EECE-4710 "IoT and TinyML"

Image Classification – Reloaded

Cristinel Ababei

MARQUETTE UNIVERSITY

BE THE DIFFERENCE.

1

Image Classification
Introduction & Challenges

2

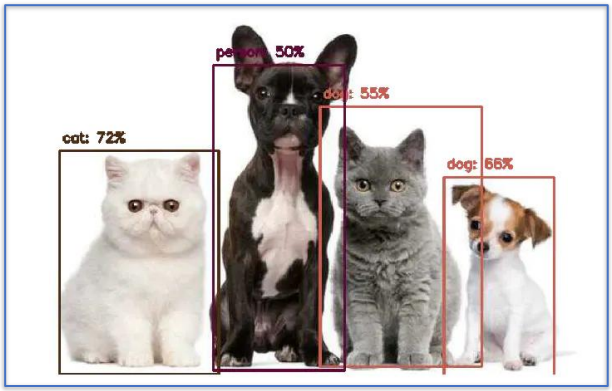
Computer Vision Main Problem Types

Image Classification
(**Multi-Class** Classification)

Cat: 70%

Dog: 80%

Object Detection
Multi-Label Classification + Object **Localization**



3

Person Detection (Visual Wake Words)

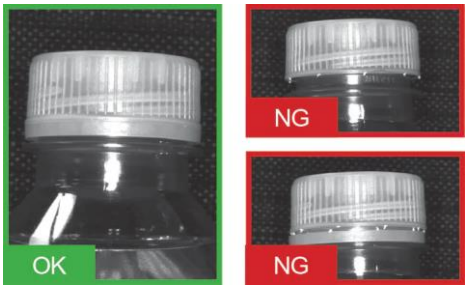
✓

X

4

TinyML - Image Classification Examples

Mask Detection



Deep Learning at the Edge Simplifies Package Inspection

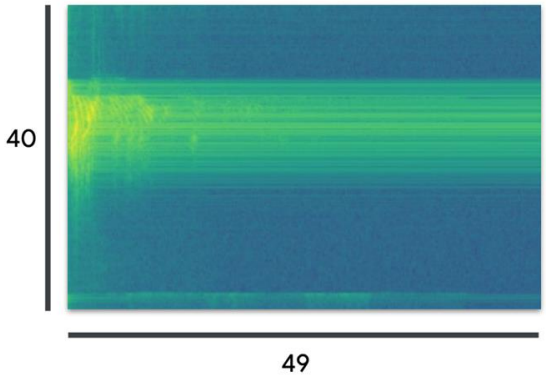
5

Image Classification Challenges

$$224 \times 224 \times 3 \times 4 = 602{,}112 \text{ Bytes}$$

Pixels — RGB (# channels) — Bytes/Pixel

6



$49 \times 40 \times 1 \times 4 = 7{,}840$ Bytes

Pixels | RGB (# channels) | Bytes/Pixel

$224 \times 224 \times 3 \times 4 = 602{,}112$ Bytes

Pixels | RGB (# channels) | Bytes/Pixel

7

Image Classification Challenges

Always-on?

→ Much more data (than KWS)
- Higher **latency**
- Higher **power consumption** (drains battery)
  → Lower **user satisfaction**

8

Memory (CNN Models)

Model	Size	Top-1 Accuracy
Xception	**88 MB**	0.790
VGG16	**528 MB**	0.713
ResNet50	**98 MB**	0.749
Inception v3	**92 MB**	0.779
MobileNet v1	**16 MB**	0.713
DenseNet 201	**80 MB**	0.773
NASNetMobile	**23 MB**	0.825

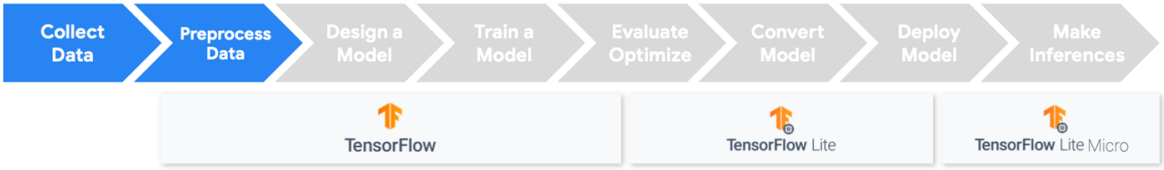
Our board
has **256 KB** of RAM (memory)

