

Reconhecimento de objetos com TensorFlow Lite

Leonardo Alves Paiva 10276911

Lucas Fernandes da Nobrega 9805320

João Pedro Doimo Torrezan 9806933

Índice

Introdução

Tensor Flow

Materiais e Métodos

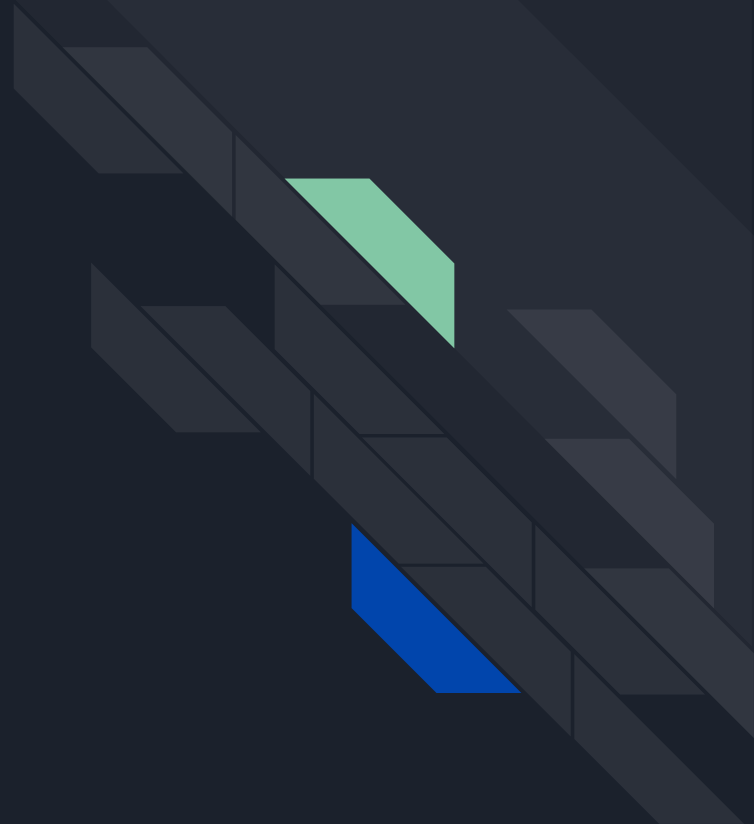
Modelos

Análise

Reconhecimento

Performance

Conclusão





Introdução

TensorFlow

- Criado pelo Google
- Treinamento de redes neurais e deep learning
- Python e C++



TensorFlow Lite

- Aplicação em dispositivos da borda (embarcados e IoT)
 - Latência
 - Privacidade
 - Conectividade
 - Consumo de Energia
- 2 componentes principais
 - Interpretador
 - Conversor
- Aplicativo Mobile para classificação

Materiais e Métodos

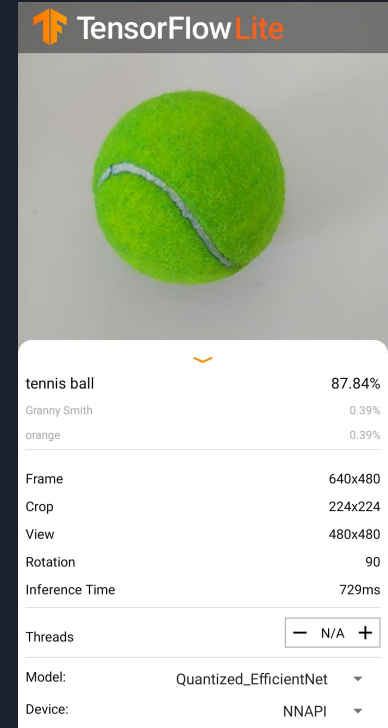
- Smartphone Xiaomi Redmi Note 8 - Android 10

- TFL Classify

- Objetos para análise:

1. Caneta esferográfica;
2. Raquete de Tênis;
3. Tênis;
4. Bola de tênis;
5. Controle remoto;
6. Garrafa PET;
7. Mouse;
8. Fósforo;
9. Régua; e
10. Banana.

