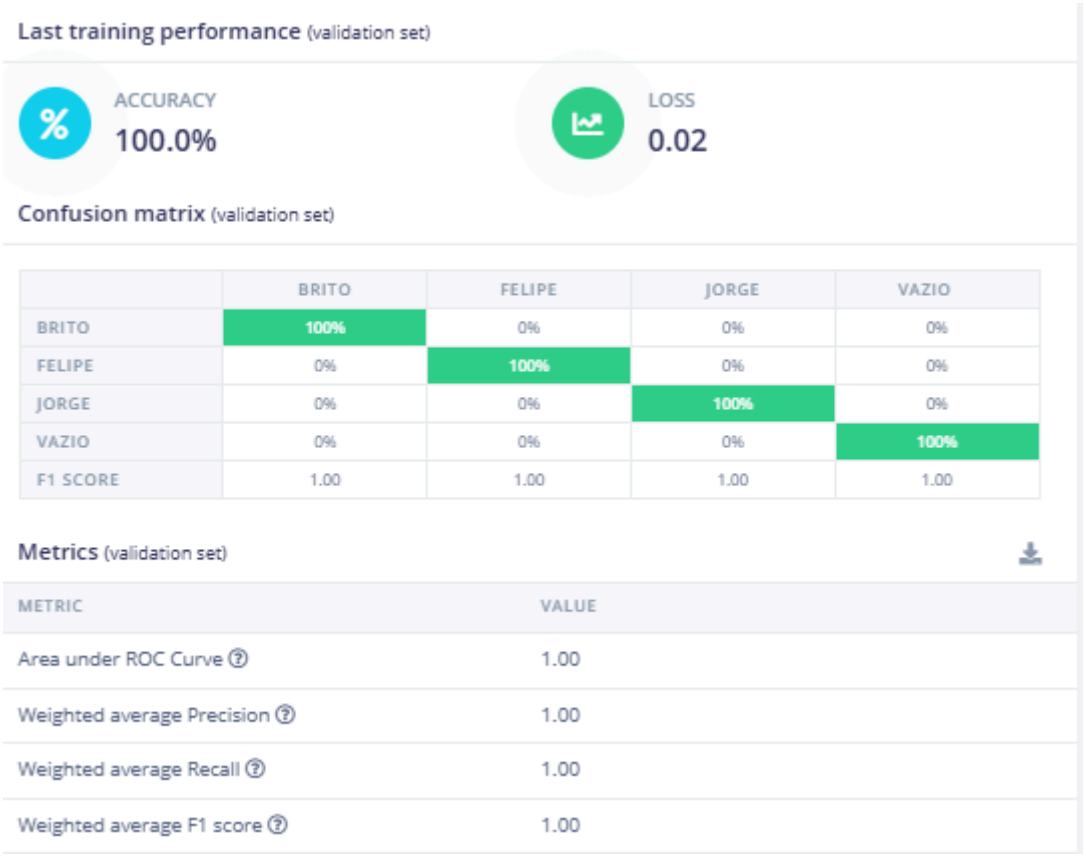


Imagem 4: Performance do treinamento.

A etapa de validação retornou ótimos resultados com acurácia de 100% e loss de 0.02.

