UNIVERSIDADE FEDERAL DE ITAJUBÁ

CLASSIFICADOR DE VOGAIS EM LIBRAS

TINYML - IESTI01

ANA CAROLINA MALVEIRA FERREIRA - 35190
ALLYSON GONCALVES RAMOS - 2017006700
RAFAEL ROCHA MACIEL - 2018005619

ITAJUBÁ, 2022

1. Introdução

A Língua Brasileira de Sinais (LIBRAS) é uma forma de comunicação baseada em gestos e símbolos feitos com as mãos e os braços, permitindo a comunicação entre pessoas sem o uso da fala. Os símbolos/gestos podem representar palavras, frases e também as letras do alfabeto, sendo que cada letra possui um símbolo específico. Esse tipo de linguagem é principalmente utilizada por pessoas com deficiência (PCD) auditiva ou com deficiência na fala.

Levando em conta a importância de gerar aplicações que permitam a inclusão e a divulgação da LIBRAS, a proposta consiste em desenvolver um classificador para os símbolos em Libras. Após a análise de alguns exemplos, datasets e verificando as especificações do Hardware foi estabelecido que o principal objetivo seria construir um modelo em TinyML para classificar os símbolos correspondentes às vogais em LIBRAS.

Para o desenvolvimento da aplicação foi utilizada a plataforma Edge Impulse, um dataset com fotos dos símbolos em LIBRAS para as letras na resolução de 64x64 - disponibilizado em https://www.kaggle.com/datasets/williansoliveira/libras - sendo que foram adicionadas imagens complementares ao dataset de autoria própria. Além disso, foi utilizado o Tiny Machine Learning Kit Arduino que consiste na placa Arduino Nano 33 BLE Sense e, como o principal elemento sensor para a aquisição dos dados para inferência, a câmera OV7675.

Os modelos foram gerados e treinados no Edge Impulse, sendo que apenas os dois últimos receberam um deploy para o Arduino Nano 33 BLE Sense.

2. Modelos

2.1. Primeiro modelo

Para o primeiro modelo foi utilizado o seguinte impulso, com imagens de tamanho 64x64 pixels. Foram utilizadas todas as imagens das vogais presentes no dataset mostrado anteriormente.

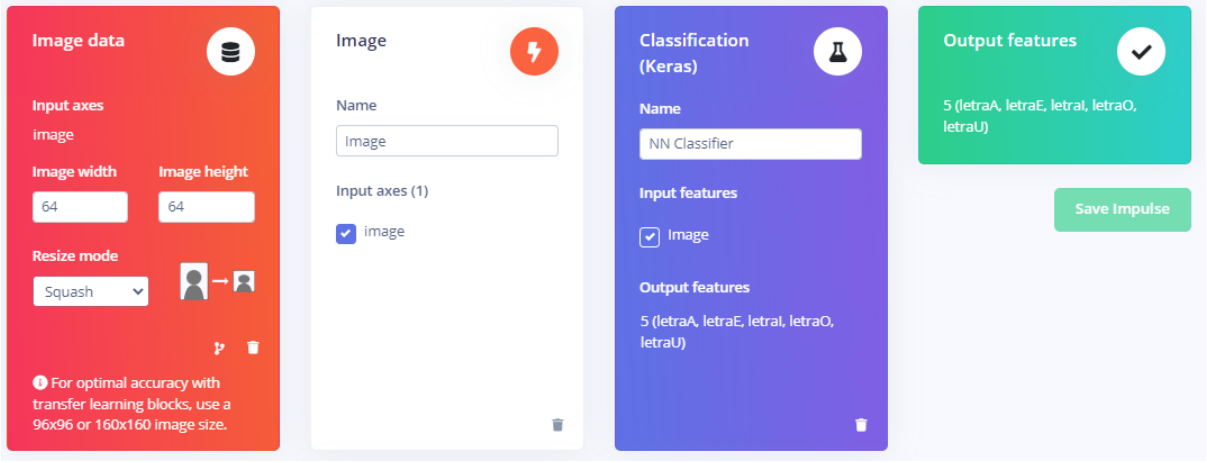


Imagem 1 - Impulse design do primeiro modelo

Com esse impulso foram obtidos os seguintes resultados no classificador.

