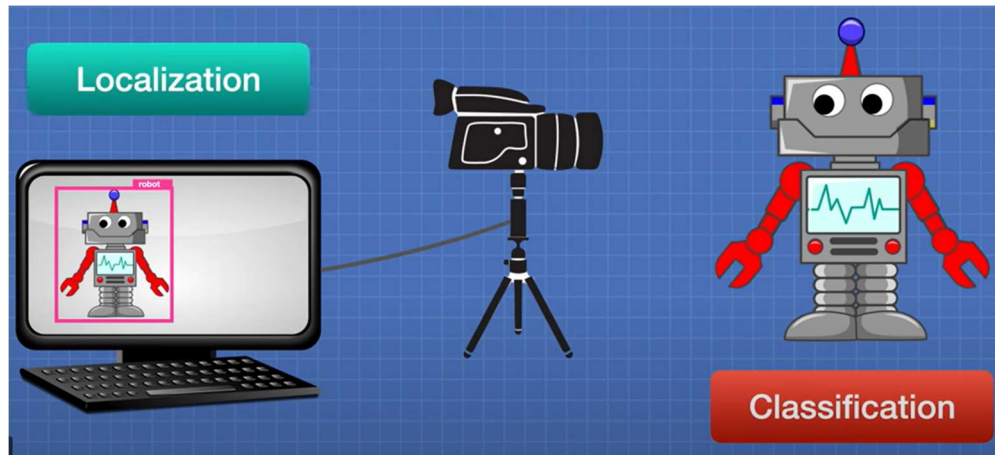


Universidade Federal de Goiás
Escola de Engenharia Elétrica, Mecânica e de Computação
Laboratório de Inovação e Automação 1 – Projetos de Inteligência Artificial

Detecção de Objetos usando Esp32-Cam

1. Detecção de Objetos

A visão computacional é um campo de estudo fascinante que automatiza o processo de atribuição de significado a imagens ou vídeos digitais. Em outras palavras, busca-se ajudar os computadores a enxergarem e entenderem o mundo ao nosso redor. Vários algoritmos e técnicas de aprendizado de máquina podem ser usados para realizar essas tarefas e, à medida que isso se torna mais rápido e eficiente, podemos implementar essas técnicas em sistemas embarcados. Por fim, a detecção de objetos é a combinação de duas técnicas diferentes de uso de imagem: 1) classificação de imagem; 2) localização de imagem.

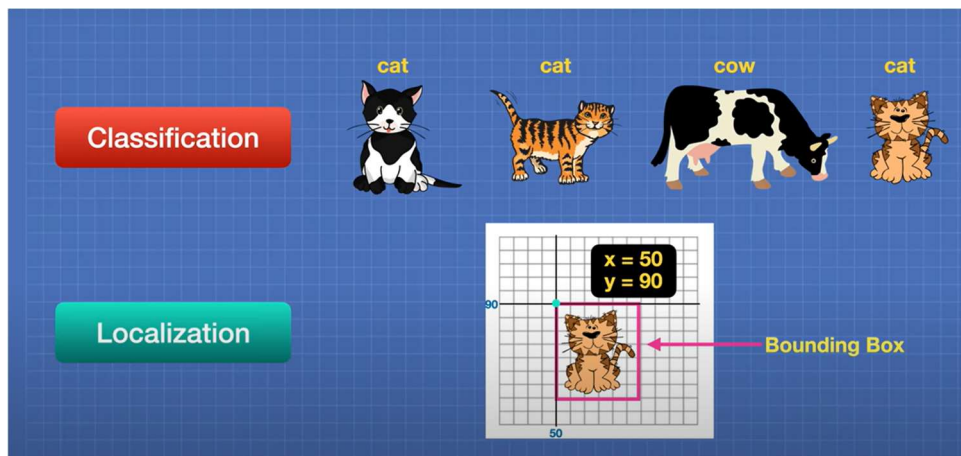


1.1 Classificação de Imagem

Contempla determinar a classe pertencente da imagem. A figura abaixo mostra duas classes de animais detectadas: gato e vaca. *São 3 gatos e 1 vaca!*