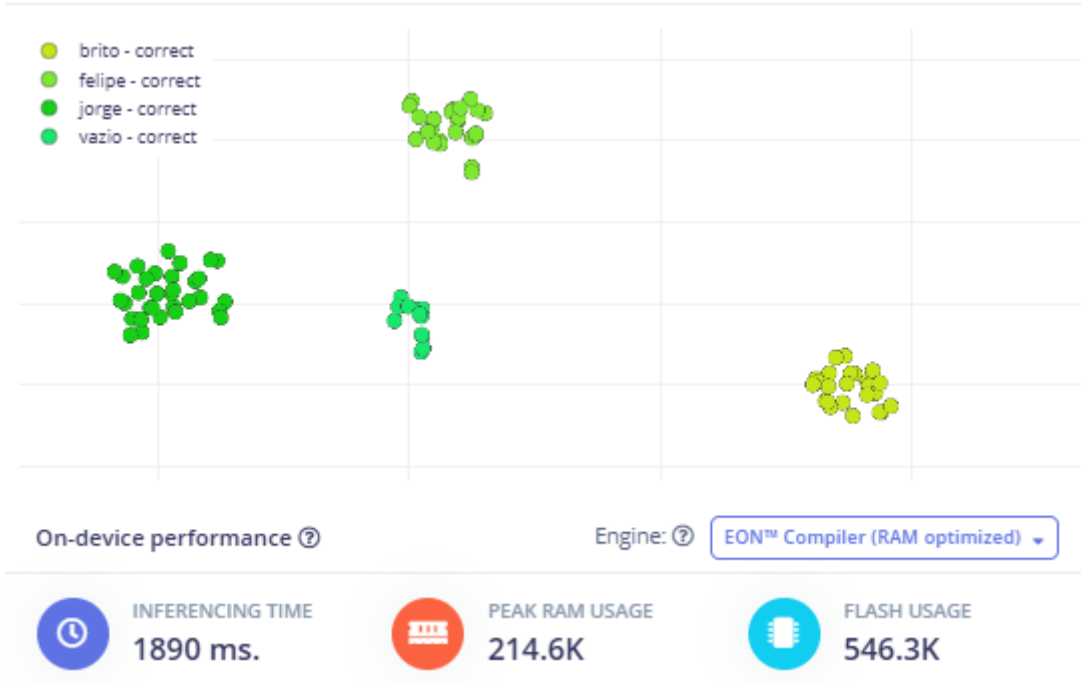


Imagem 5: Gráfico de performance.

4. Testes

Para a etapa de testes, foram selecionadas 26 imagens (21% de um total de 121). Esses foram os resultados:

