

<b>Nome:</b>	<b>nUSP:</b>
Allan Baldissin	8657904
Danilo Alves	10408390
Marcela Tiemi Shinzato	10276953

## **SSC0740 - Sistemas Embarcados**

**Professor: Vanderlei Bonato**

### **Câmera de segurança com TensorFlow Lite**

#### **Explicação do modelo Tensor Flow Lite**

O TensorFlow Lite é uma biblioteca que provê uma rede neural, que pode ser treinada ou não, para classificar imagens usando inteligência artificial. O processo de aprendizado consiste em alimentar o algoritmo com imagens rotuladas e o algoritmo deve extrair quais características dessas imagens as fazem serem rotuladas como tal. Por exemplo, ao se apresentar imagens de casas, dizendo que são casas, o algoritmo deve encontrar os padrões dessas imagens, e, quando se apresentar uma imagem de teste, ele dirá o quanto essa imagem se assemelha ao conceito que ele assimilou como casa. O TensorFlow Lite possui muitas classes de imagens treinadas, e, ao embarcá-lo no Android, ele exibe as 3 classes com maior similaridade à imagem de teste, juntamente com suas porcentagens de similaridade.

#### **Aplicação escolhida**

Na presente aplicação, usaremos a rede já treinada Mobile Net V2 para implementar um algoritmo que detecta a presença de algum ser humano na câmera de um celular Android e, se for encontrado, envia uma mensagem ao dono do dispositivo via app Whatsapp. Dessa forma, configura-se um sistema de segurança, no qual o usuário poderia deixar a câmera ligada para vigiar uma área, a qual espera-se que não seja adentrada por pessoas.

#### **Desenvolvimento da aplicação**

Para a construção do algoritmo, usou-se o *software* Android Studio e o código do TensorFlow presente em <<https://github.com/tensorflow/examples>>. O celular Android deve estar conectado a internet e, assim, torna-se possível usar a aplicação no sistema embarcado.

Conforme solicitado, o modelo original foi usado no Google Colab, e convertido para uso no Android usando a função de conversão da classe TFLiteConverter. Para fazer a comunicação, criou-se um servidor em Node.JS no qual foi implementada a comunicação com a API Twilio para que as mensagens fossem enviadas para o Whatsapp. O servidor funciona na porta 3000 e precisa de credenciais do Twilio nas variáveis do ambiente da máquina onde está rodando e também credenciais do protocolo SSL.

