Treinando uma CNN no Edge Impulse

Introdução

Agora que descobrimos como funcionam as redes neurais convolucionais (CNNs), vamos treinar uma! Usaremos o mesmo conjunto de dados do primeiro módulo para treinar um modelo CNN no Edge Impulse, em vez de uma simples rede neural densa.

Convertendo Images

No momento, o Edge Impulse aceita apenas imagens .png e .jpg. Se você coletou originalmente seu conjunto de dados como arquivos .bmp, precisará convertê-los para uma dessas duas categorias. Eu recomendo .png, pois é um formato sem perdas.

Windows: use o <u>Bulk Image Converter</u>
 Mac: use built-in tools ou XnConvert

Linux: <u>ImageMagick</u>

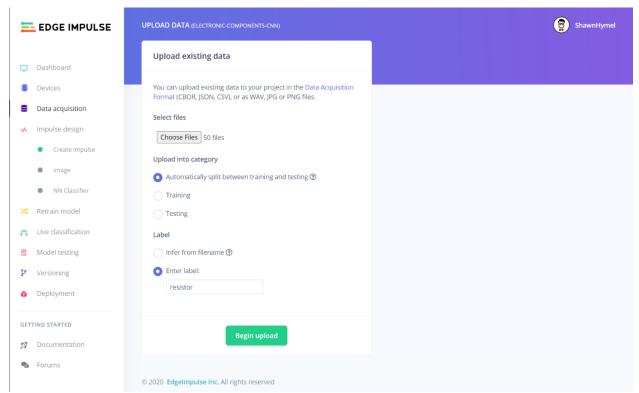
Ou, se estiver particularmente entusiasmado, você pode escrever um script Python rápido (usando algo como PIL) para realizar o trabalho.

Eu recomendo manter a mesma estrutura de diretórios para que você possa determinar a classe de cada amostra apenas olhando o nome da pasta que a contém.

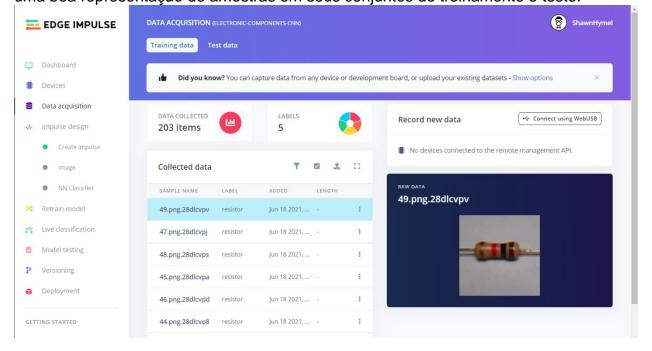
Carregando o Dataset

A boa notícia é que podemos fazer upload de dados de imagem diretamente para o Edge Impulse, em vez de precisar extrair recursos manualmente usando o Colab.

Inicie um novo projeto Edge Impulse. Vá para **Aquisição de dados**. Clique no botão para fazer upload dos dados existentes. Clique em **Escolher arquivos** e selecione todas as imagens de uma classe (por exemplo, classe de resistor). Deixe a opção **Dividir automaticamente** entre treinamento e teste selecionada. Para Label, insira um rótulo para sua primeira categoria (por exemplo, "resistor").



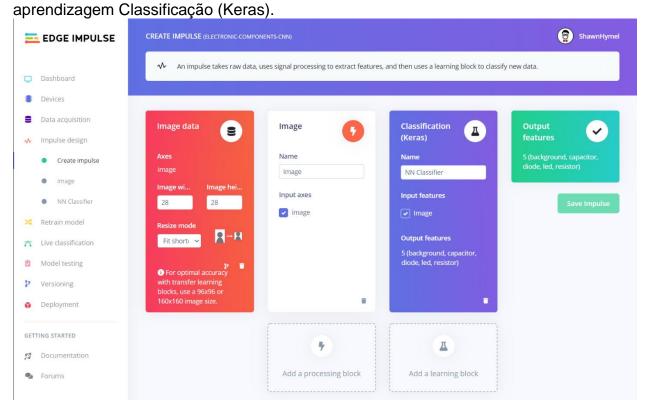
Clique em Iniciar upload. Repita esse processo para o restante das categorias do seu conjunto de dados. Quando terminar, vá para Aquisição de dados. Certifique-se de ter uma boa representação de amostras em seus conjuntos de treinamento e teste.



Criar Impulse

Vá para o design Impulse. Altere a configuração de dados de imagem para 28 x 28 (largura x altura). Se as imagens originais não forem todas quadradas, você poderá ajustar o modo Redimensionar conforme necessário para cortar ou comprimir as imagens.

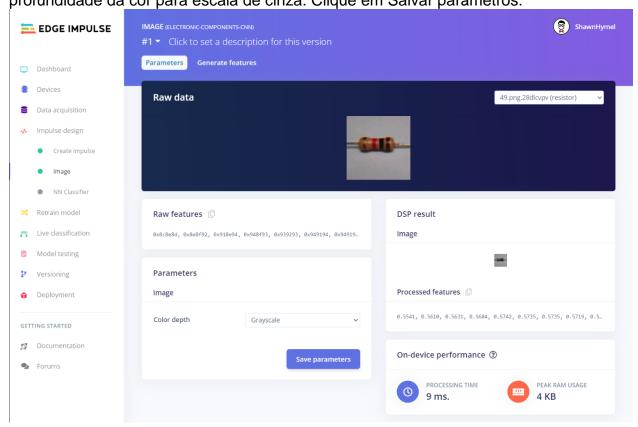
Adicione um bloco de processamento de imagem. Adicione um bloco de



Clique em Salvar Impulse.

