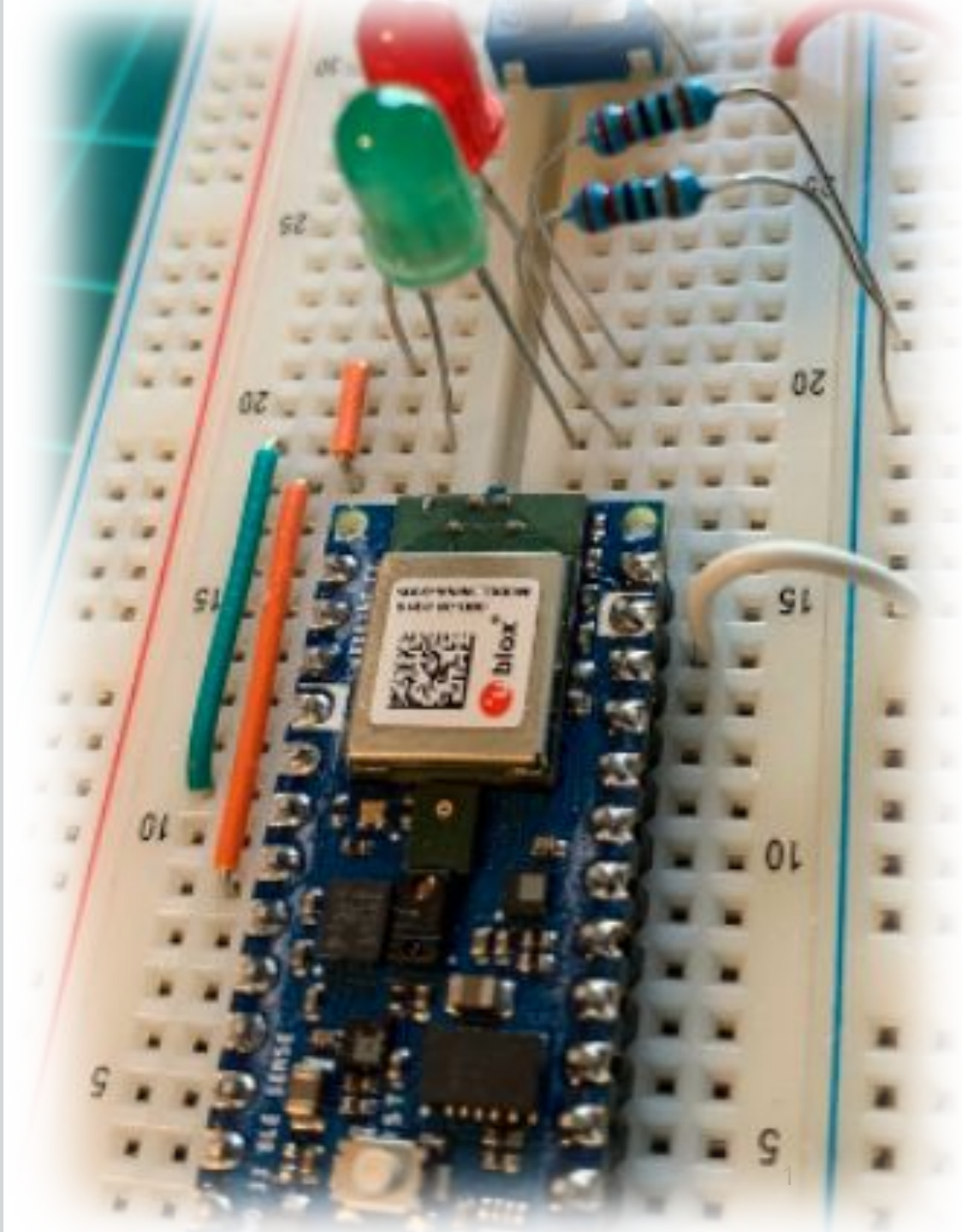
IESTI01 – TinyML

Embedded Machine Learning

25. Image Classification Introduction



Prof. Marcelo Rovai
UNIFEI

Image Classification,
Introduction & Challenges

Computer Vision Main Types

Image Classification