1.2 Localização de Imagem

Contempla determinar a posição do objeto, dentro do campo de visão, usando uma caixa delimitadora (do inglês: bounding box) que circula o objeto, apresentando as coordenadas X e Y, do canto superior esquerdo. *Um gato foi localizado na imagem acima!*

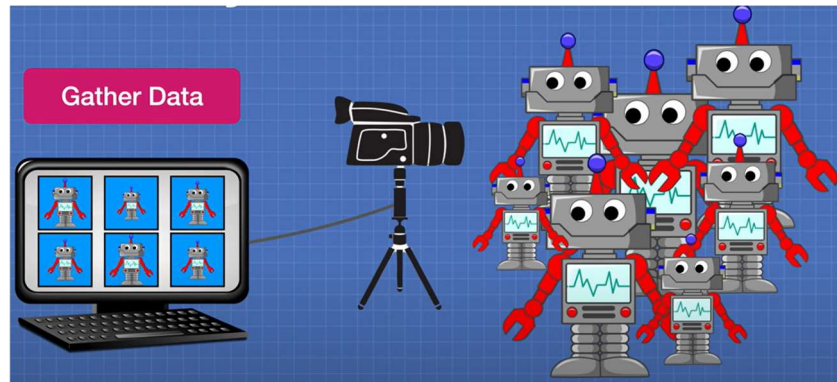


2. Passos Básicos para o Primeiro Modelo

O fluxo de trabalho para um projeto completo contempla alguns passos básicos.

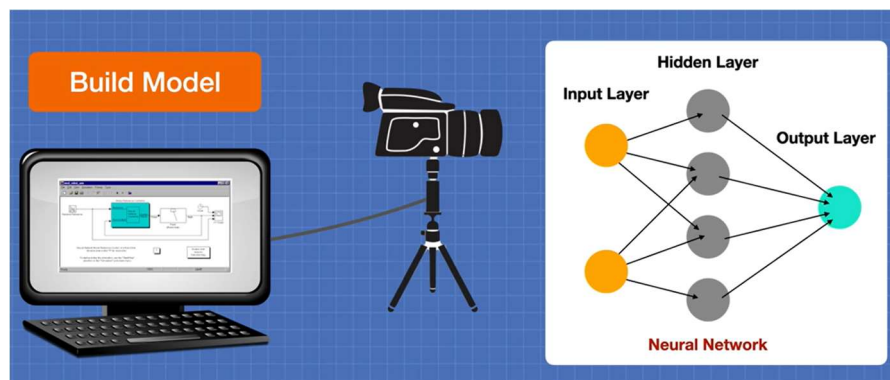
2.1. Primeiro passo – Coletar e Rotular Imagens

Coletar um conjunto de imagens do objeto a ser detectado. Pode-se usar um equipamento para capturar as imagens ou um banco de dados pronto com as referidas imagens de objetos.



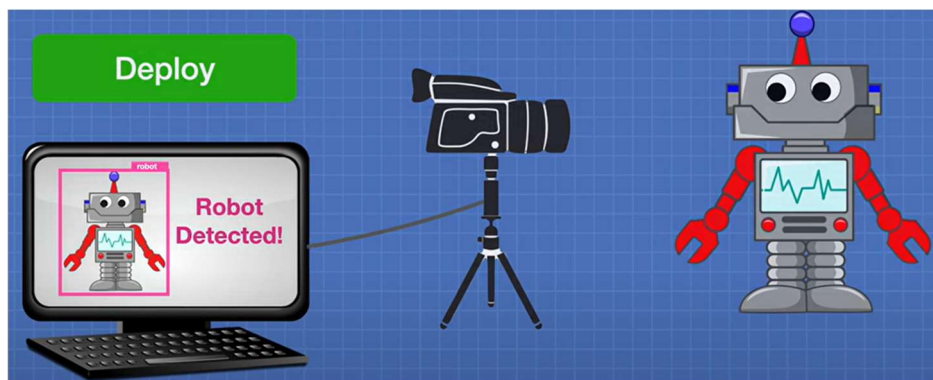
2.2. Segundo passo – Construir o Modelo

Baseado numa rede neural, determinar a resposta para a detecção de um objeto apresentado. As imagens são separadas em treino e teste, numa proporção de 80 e 20%, respectivamente.