9

Image Classification
Data Collection and Processing

10

10



11

Visual Wake Words **Dataset**

Data collection is **DIFFICULT**

- This dataset and collection process is ***limited*** and has bias
- Small number of relevant images
- Large quantity of irrelevant images

Example: Visual Wake Words Dataset

Visual Wake Words Dataset

Aakanksha Chowdhery, Pete Warden, Jonathon Shlens,
Andrew Howard, Rocky Rhodes
Google Research
{chowdhery, petewarden, shlens, howarda, rocky}@google.com

https://arxiv.org/pdf/1906.05721.pdf

12

Example: Visual Wake Words Dataset

Label: "person"

Label: "person"

Label: "not-person"

(Labeled from COCO dataset)

13

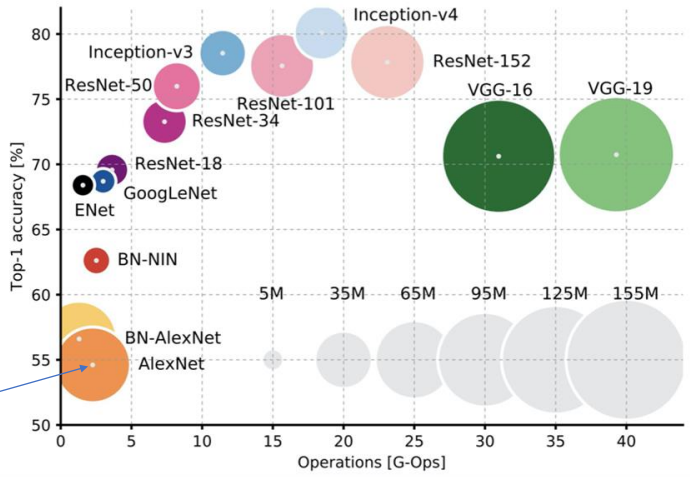
Image Classification
Model

14

14

15

Model **Evolution**

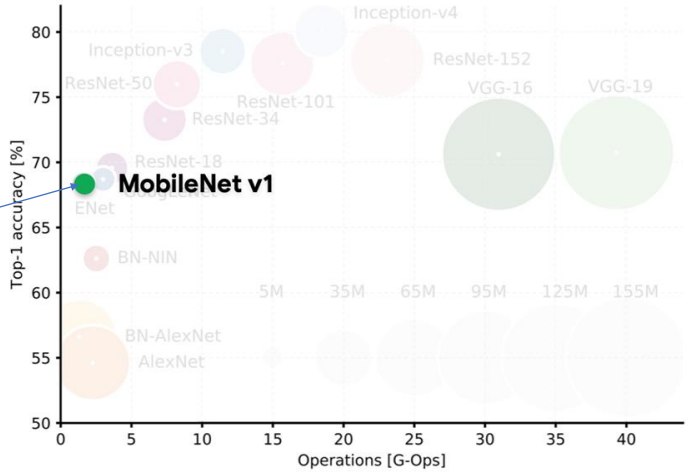


(2012)

16

Model **Evolution**

(2017)