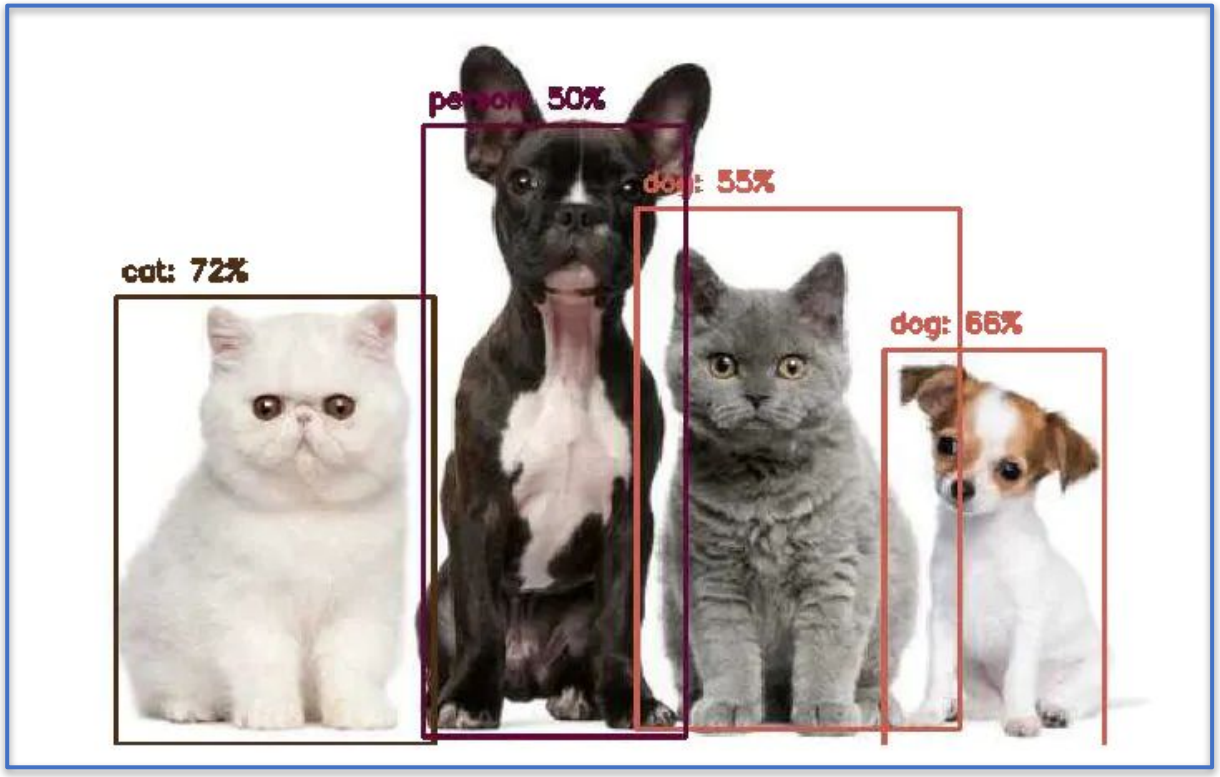
(**Multi-Class** Classification)

Cat: 70%

Dog: 80%

Object Detection

Multi-Label Classification + Object **Localization**



Computer Vision Main Types

Image Classification

(**Multi-Class** Classification)