	LETRAA	LETRAE	LETRAI	LETRAO	LETRAU
LETRAA	99.7%	0.3%	0%	0%	0%
LETRAE	0%	99.7%	0.3%	0%	0%
LETRAI	0%	0%	100%	0%	0%
LETRAO	0%	0.3%	0%	99.7%	0%
LETRAU	0%	0%	0%	0%	100%
F1 SCORE	1.00	1.00	1.00	1.00	1.00

Imagem 2 - Resultados do primeiro modelo

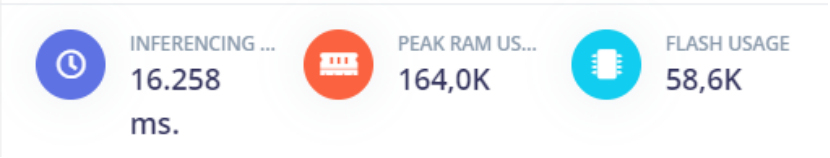


Imagem 3 - Performance do primeiro modelo

Com esses resultados, foi concluído que esse modelo seria inviável para o kit de desenvolvimento, visto que utilizaria uma quantidade elevada de memória RAM e levaria um tempo elevado para a inferência.

2.2. Segundo modelo

Para tentar diminuir a quantidade de RAM e o tempo de inferência, foi criado um impulso que utilizava imagens de 32x32 pixels.

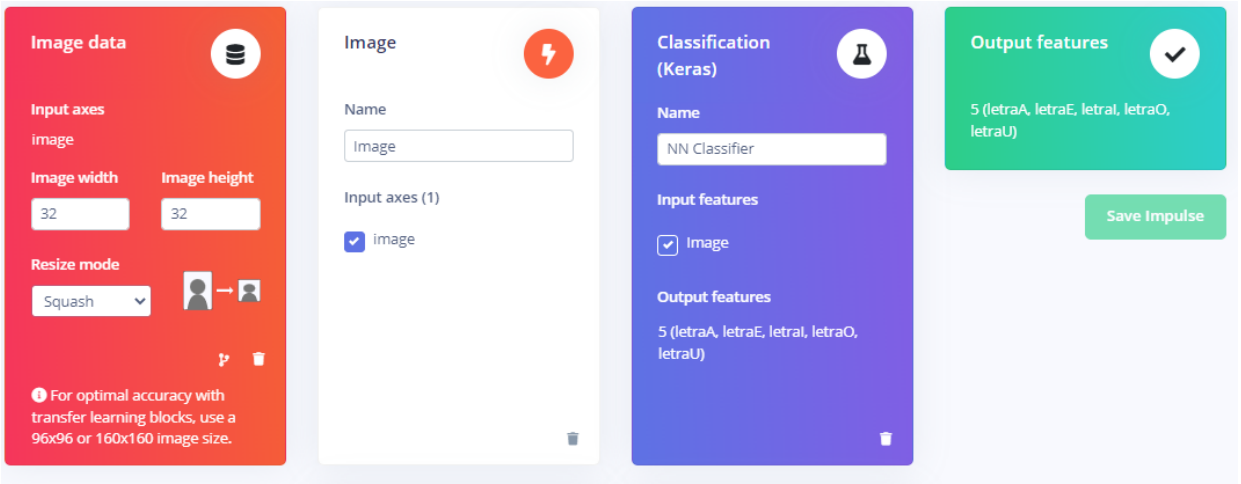


Imagem 4 - Impulse design do segundo modelo

Com esse impulso foram obtidos os seguintes resultados.