17

MobileNet v1

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

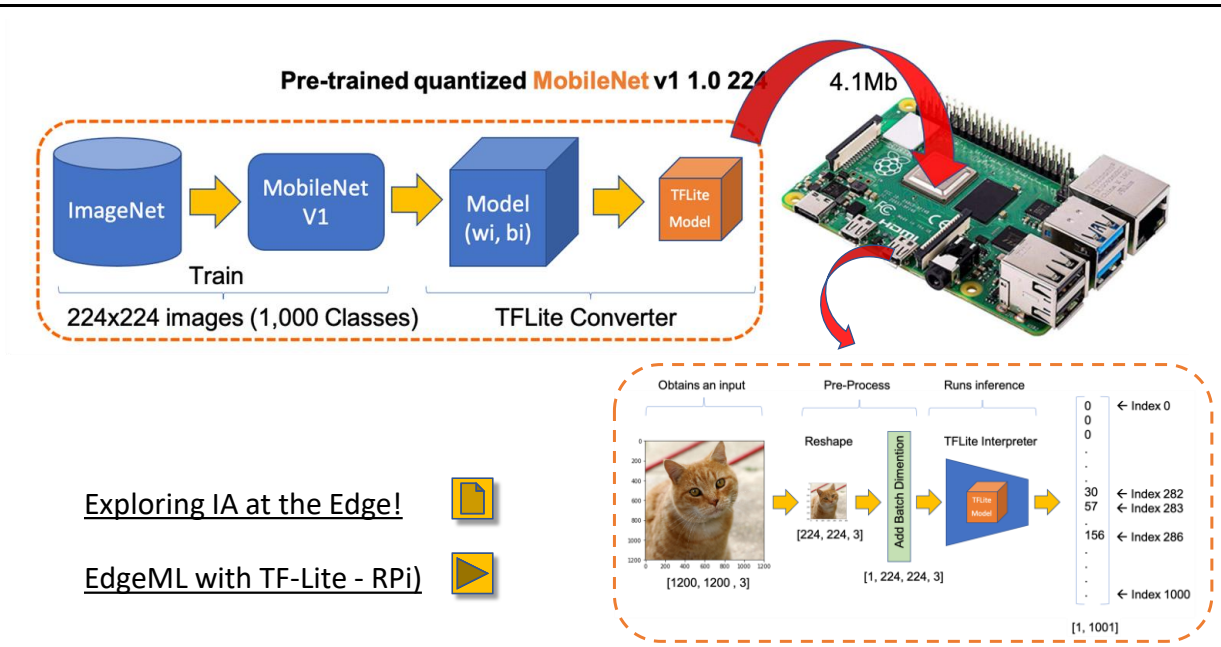
Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko
Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam

Google Inc.

{howarda,menglong,bochen,dkalenichenko,weijunw,weyand,anm,hadam}@google.com

https://arxiv.org/pdf/1704.04861.pdf

18



Exploring IA at the Edge!

EdgeML with TF-Lite - RPi)

19

MobileNet v1

Model	Size	Top-1 Accuracy
MobileNet v1	***16 MB*** *	0.713

* Not Quantized

Fine for mobile phones or Rpi with GB of RAM, but not for microcontroller

Our Arduino Nano only has 256KB of memory RAM

20

Further **Optimizations**

Multiply-Accumulates

α	Image Size	MACs (millions)	Params (millions)	Top-1 Accuracy
1	224	569	**4.24**	**70.7**
1	128	186	4.14	64.1
0.75	224	317	**2.59**	**68.4**
0.75	128	104	2.59	61.8
0.5	224	150	**1.34**	**64.0**
0.5	128	49	1.34	56.2
0.25	224	41	**0.47**	**50.6**
0.25	128	14	0.47	41.2

21

Model

MobileNetV1 96x96 0.25

A pre-trained multi-layer convolutional network designed to efficiently classify images. Uses around 105.9K RAM and 301.6K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

Image Size

MobileNetV1 96x96 0.2

Uses around 83.1K RAM and 218.3K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

Alpha

MobileNetV1 96x96 0.1

Uses around 53.2K RAM and 101K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

ALPHA: Controls the width of the network. This is known as the width multiplier in the MobileNet paper. - If alpha < 1.0, proportionally decreases the number of filters in each layer.

22

Image Classification
Training a Model

23

23

24

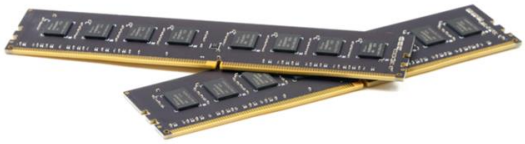
Training Pipeline: **Need Lots of Data**

1000 Classes

1000 Images / Class

25

Training Pipeline: **Need Compute Resources**



Memory

Compute

GPU and Accelerators

26

Training Pipeline: **Need Compute Resources**

Computationally Intensive
Repeated Many Times (Epochs)