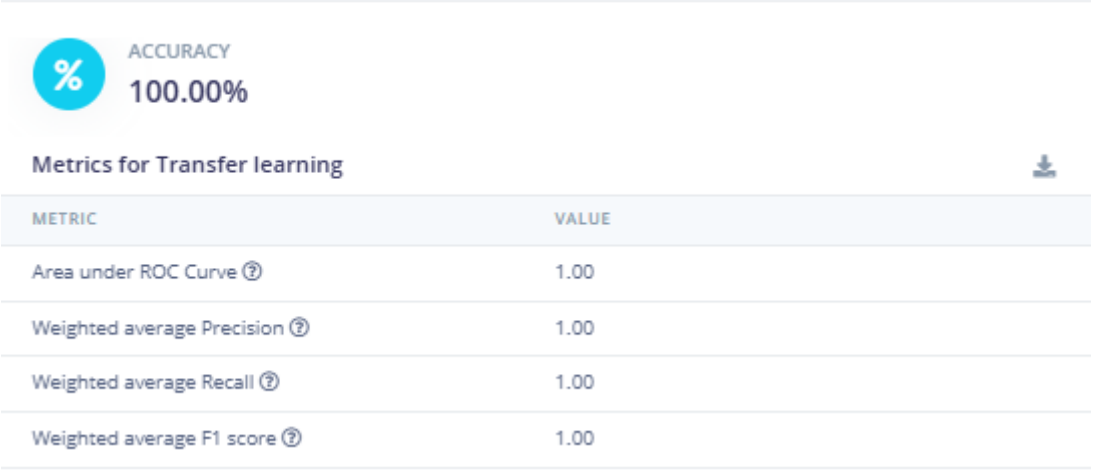


Imagem 6: Resultados do teste

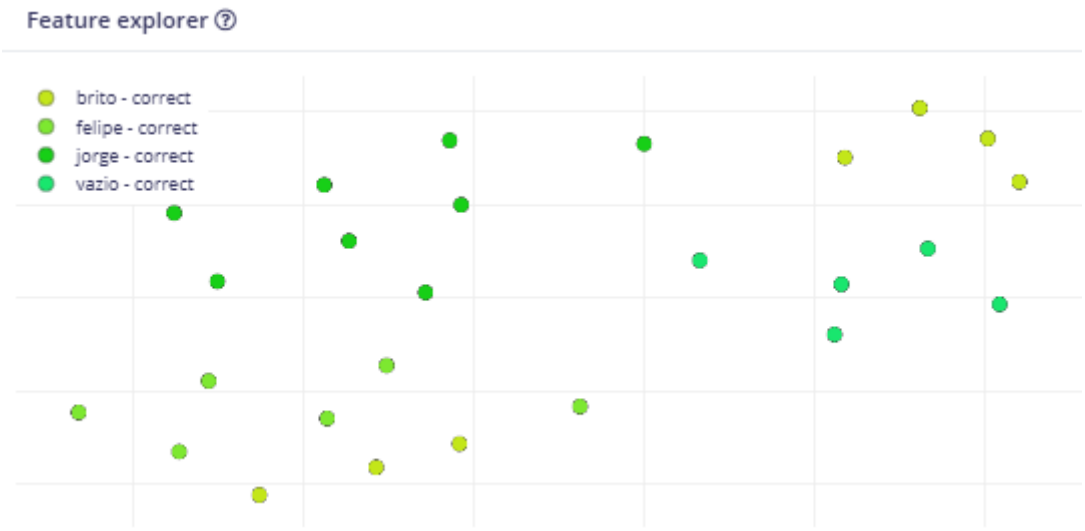


Imagem 7: Feature Explorer para resultados do teste

Foram alcançados bons resultados, indicando que o modelo é capaz de distinguir com precisão as classes treinadas. Entretanto, é importante ressaltar que resultados perfeitos podem indicar possível overfitting, sendo assim resta aos testes com a implementação na placa para validar o desempenho do modelo em condições reais de uso.

5. Deploy e Inferências

Após o treinamento do modelo no Edge Impulse, realizamos o processo de deploy na placa XIAO ESP32S3 Sense. Nessa etapa, selecionamos a opção "Arduino Library" como formato de exportação, essa opção gera um arquivo .zip contendo todo o código necessário para rodar o modelo de inferência diretamente em ambientes com a IDE do Arduino.

Ao fazer o build do modelo, a plataforma compila o modelo e gera a biblioteca com suporte à aceleração de inferência via EON Compiler, otimizada para execução em dispositivos com recursos computacionais limitados como o ESP32S3. Foi escolhido o modelo não otimizado, pois o modelo otimizando demonstrou problemas ao fazer o deployment na placa.