	LETRAA	LETRAE	LETRAI	LETRAO	LETRAU
LETRAA	99.7%	0.3%	0%	0%	0%
LETRAE	0.9%	99.1%	0%	0%	0%
LETRAI	0%	0%	100%	0%	0%
LETRAO	0.3%	0%	0%	99.4%	0.3%
LETRAU	0%	0%	0%	0%	100%
F1 SCORE	0.99	0.99	1.00	1.00	1.00

Imagem 5 - Resultados do segundo modelo

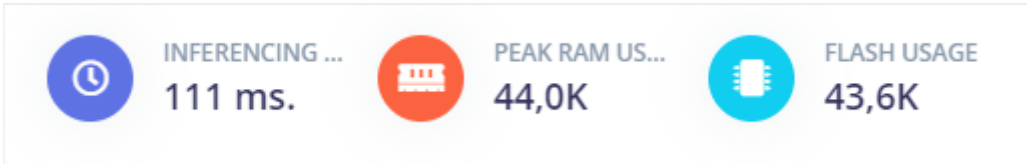


Imagem 6 - Performance do segundo modelo

Como os resultados foram promissores, realizou-se o deploy deste modelo no celular, de modo a verificar se a inferência ocorreria de forma adequada. Nesse caso, a inferência não aconteceu como o esperado. Um dos possíveis motivos para essa falha na inferência, pode ser pelo fato do modelo utilizar imagens pequenas, que podem não dar ênfase nas pequenas diferenças entre os sinais.

2.3. Terceiro modelo

Visando a utilização de um modelo com imagens 64x64 pixels, foi utilizado apenas uma quantidade reduzida de imagens do dataset original. Essas imagens foram selecionadas, de modo a possuir características semelhantes. Por exemplo, as imagens com pouca iluminação ou com parte do braço foram excluídas.

Também, foram coletadas imagens pelo grupo, visando deixar o dataset um pouco mais heterogêneo. Essas imagens foram fotografadas com uma boa iluminação e seguindo o padrão das fotos mantidas do dataset utilizado (Fundo de cor mais clara e imagem apenas da mão).

Outra adição nesse novo modelo foi a classe “Nada”, que foi criada com fotos de uma parede vazia, sem a presença da mão fazendo algum sinal. Outra variação nesse modelo, foi a mudança de “Classification (Keras)” para “Transfer Learning”.

O impulso utilizado e os resultados obtidos estão mostrados abaixo.