Memory

Compute

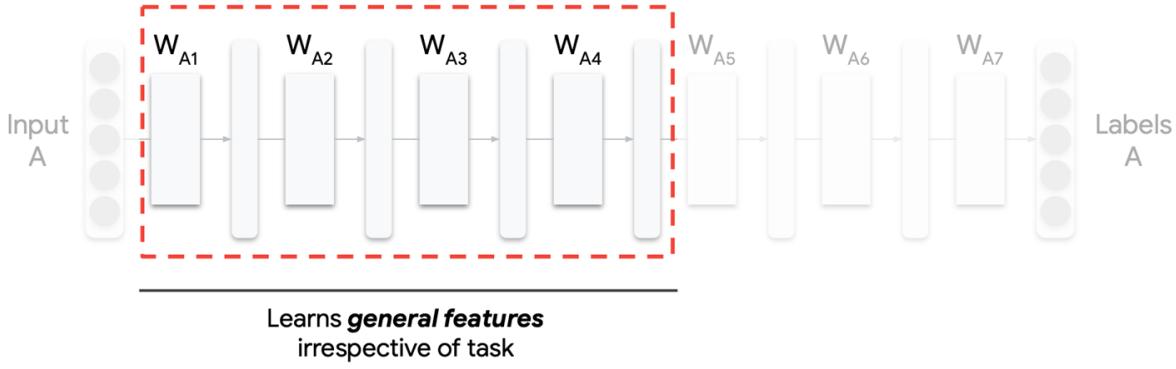
GPU and
Accelerators

27

End Result of Training

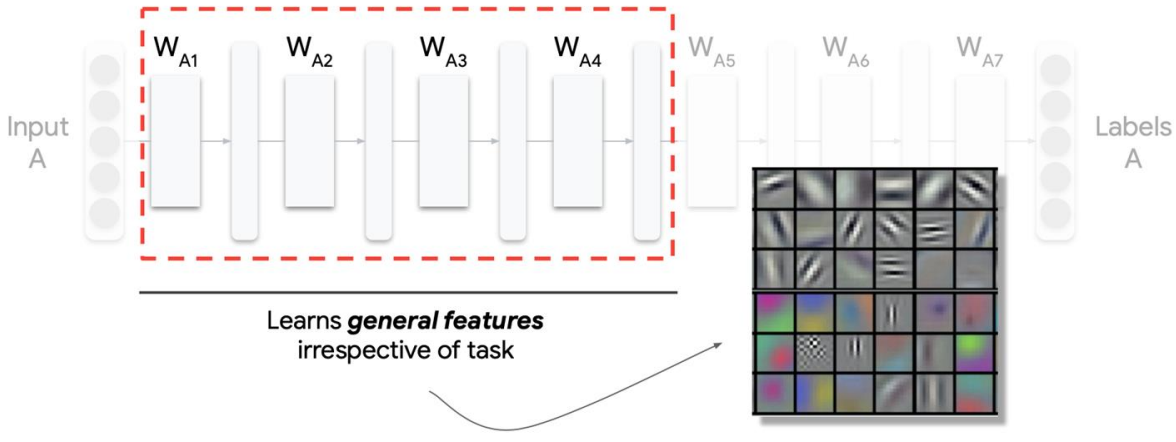
28

End Result of Training



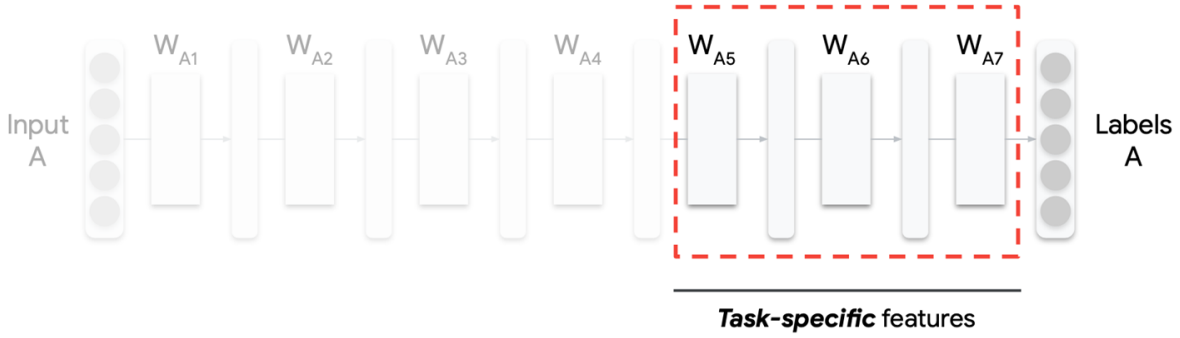
29

End Result of Training