2.3. Terceiro passo – Deploy (Implantar) o Modelo

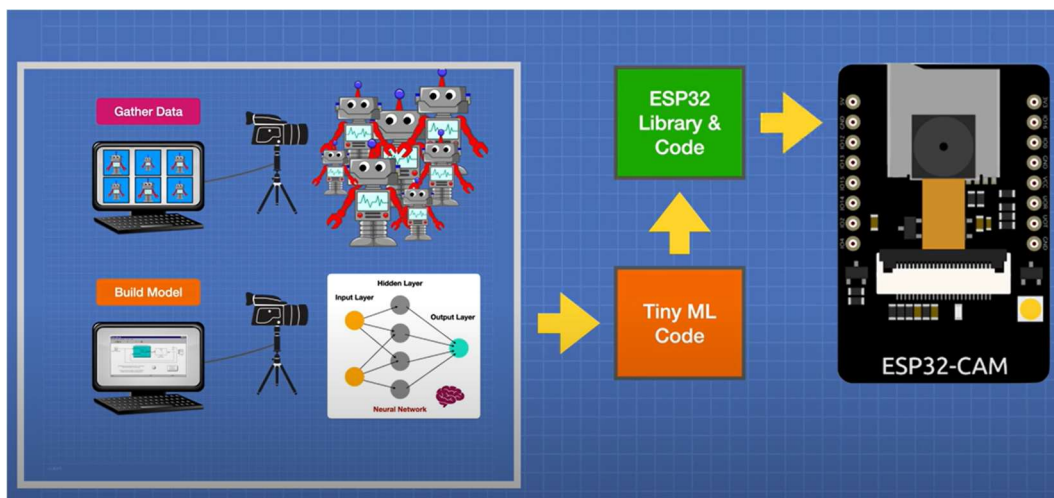
Liberar para produção, ou seja, detectar objetos apresentados no campo de visão do sistema.



3. Modelo Embarcado para Microcontrolador (Tiny ML)

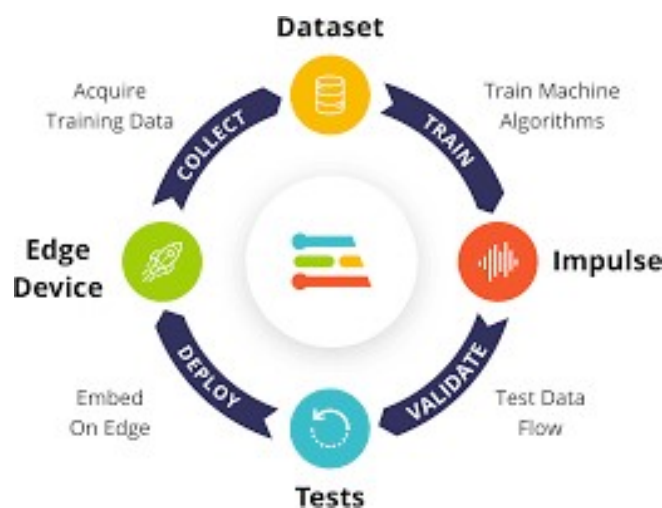
Os projetos de detecção de objetos tradicionais demandam muito consumo de hardware, como CPU ou GPU, exigindo altos investimentos financeiros e de infraestrutura. Existe uma alternativa para projetos menores e mais simples. O Modelo Embarcado de Aprendizado de Máquina (do inglês: Embedded ML) é específico para aplicações em um microcontrolador, sendo uma solução viável para projetos de IoT.

As tarefas de obtenção/rotulação de imagens e a criação do modelo podem ser realizadas em ambientes on-line gratuitos. Esses ambientes geram um modelo simplificado, chamado de Tiny ML Code. Esse modelo criado é exportado para um formato de código, construindo assim uma biblioteca utilizável no Esp32 ou Esp32-Cam, denominada de Esp32 Library & Code.