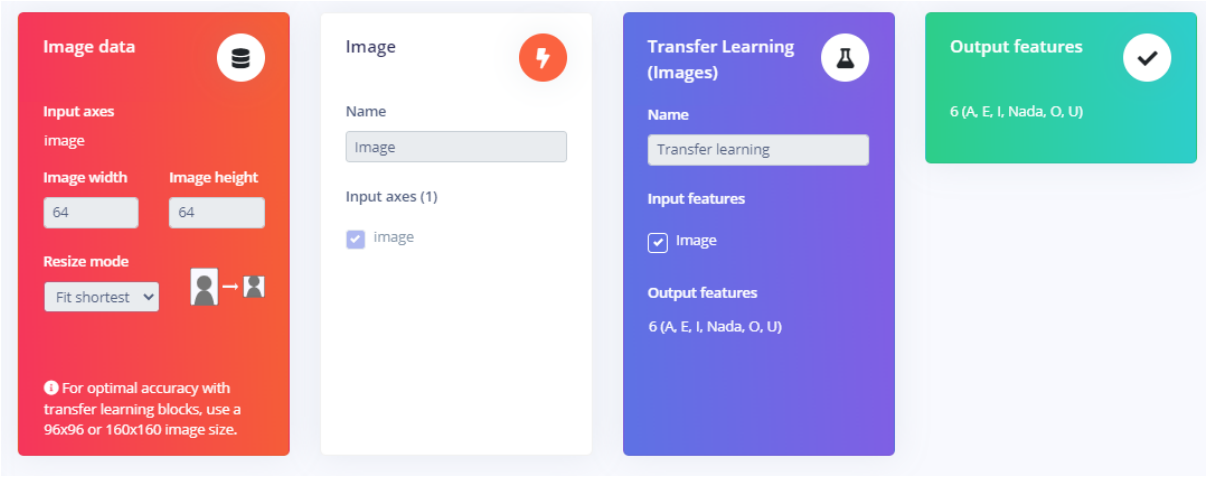
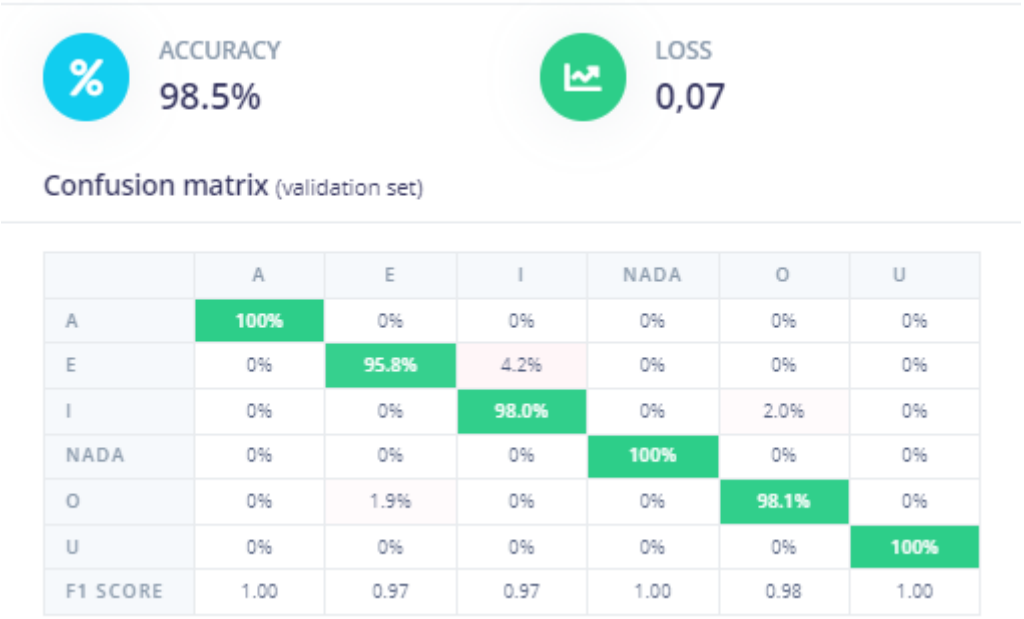


Imagem 7 - Impulse Design para o terceiro modelo



	A	E	I	NADA	O	U
A	100%	0%	0%	0%	0%	0%
E	0%	95.8%	4.2%	0%	0%	0%
I	0%	0%	98.0%	0%	2.0%	0%
NADA	0%	0%	0%	100%	0%	0%
O	0%	1.9%	0%	0%	98.1%	0%
U	0%	0%	0%	0%	0%	100%
F1 SCORE	1.00	0.97	0.97	1.00	0.98	1.00

Imagem 8 - Resultados do terceiro modelo

Imagem 9 - Performance do terceiro modelo

Como foram obtidos bons resultados com o modelo, foi realizado o deploy no celular. A inferência pelo celular ocorreu de maneira satisfatória, como mostrado nas imagens abaixo.