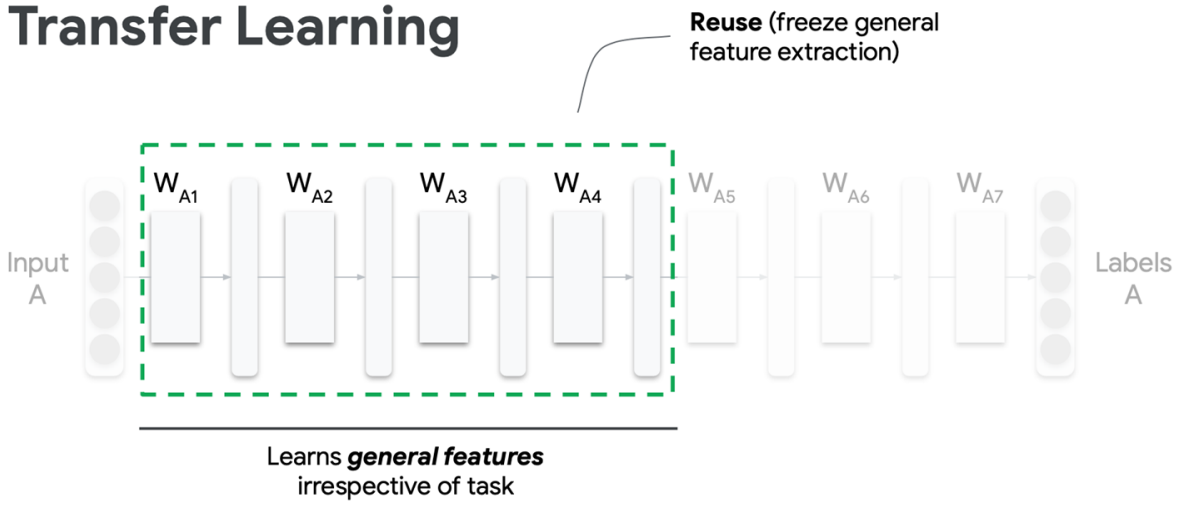
30

End Result of Training



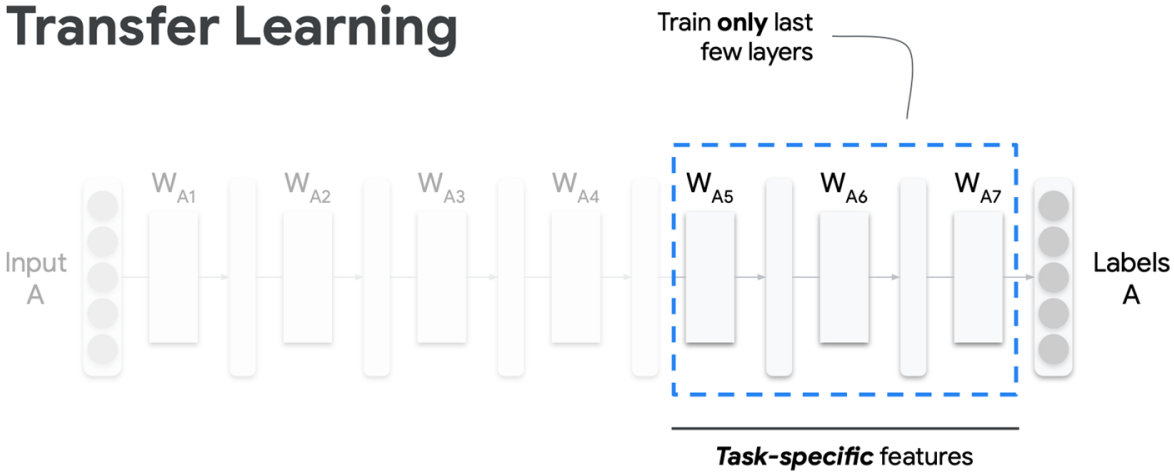
31

Transfer Learning



32

Transfer Learning



33

Using Transfer Learning for:

1) Cat Dog Detection
2) Mask Detection

Code Time!