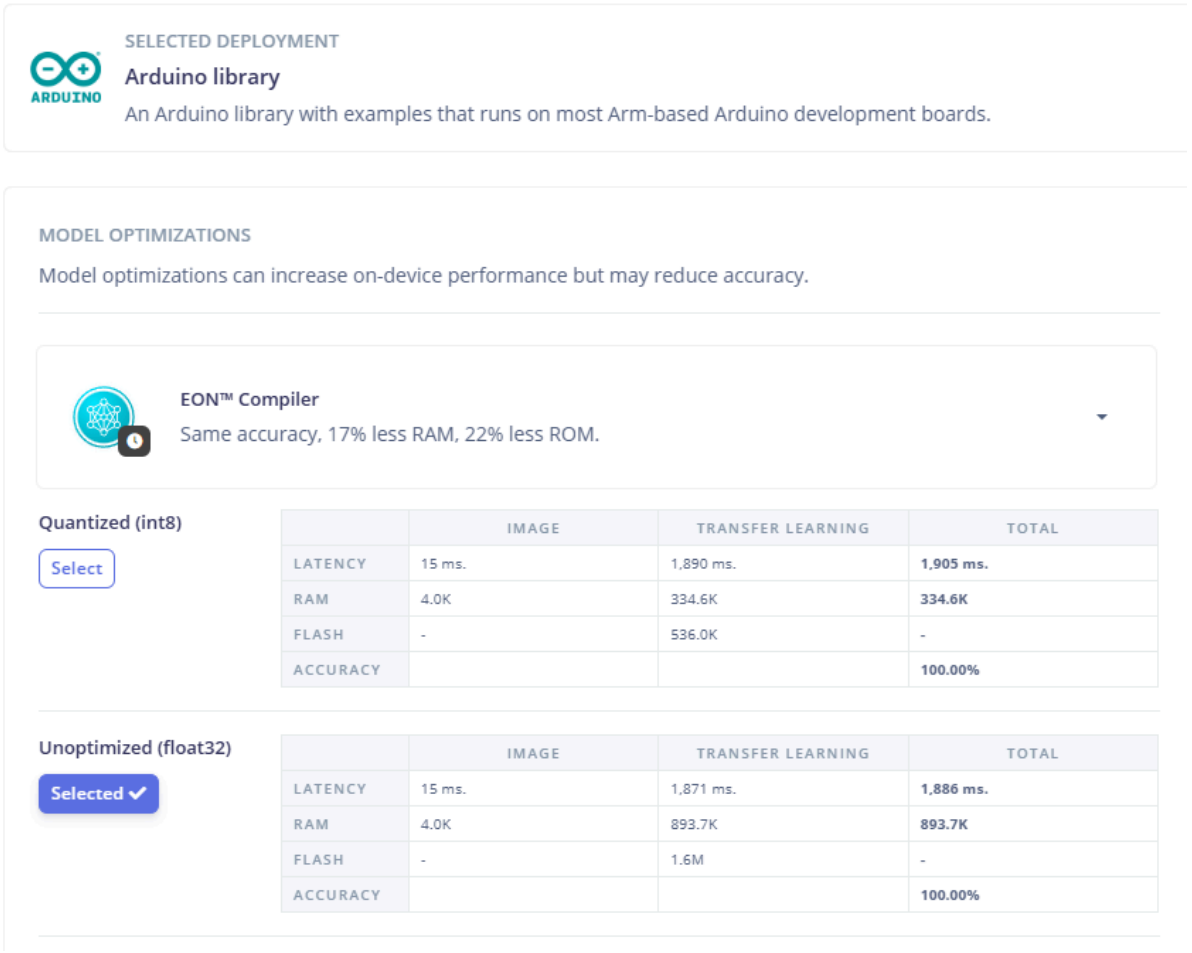


Imagem 8 - Configurações do deployment

Com o código em execução, no Serial Monitor da plataforma Arduino IDE, é exibido a previsão indicando a classe detectada com as respectivas probabilidades. Além disso, foi construído um sistema de autenticação utilizando um circuito com dois

leds, quando uma pessoa tiver permissão acende o led verde e quando for detectado uma pessoa sem permissão ou não for identificado uma pessoa acende o led vermelho.

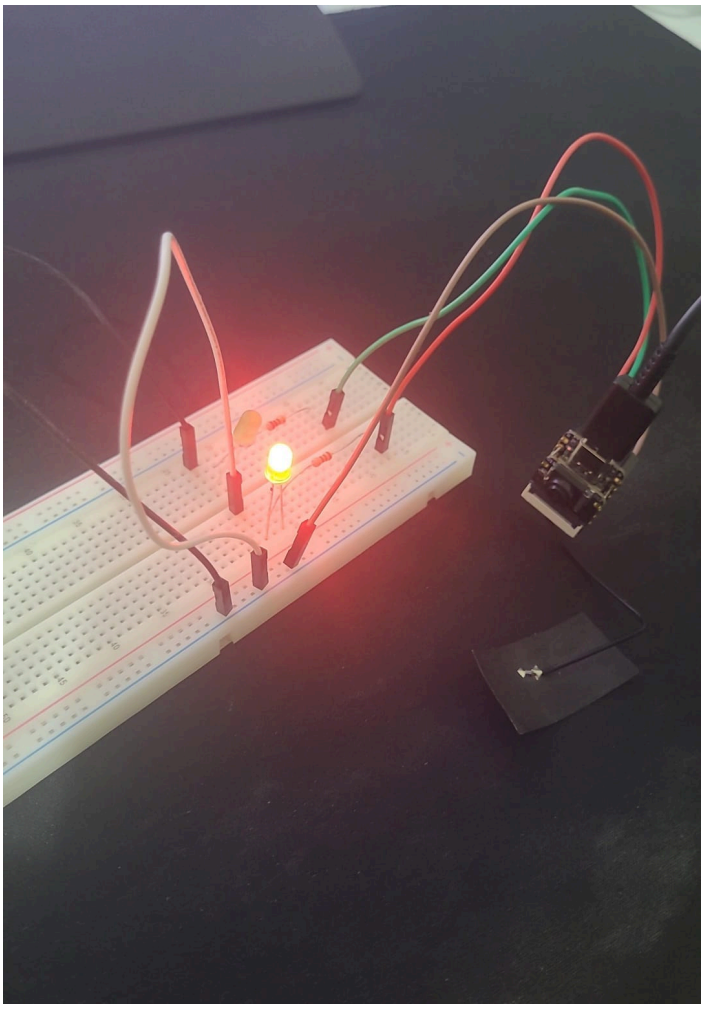


Imagem 9 - Circuito com leds fazendo a inferência

```
Predictions (DSP: 11 ms., Classification: 4439 ms., Anomaly: 0 ms.):
Predictions:
  brito: 0.00273
  felipe: 0.26662
  jorge: 0.65472
  vazio: 0.07593
  Prediction: jorge with probability 0.65
```

Imagem 10 - Inferência no serial monitor

6. Conclusão

Embora o modelo de reconhecimento facial treinado na plataforma Edge Impulse tenha apresentado 100% de acurácia durante a fase de validação, os testes em tempo real realizados diretamente na placa evidenciaram algumas divergências na inferência.

Durante os experimentos, foi observado que o modelo teve dificuldade em distinguir corretamente entre duas pessoas, gerando classificações equivocadas em alguns cenários. Esses erros são atribuídos a fatores externos que impactam diretamente a qualidade da imagem capturada, tais como variação no ângulo da câmera, condições de iluminação inconsistentes, baixa resolução da imagem.