- https://drive.google.com/file/d/1lrhDxiPivSMaN_31HfAl_GdkfXEmUpgv/view?usp=sharing





Modelos

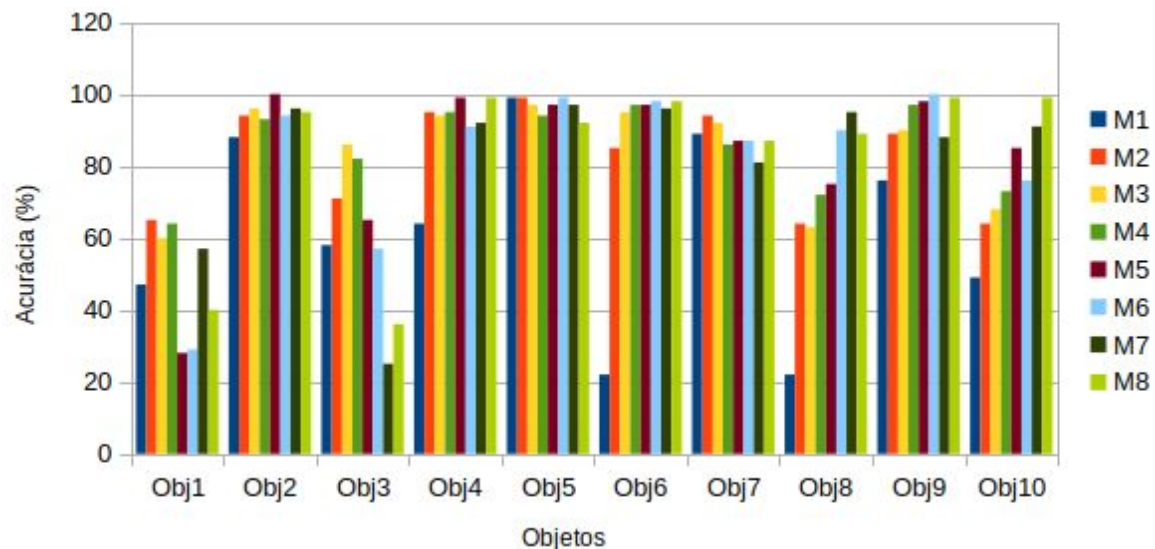
- MobileNetV1
 - Convolução de redes neurais artificiais
 - Maximizar a precisão eficientemente
 - Consideram as restrições de recursos para um aplicação embarcada
 - Modelos pequenos, com baixa latência e baixo consumo de energia
- EfficientNetLite
 - Aumento da escala de CNNs
 - Dimensiona uniformemente cada dimensão com um conjunto fixo de coeficientes de escala
 - Superou com eficiência de até 10 vezes (menor e mais rápido) a precisão dos melhores modelos de então
- Modelos Quantizados
 - Menor precisão, como números inteiros de 8 bits, em vez de flutuantes de 32 bits
 - Uma precisão menor é um requisito para aproveitar determinado hardware.

Análise

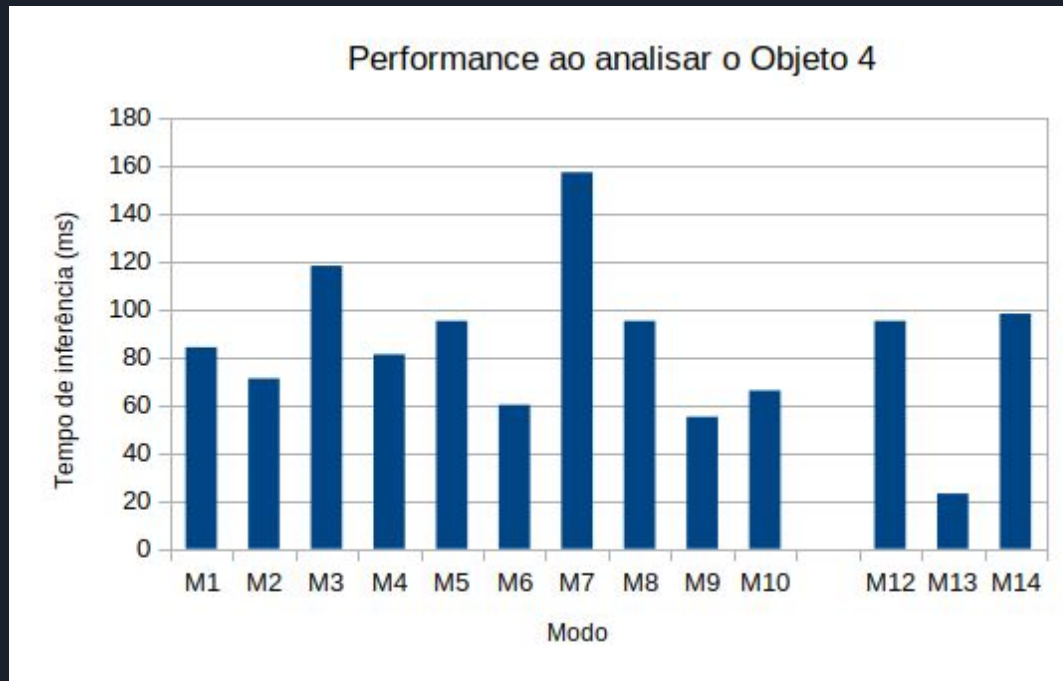
	Modo													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Thread	1	8	1	8	1	8	1	8	N/A	N/A	N/A	N/A	N/A	N/A
Modelo	Quantized Efficient Net	QE Net	Float Efficient Net	FENet	Quantized MobileNet	QM Net	Float Mobile Net	FM Net	FE Net	FM Net	QE Net	QE Net	FENet	FENet
Disp.	CPU	CPU	CPU	CPU	CPU	CPU	CPU	CPU	GPU	GPU	NN API	NN API	NN API	NN API