1) Cat_Dog_Detection_using_Transfer_Learning_TFL_Micro.ipynb
2) Mask_Detection_using_Transfer_Learning.ipynb

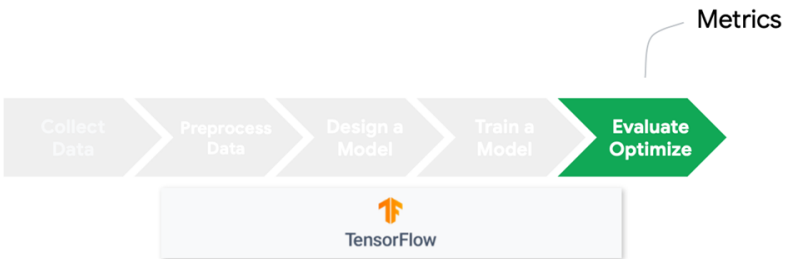
34

34

Image Classification
Metrics

35

35



36

Common Metrics

Accuracy

Quantitative

Efficiency

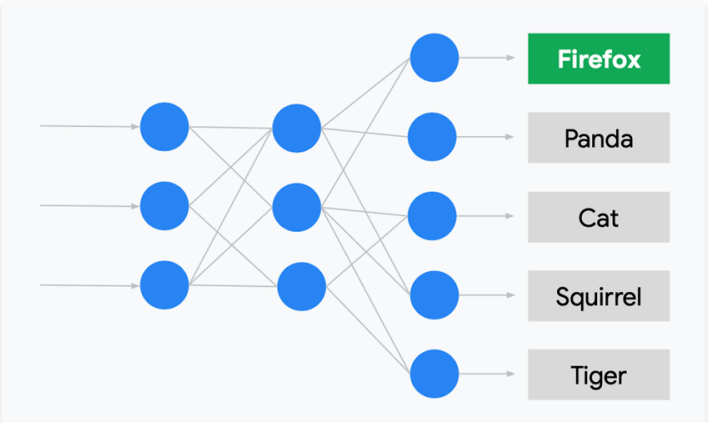
Quantitative

User Experience

Qualitative

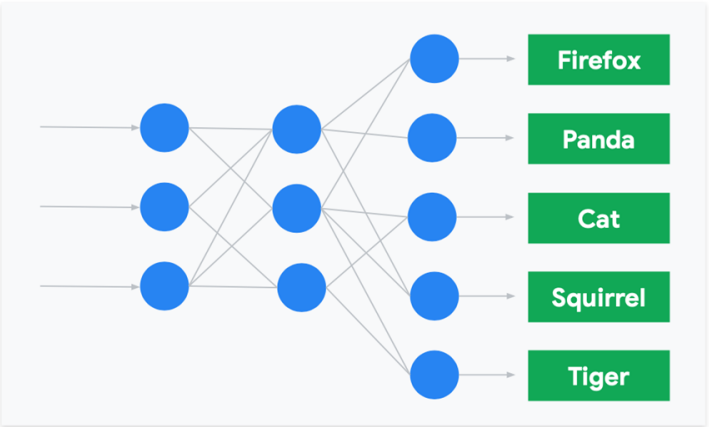
37

Top-1 Accuracy



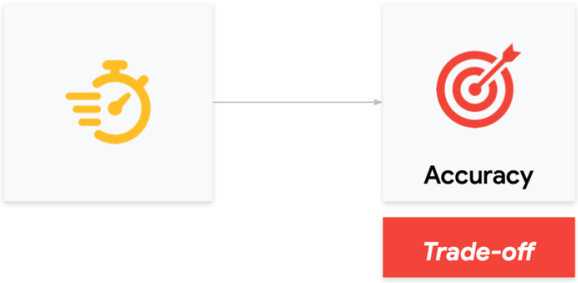
38

Top-5 Accuracy



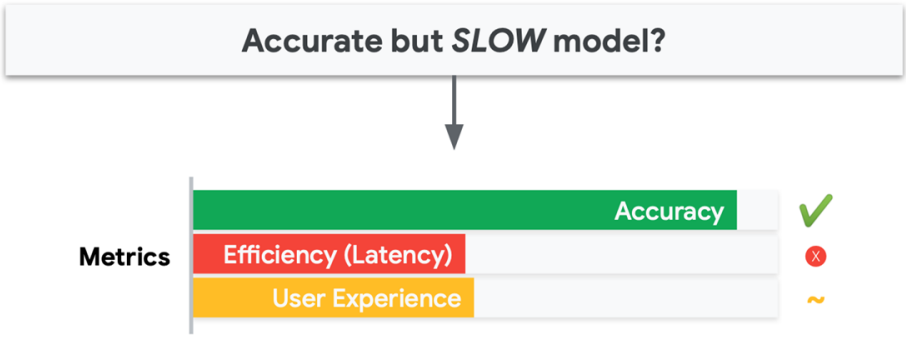
39

Latency



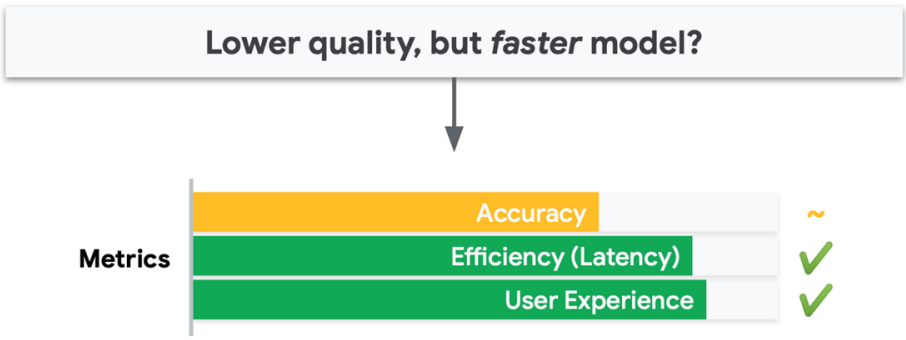
40

Latency



41

Latency



42

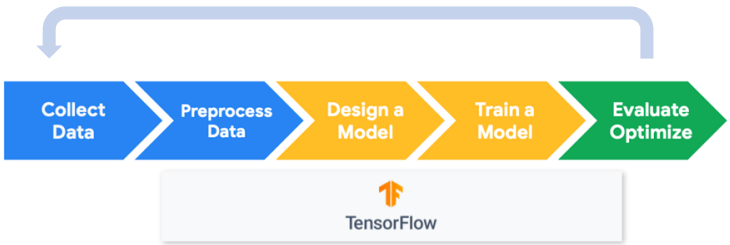
Fairness

User in ***majority*** group of training data?

***Diverse*, representative data** is important because it enables fair use (equal performance) across populations

43

Achieving Ideal Metrics: Revisit pipeline



44

Credits

- A previous edition of this course was developed in collaboration with Dr. Susan C. Schneider of Marquette University.
- We are very grateful and thank all the following professors, researchers, and practitioners for jump-starting courses on TinyML and for sharing their teaching materials:

- Prof. Marcelo Rovai - TinyML - Machine Learning for Embedding Devices, UNIFEI
  - https://github.com/Mjrovai/UNIFEI-IESTI01-TinyML-2022.1