Cat: 70%

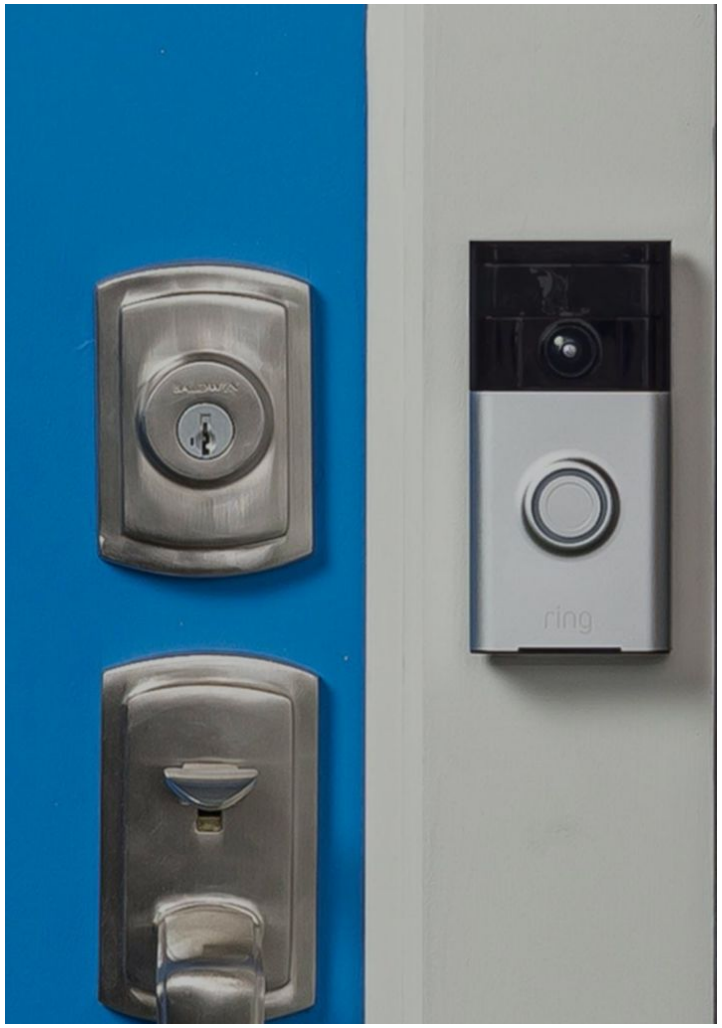
Dog: 80%

Object Detection

Multi-Label Classification + Object **Localization**

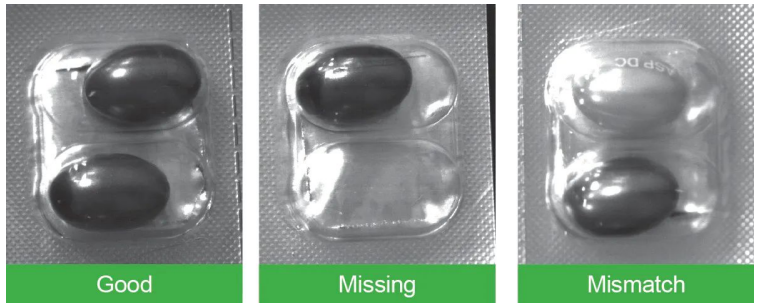
Person Detection (Visual Wake Words)

X



TinyML - Image Classification examples

Mask Detection



Deep Learning at the Edge Simplifies Package Inspection

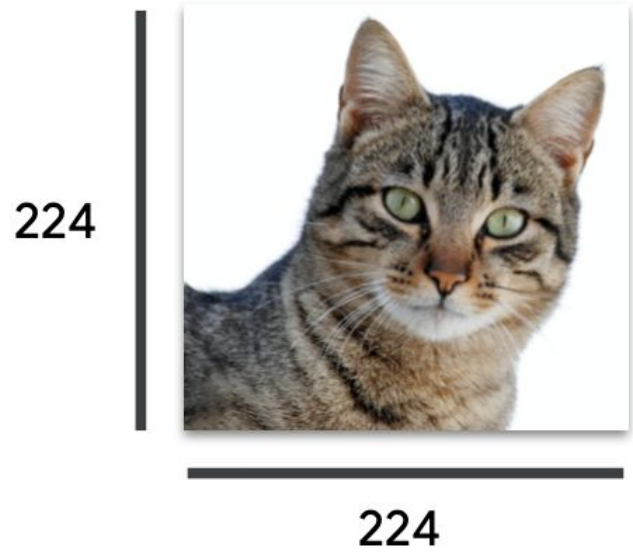
Image Classification Challenges

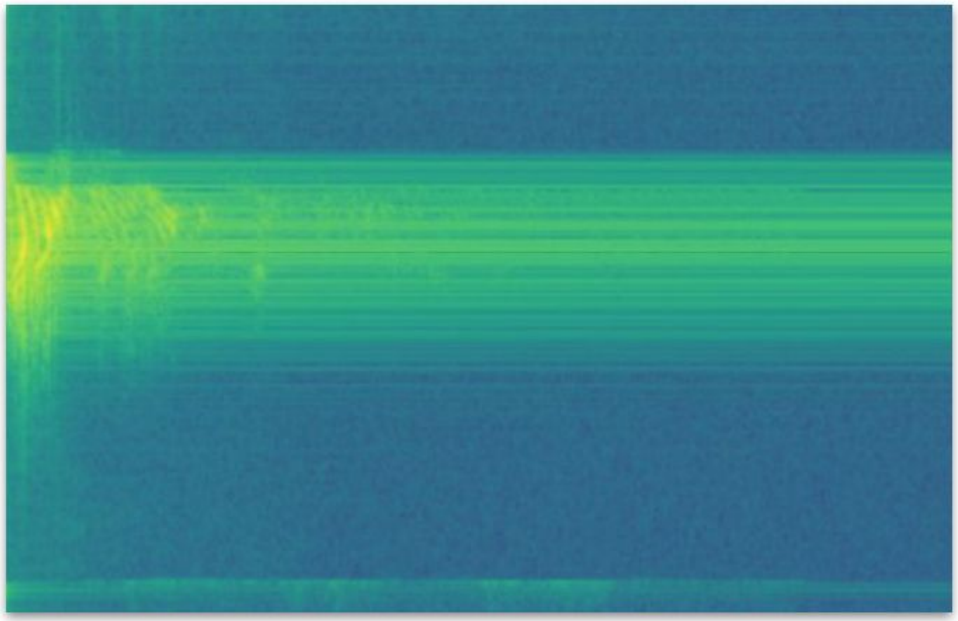
$$224 \times 224 \times 3 \times 4 = 602{,}112 \text{ Bytes}$$