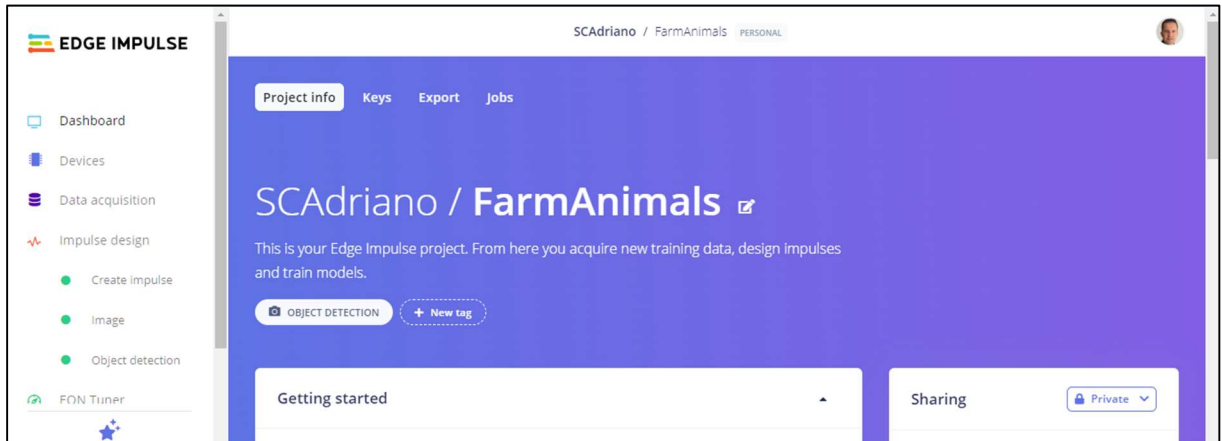
4. Edge Impulse – online e gratuito

A plataforma Edge Impulse (edgeimpulse.com) possibilita gratuitamente a criação de um conjunto de dados (dataset), treinamento, construção, implantação (deploy) e aplicação final de um modelo de aprendizado de máquina, por meio da criação de bibliotecas para execução em um microcontrolador, conhecidos como Tiny ML.

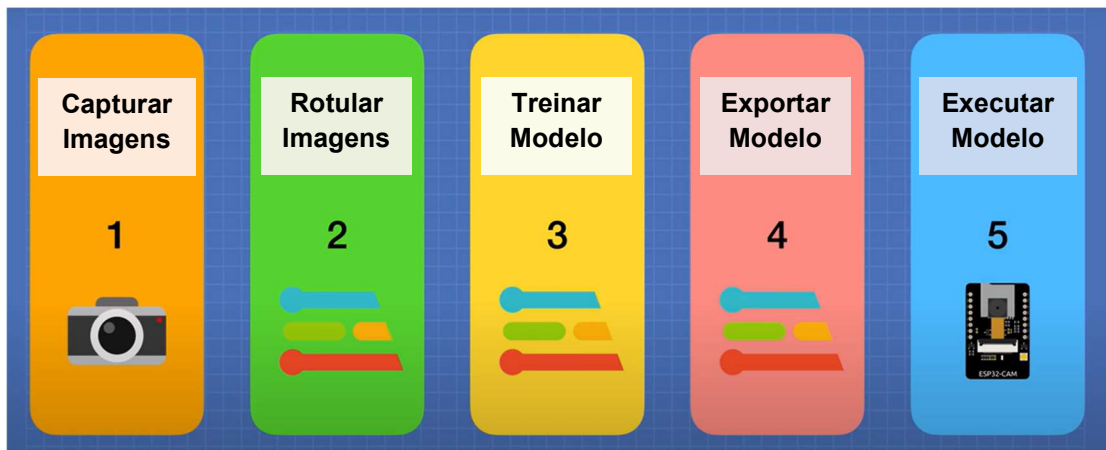


4.1 Acessando o Edge Impulse

Crie uma conta! Acesse o sistema e estará em seu primeiro projeto de EML (Embedded Machine Learning), coletando novos dados, projetando e treinando modelos. Aproveite!