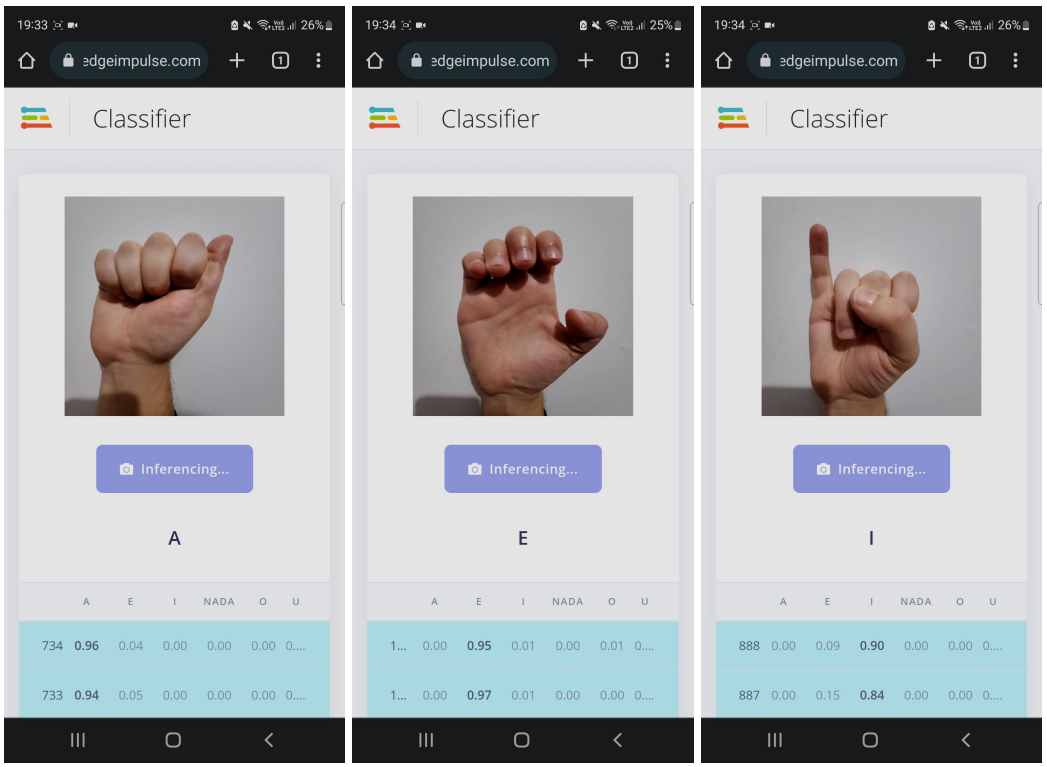


Imagem 10 - Teste para os sinais das letras A, E e I

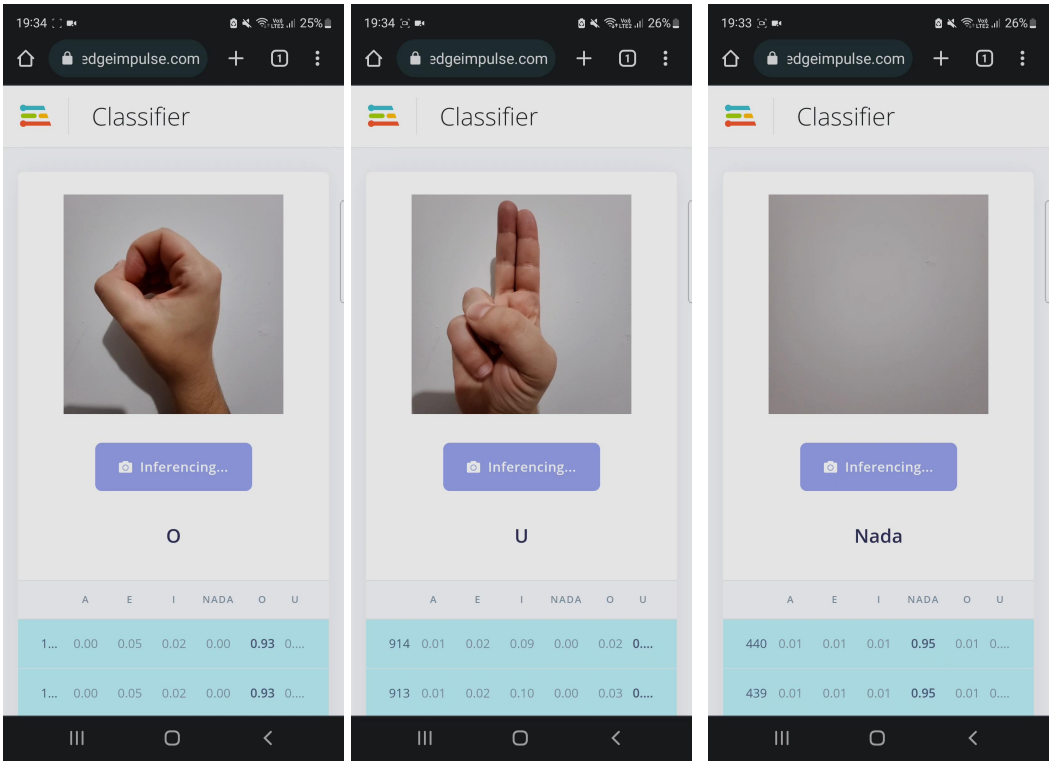


Imagem 11 - Teste para os sinais das letras O, U e Nada

Concluído o teste pelo celular, foi realizado o deploy no arduino 33 BLE sense. Ao contrário do teste do celular, a inferência no kit não ocorreu conforme o esperado.

As classes confundidas eram das letras I e U. Assim, analisando melhor o dataset, notou-se a presença de letras I com a mão esquerda e mão direita, podendo esse ser o motivo da inferência não acontecer de maneira adequada.

2.4. Quarto modelo

De acordo com os resultados obtidos, a classe da letra I, foi dividida em “I_direita” e “I_esquerda”. Também foram coletados mais dados pelos integrantes do grupo.

O impulso para o modelo e os parâmetros utilizados na rede neural estão mostrados abaixo.