Pixels

RGB (# channels)

Bytes/Pixel





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

Pixels | RGB (# channels) | Bytes/Pixel

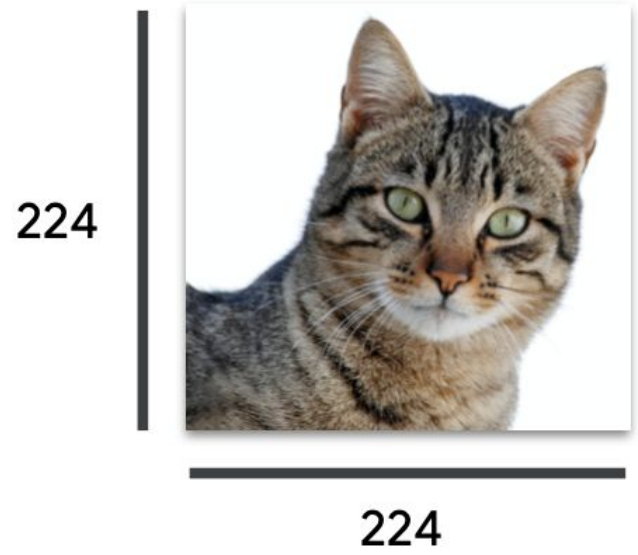
$$224 \times 224 \times 3 \times 4 = 602{,}112 \text{ Bytes}$$

Pixels | RGB (# channels) | Bytes/Pixel

Image Classification Challenges

Always-on?

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

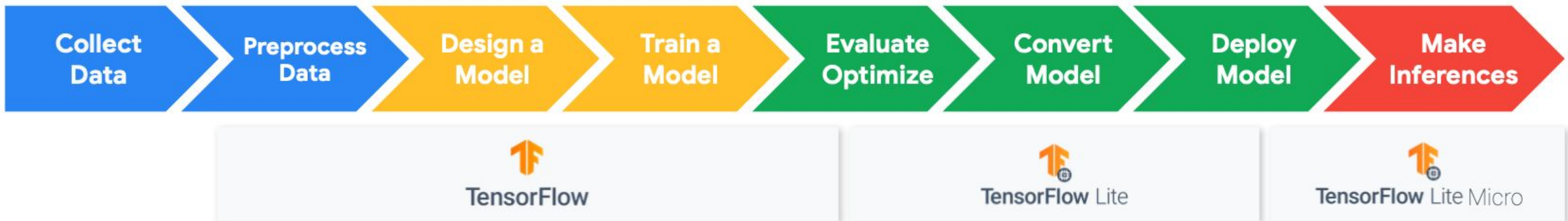


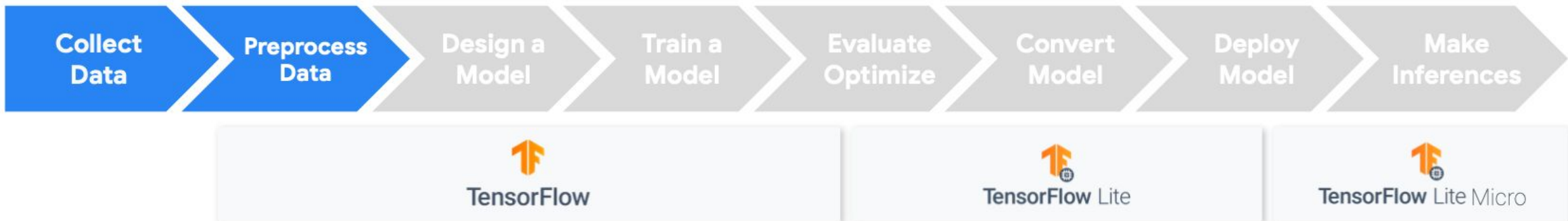
Memory (CNN Models)

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)