Análise

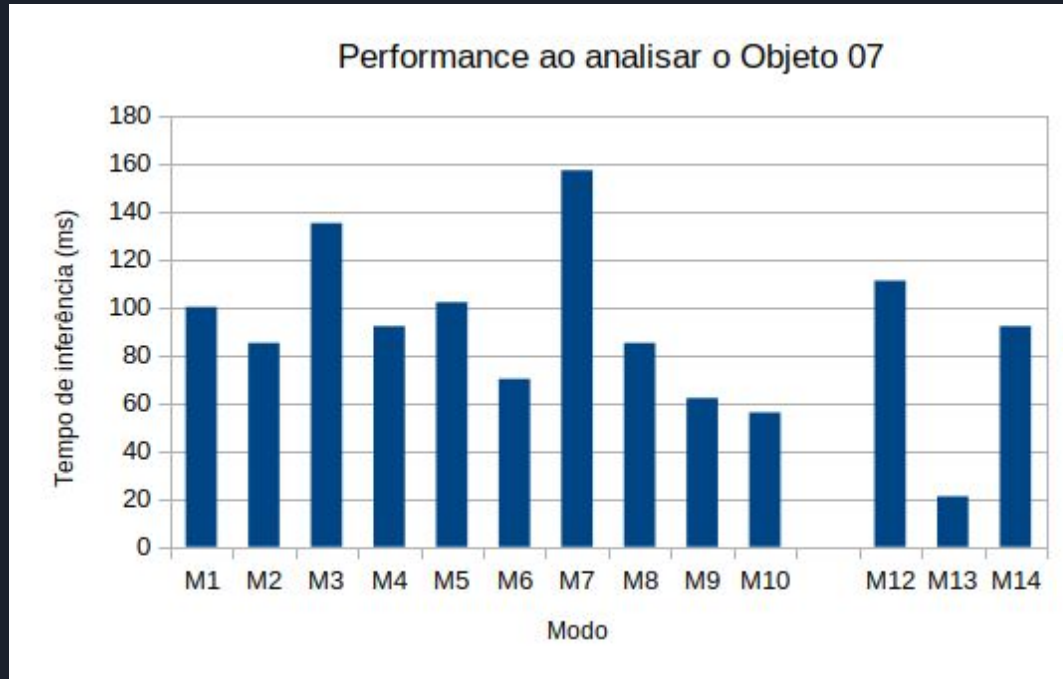
Valor de Ativação dos Modos Testados



Análise



Análise





Dificuldades encontradas

- Instalação e atualização dos softwares
 - SDK
 - Android Studio
 - TFL Classify
- Software muito sensível à variações na imagem
 - Iluminação
 - Movimento
 - Distância do objeto
 - Outros objetos
 - Não tira foto, foi tirado *printscreen*



Conclusão

- Melhor modo foi o M13
 - NNAPI
 - API para processamento de rede neural do Android
 - Otimização
 - Menor tempo de inferência (aprox. 20 ms)
 - Valor de ativação comparável aos demais modos
- Sistema foi capaz de distinguir e identificar os objetos selecionados
- Número de Threads na CPU melhoram a performance
- GPU é mais adequado do que a CPU



Referências

- EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling. Google AI Blog, 2020. Disponível em: <https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html>. Acesso em: 18/12/2020
- Guia do TensorFlow Lite. TensorFlow, 2020. Disponível em: <https://www.tensorflow.org/lite/guide>. Acesso em: 15/12/2020
- MobileNets: Open-Source Models for Efficient On-Device Vision. Google AI Blog, 2020. Disponível em: <https://ai.googleblog.com/2017/06/mobilenets-open-source-models-for.html>. Acesso em: 18/12/2020