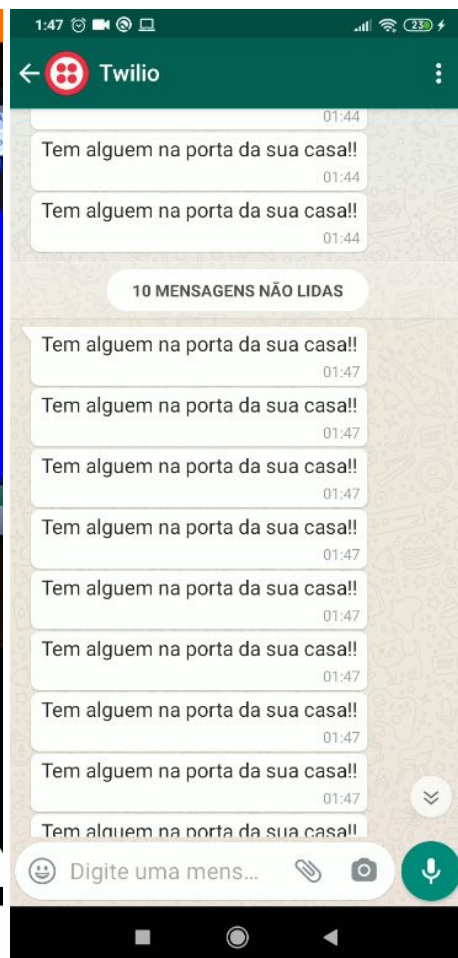
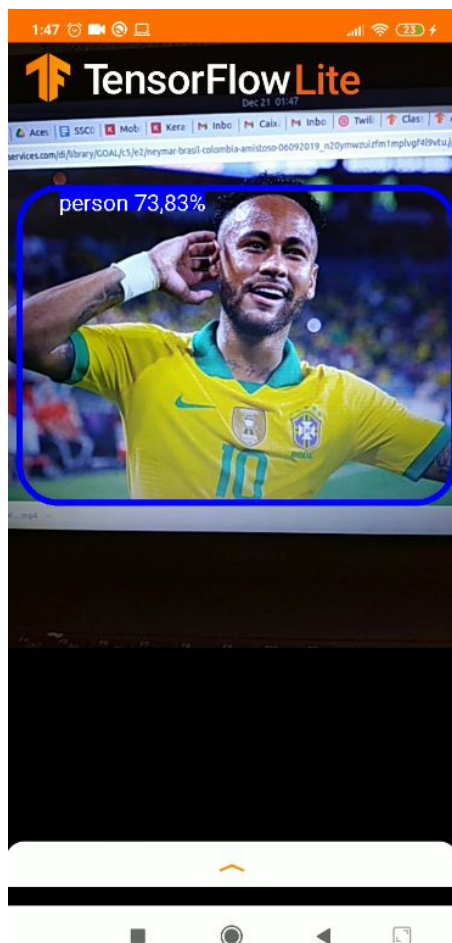
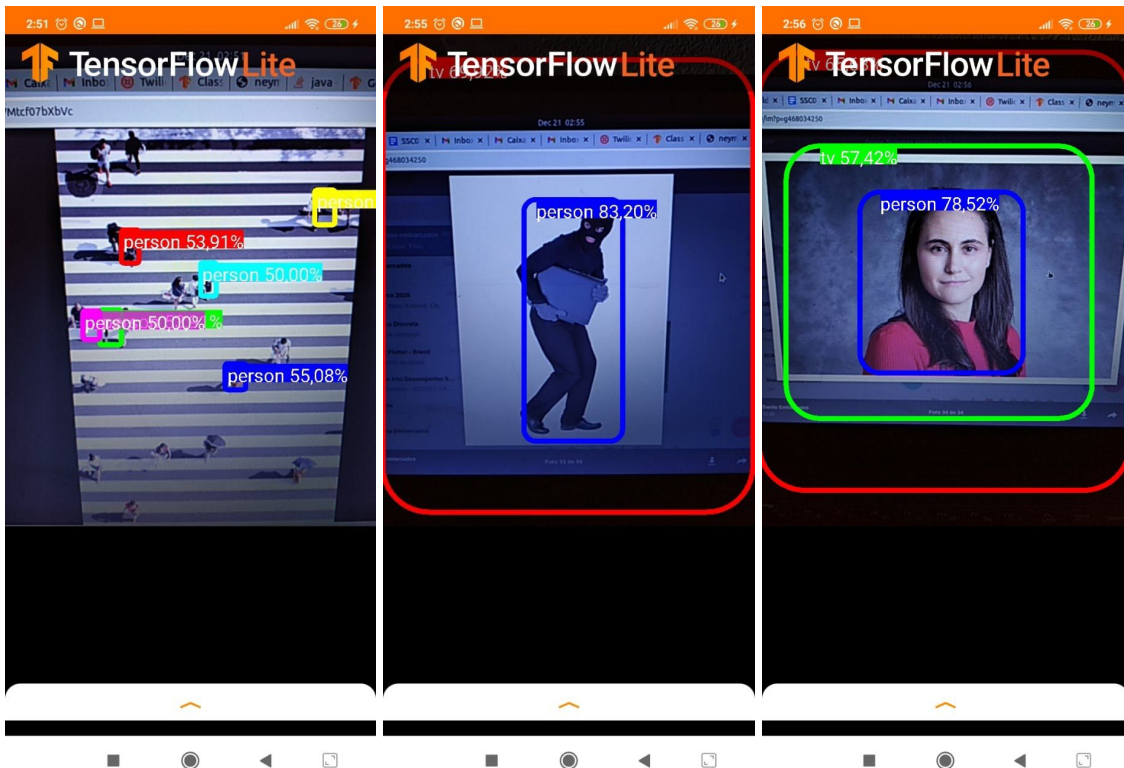
Assim, sempre que a aplicação identifica uma pessoa, é enviado um request para o servidor e uma mensagem é enviada no Whatsapp do número descrito no código.

## Desempenho

No Google Colab em computadores, usando imagens estáticas, os resultados não foram favoráveis. Por exemplo, uma foto caricata de um ladrão retornou “máscara de esqui” com precisão 78,9904%; uma foto de Mark Zuckerberg retornou rótulos com baixa precisão (a maior foi 9,264664% para “terno”); uma silhueta masculina retirada de uma câmera de segurança retornou “aspirador de pó” com 30,389252%; a foto de uma mulher retornou “estola” com precisão 39,878854%.



No entanto, ao embarcar a aplicação no celular Android, ambos modelos (original e convertido) funcionaram corretamente: identificaram pessoas e enviaram as mensagens de aviso no Whatsapp. A seguir, mostramos capturas de tela do app:



Nota-se uma precisão superior a 50% em todos os casos, rapidamente detectada, e também nota-se que as mensagens foram enviadas imediatamente.

### **Dificuldades encontradas**

O software Android Studio exige muita memória RAM, portanto não foi usado por todos os membros do grupo.

A classificação de imagens se mostrou diferente usando imagem estática como entrada no Google Colab e pela aplicação Android, e o motivo dessa diferença não foi identificado. Suspeitamos que na aplicação Android exista um pré processamento da imagem que possa melhorar a identificação, ou que o nosso modelo não esteja sendo carregado devidamente na aplicação Android.