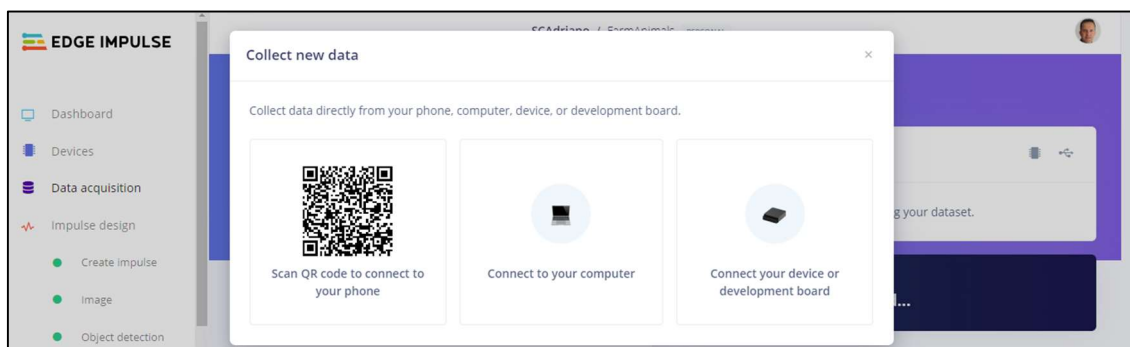
O fluxo de trabalho para um projeto completo contempla cinco passos básicos.



4.1.1 Capturar Imagens

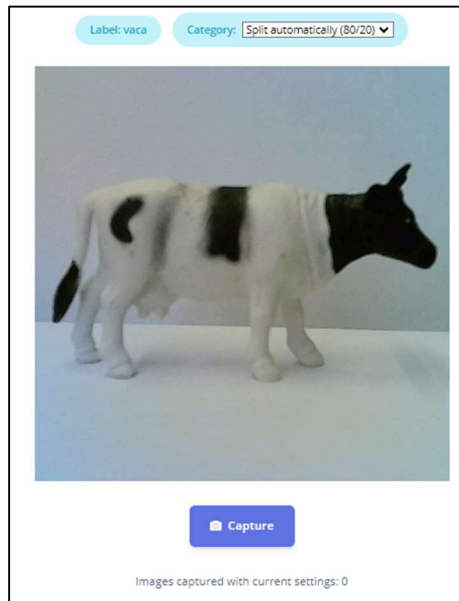
Para capturar as imagens, clique em < **Data acquisition** >, depois em < **Connect a device** > e selecione a fonte de imagens, podendo ser de celular, computador ou outro dispositivo.

Após a fonte de captura ser reconhecida e validada, clique em < **Collecting images?** > e na sequência nas demais opções para habilitar o devido acesso à fonte de imagens.



Se uma webcam estiver ligada, apresentando a imagem de um objeto (vaca) para ser coletada, realize os próximos passos:

- 1) Clique em < **Label:** ... > para nomear a imagem a ser capturada para “vaca”.
- 2) Selecione em < **Category:** ... > a opção desejada para a divisão das imagens entre “Treino e Teste” ou deixe no modo automático (Split automatically (80/20)).
- 3) Clique em < **Capture** > para capturar a referida imagem.