- Prof. Vijay Janapa Reddi - CS249r: Tiny Machine Learning, Applied Machine Learning on Embedded IoT Devices, Harvard
  - https://sites.google.com/g.harvard.edu/tinyml/home
- Prof. Rahul Mangharam – ESE3600: Tiny Machine Learning, Univ. of Pennsylvania
  - https://tinyml.seas.upenn.edu/#
- Prof. Brian Plancher - Harvard CS249r: Tiny Machine Learning (TinyML), Barnard College, Columbia University
  - https://a2r-lab.org/courses/cs249r_tinyml/

References

- Additional references from where information and other teaching materials were gathered include:

- Applications & Deploy textbook: “TinyML” by Pete Warden, Daniel Situnayake
  - https://www.oreilly.com/library/view/tinyml/9781492052036/
- Deploy textbook “TinyML Cookbook” by Gian Marco Iodice
  - https://github.com/PacktPublishing/TinyML-Cookbook
- Jason Brownlee
  - https://machinelearningmastery.com/
- TinyMLedu
  - https://tinyml.seas.harvard.edu/
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
  - https://www.edx.org/professional-certificate/harvardx-tiny-machine-learning
- Introduction to Embedded Machine Learning - Coursera/Edge Impulse
  - https://www.coursera.org/learn/introduction-to-embedded-machine-learning
- Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse
  - https://www.coursera.org/learn/computer-vision-with-embedded-machine-learning