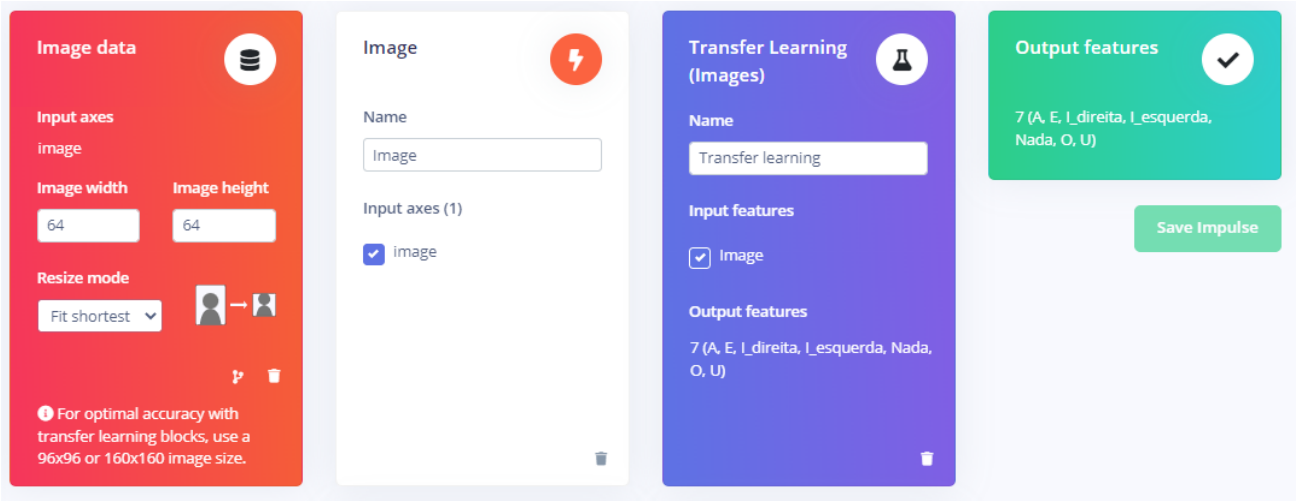


Imagem 12 - Impulse Design do quarto modelo

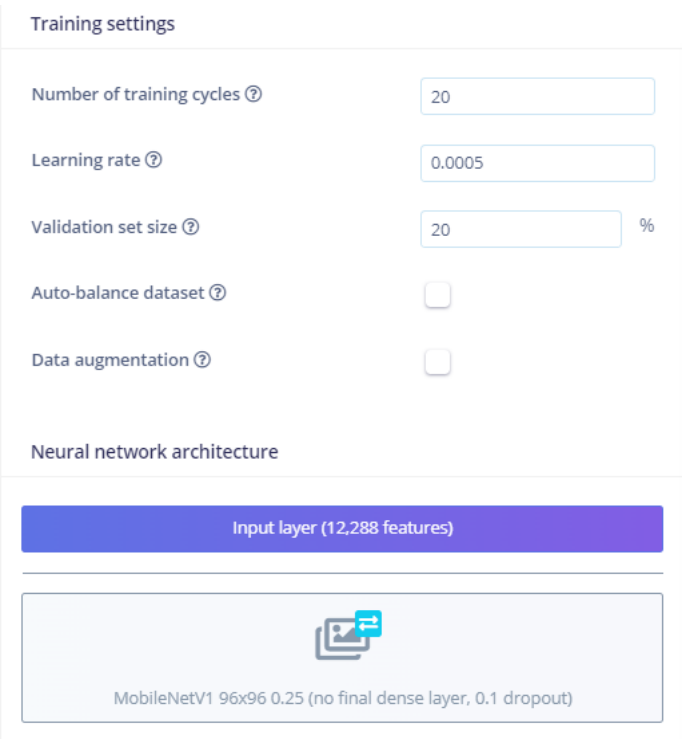


Imagem 13 - Configurações da rede neural

Com esse modelo, a inferência aconteceu corretamente no kit de desenvolvimento. Como esse foi o modelo final, a próxima seção será dedicada a comentar sobre os seus resultados e deploy.

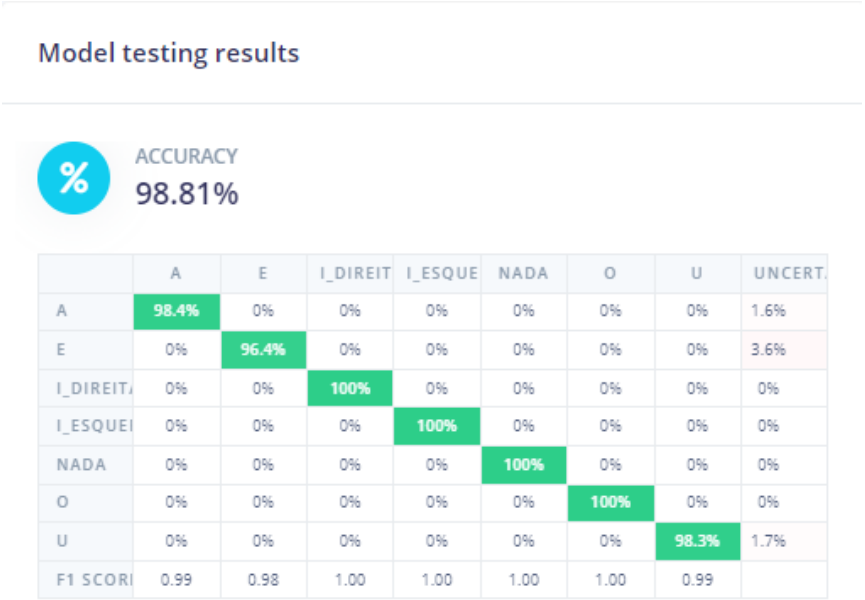
3. Resultados e Deploy

Os resultados obtidos para o modelo final foram os seguintes.