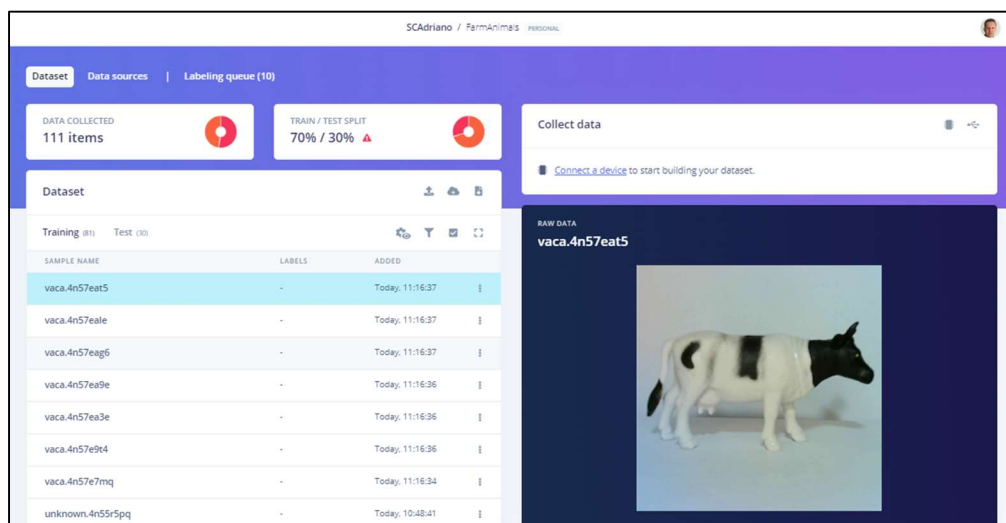


Dica: mantenha o objeto no centro da imagem sempre!

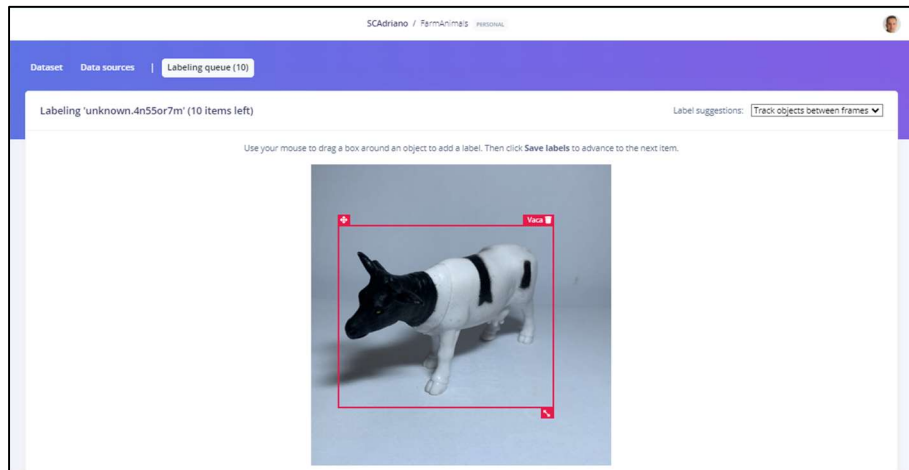
Observe que na parte de baixo, há um contador de imagens capturadas na configuração atual.

4.1.2 Rotular Imagens

Na tela principal, em < **Data acquisition** >, observa-se o **Dataset** e o **Labeling queue**.



Para rotular as imagens, clique em **< Labeling queue >**, escreva o **“label”** da imagem e enquadre a imagem, se necessário. *Faça isso para todas as imagens!*