Extração de Features

Vá para Imagem em Design Impulse na barra de navegação esquerda. Altere a profundidade da cor para escala de cinza. Clique em Salvar parâmetros.



Na próxima tela, clique em Gerar recursos. Aguarde até que o script de geração de recursos seja concluído. Aqui, as imagens estão sendo convertidas em tons de cinza e

28x28 pixels. ShawnHymel IMAGE (ELECTRONIC-COMPONENTS-CNN) EDGE IMPULSE Parameters Generate features Dashboard Devices Training set Feature explorer (203 samples) Data acquisition Data in training set 203 items X Axis ◆ Impulse design Y Axis Z Axis Visualization li v Visualization li 🗸 Visualization la 🗸 Create impulse Classes 5 (background, capacitor, diode, led, resistor) Image background capacitor diode led NN Classifier 24 Retrain model Live classification Model testing Feature generation output Mon Aug 2 22:20:01 2021 Construct embedding Still running... completed 0 / 500 epochs completed 50 / 500 epochs Versioning Deployment
 completed
 50 / 500 epochs

 completed
 150 / 500 epochs

 completed
 500 / 500 epochs

 completed
 200 / 500 epochs

 completed
 300 / 500 epochs

 completed
 300 / 500 epochs

 completed
 350 / 500 epochs

 completed
 400 / 500 epochs

 completed
 450 / 500 epochs

 completed
 450 / 500 epochs
 GETTING STARTED Documentation completed 450 / 500 epochs

Completed 450 / 500 epochs

Mon Aug 2 22:20:03 2021 Finished embeddin

Reducing dimensions for visualizations OK Forums On-device performance ② PROCESSING TIME PEAK RAM USAGE

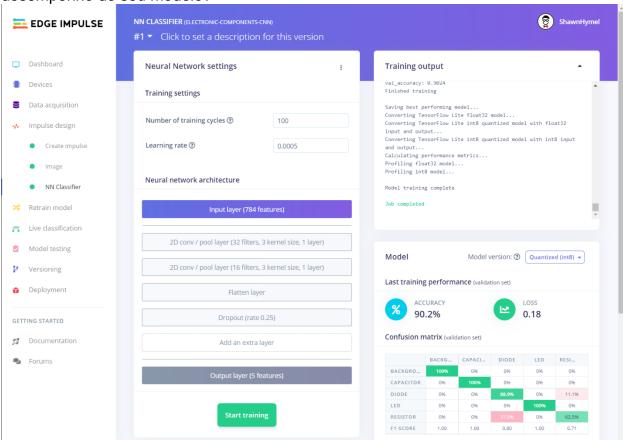
Trainamento!

Vá para Classificador NN. Recomendo alterar o Número de ciclos de treinamento para algo entre 100 e 200 (pois estamos trabalhando com um conjunto de dados pequeno). Clique em Iniciar treinamento. Quando o treinamento é concluído, qual foi o

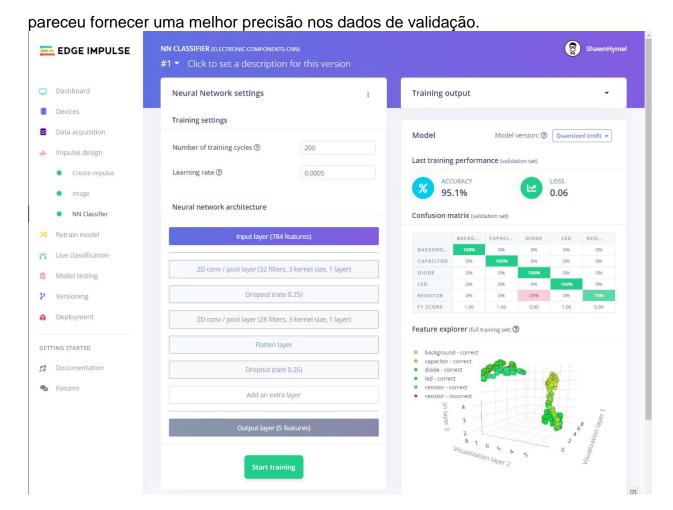
9 ms.

4 KB

desempenho do seu modelo?



Tente ajustar os hiperparâmetros do modelo, como número de ciclos de treinamento, número de filtros e tamanhos de kernel. Você pode tentar adicionar e remover camadas. Por exemplo, adicionei uma camada de eliminação após a primeira camada de convolução e alterei a segunda camada de convolução para ter 28 filtros. Isso



Conclusão

Comparar os resultados de precisão deste modelo com os do exemplo da rede neural densa pode ser enganoso. Lembre-se de que estamos trabalhando com um conjunto de dados relativamente pequeno (50 amostras por classe) e as amostras provavelmente estão distribuídas aleatoriamente entre conjuntos de treinamento e teste. Mover uma única amostra para dentro ou para fora do conjunto de treinamento pode fazer com que os resultados da precisão da validação oscilem 0,5% (assumindo 200 amostras no conjunto de treinamento). Um conjunto de dados maior ajudaria a evitar o overfitting e lhe daria uma ideia muito melhor da viabilidade do modelo. Exploraremos maneiras de criar conjuntos de dados maiores em uma palestra e projeto futuro.