	A	E	I_DIREITA	I_ESQUER	NADA	O	U
A	100%	0%	0%	0%	0%	0%	0%
E	0%	100%	0%	0%	0%	0%	0%
I_DIREITA	2.6%	5.3%	92.1%	0%	0%	0%	0%
I_ESQUERDA	0%	0%	4.5%	95.5%	0%	0%	0%
NADA	0%	0%	0%	0%	100%	0%	0%
O	0%	0%	0%	0%	0%	100%	0%
U	0%	0%	0%	0%	0%	0%	100%
F1 SCORE	0.99	0.97	0.93	0.98	1.00	1.00	1.00

Imagem 14 - Resultados do modelo final (Treinamento)

Imagem 15 - Performance do modelo final



	A	E	I_DIREIT	I_ESQUE	NADA	O	U	UNCERT.
A	98.4%	0%	0%	0%	0%	0%	0%	1.6%
E	0%	96.4%	0%	0%	0%	0%	0%	3.6%
I_DIREITA	0%	0%	100%	0%	0%	0%	0%	0%
I_ESQUE	0%	0%	0%	100%	0%	0%	0%	0%
NADA	0%	0%	0%	0%	100%	0%	0%	0%
O	0%	0%	0%	0%	0%	100%	0%	0%
U	0%	0%	0%	0%	0%	0%	98.3%	1.7%
F1 SCORE	0.99	0.98	1.00	1.00	1.00	1.00	0.99	

Imagem 16 - Resultados do modelo final (Teste)

Como mostrado acima, a acurácia, tanto no teste quanto no treinamento, foram maiores que 98%. Esse valor é bem alto e só foi alcançado porque foram utilizadas imagens que tinham um padrão de iluminação e disposição bem definidos. Esses padrões foram utilizados para que fosse possível criar um modelo que funcionasse corretamente no Arduino Nano 33 BLE Sense.

O tempo de inferência mostrado no Edge Impulse também está dentro das expectativas, visto que 552 ms é um tempo capaz de apresentar a inferência dos símbolos de maneira convincente.

Em relação ao uso de 90,2K de RAM, foram encontrados problemas ao realizar o deploy no Arduino. Mesmo que essa quantidade devesse ser o suficiente para o bom funcionamento no kit, foi necessário desligar o EON, diminuindo assim para 76,2K de RAM. Com isso, o deploy e as inferências funcionaram. A inferência em tempo real pode ser vista com detalhes no vídeo de apresentação.

4. Conclusão

Ao final do desenvolvimento, obteve-se um modelo efetivo e com boa acurácia para classificar as vogais em LIBRAS tanto na plataforma Edge Impulse quanto no Arduino Nano 33 BLE Sense. Os problemas encontrados foram solucionados a partir da análise dos dados juntamente com os conhecimentos adquiridos durante o curso.

O ponto mais interessante desse projeto foi a construção do modelo juntando vários materiais já disponibilizados online, além de valer como um meio para divulgar tanto TinyML quanto LIBRAS.

De forma a dar essa continuidade ao projeto e ao conhecimento em si, o capítulo seguinte apresenta algumas sugestões que foram notadas pelo grupo na medida em que o projeto foi desenvolvido.

5. Sugestões para trabalhos futuros

Como possuíamos algumas limitações de hardware, o projeto não pôde ser expandido. Em vista disso, algumas sugestões de trabalhos futuros são as seguintes:

- Adicionar letras restantes ao modelo, de modo que a latência e o consumo de memória fiquem dentro dos requisitos da placa;
- Adicionar símbolos que representam palavras;
- Desenvolvimento de uma extensão que permita ver a imagem e selecionar o momento exato para tirá-la;
- Melhora no dataset para permitir que as imagens sejam mais heterogêneas.

6. Links

Link para o vídeo de apresentação no YouTube:
https://www.youtube.com/watch?v=0Sqt4lTYs6w

Link para o projeto público no Edge Impulse:
https://studio.edgeimpulse.com/public/116471/latest