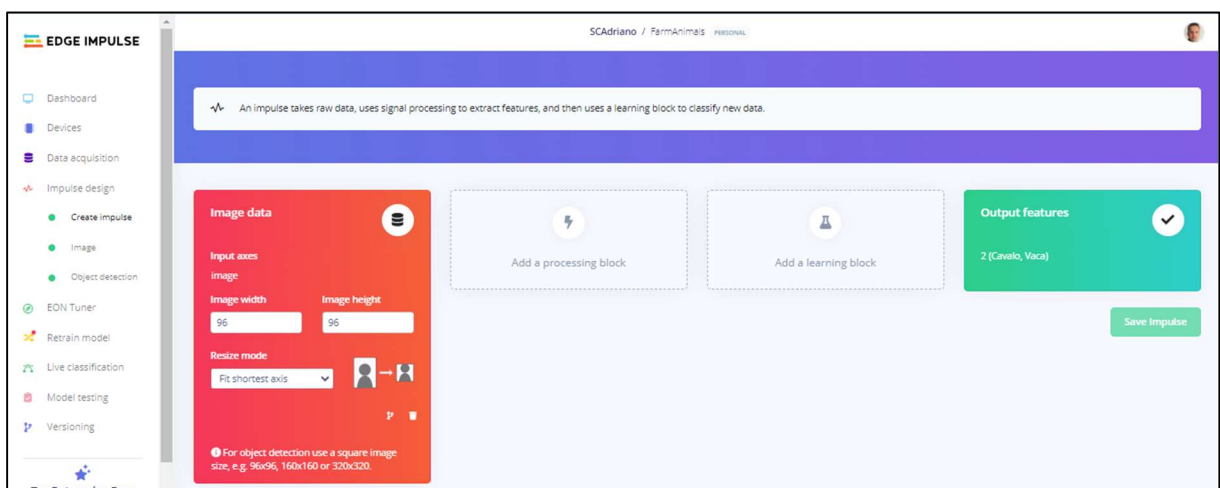


4.1.3 Treinar o modelo

Para treinar o modelo, em **Impulse design**, realize os seguintes passos:

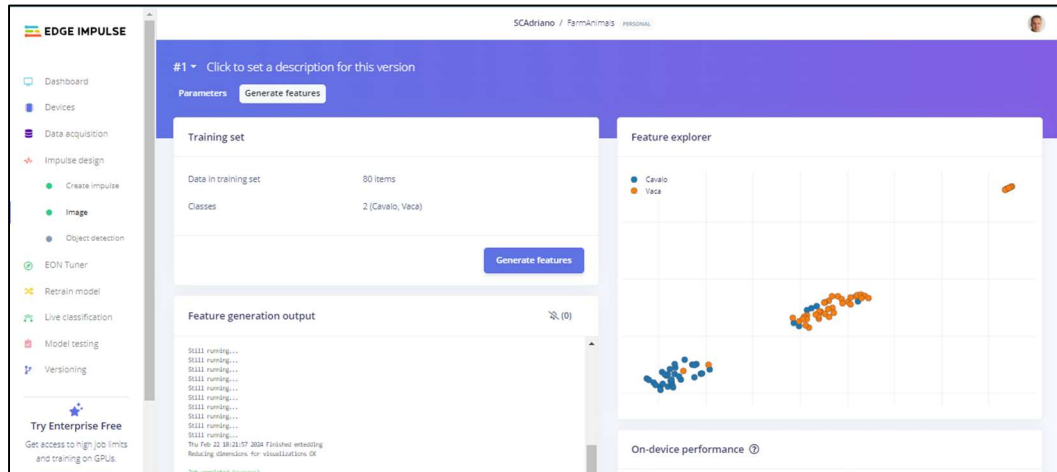
a) Clique em **< Create impulse >**:

- Na primeira caixa, mantenha as informações padrão em **< Image data >**.
- Na segunda caixa, clique em **< Add a processing block >** e depois em **< Add >**.
- Na terceira caixa, clique em **< Add a learning block >**, na tela seguinte, em Edge Impulse, clique em **< Add >**.
- Na quarta caixa, confira a quantidade de classes, clique em **< Save Impulse >**.



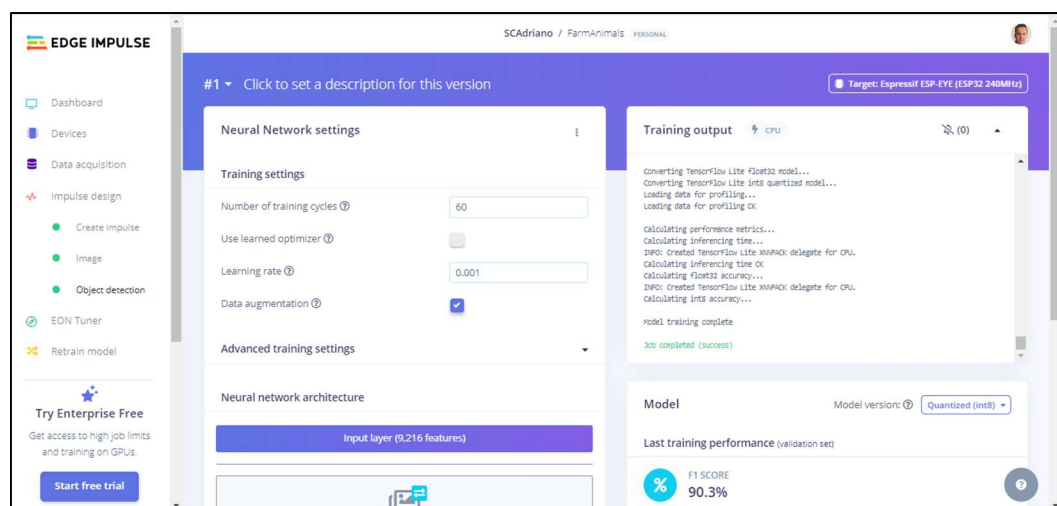
b) Clique em **< Image >**:

- Na primeira tela, em **< Color depth >**, selecione **< Grayscale >** e clique em **< Save parameters >**.
- Na segunda tela, clique em **< Generate features >**.
- Por fim, observe em **Feature explorer** a distribuição das classes.



c) Clique em **< Object detection >**:

- Clique em **< Choose a different model >** e selecione FOMO (Faster Objects, More Objects) MobileNetV2.0.1, clicando no respectivo **< Add >**.
- No alto à direita, selecione o microcontrolador - **Target: Expressif ESP....**
- Por fim, clique no botão **< Start training >**.

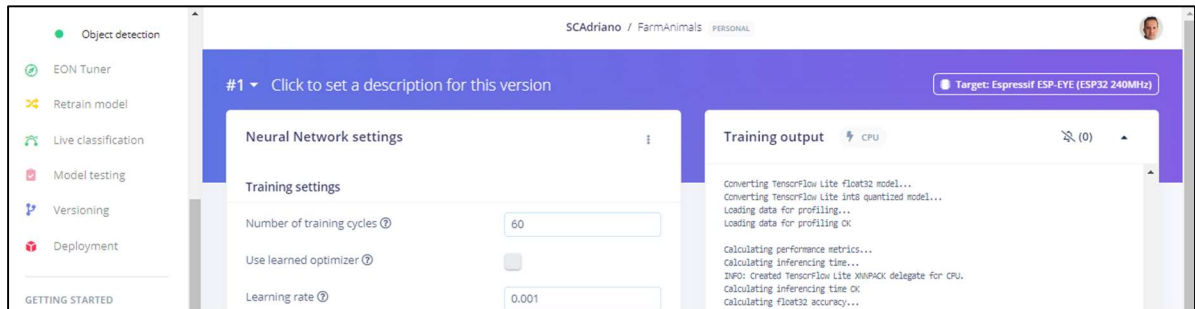


Aguarde o modelo ser treinado! Pronto.

No exemplo do material, na imagem acima, o modelo está treinado e apresenta a métrica F1 Score em 90,3%. Está boa, mas poderia ser melhor! O ideal é 100%.

5. Exportar o modelo

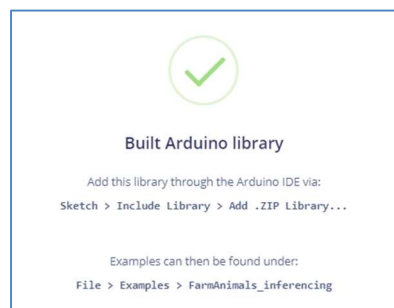
Para exportar o modelo para um microcontrolador, clique em **< Deployment >**.



Na tela principal, na lupa, selecione **< Arduino library >** e depois clique no botão **< Build >**.



Um arquivo da biblioteca será criado, em um arquivo do tipo zipfile e baixado no computador.



6. Carregar a biblioteca na IDE do Arduino

A biblioteca criada pelo Edge Impulse para o Esp32 deve ser adicionada na IDE do Arduino.

Abra a IDE do Arduino, clique em **Sketch > Include Library > Add. ZIP Library...**

Para executar o projeto, acesse **< File >**, **< Examples >**, **< Nome da biblioteca gerada >**, **< esp32 >** e **< esp32_camera >**.

Por fim, faça o upload do projeto no Esp32-Cam. Apresente ao Esp32-Cam, objetos identificáveis pelo modelo criado. Avalie a acurácia de cada objeto apresentado.

Se você chegou até aqui, Parabéns! Você sabe fazer projeto de Visão Computacional!