Image Classification,
Data Collection and Processing


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro

Example: Visual Wake Words **Dataset**

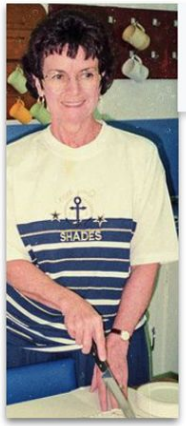
Visual Wake Words Dataset

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

Example: Visual Wake Words Dataset

Label: “person”



Label: “person”

Label: “not-person”

(Labeled from COCO dataset)

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

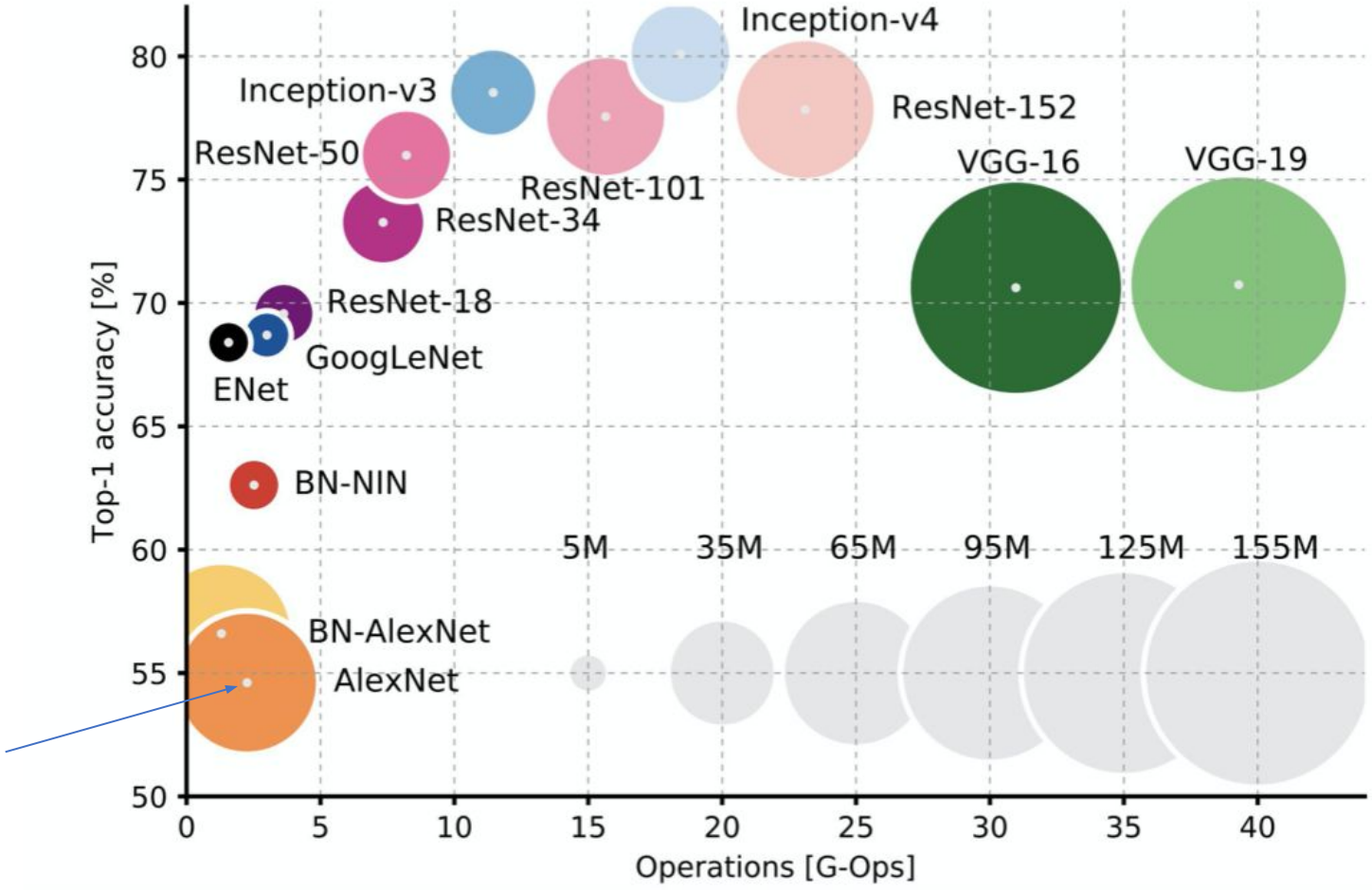
Visual Wake Words Dataset

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

Image Classification, Model


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro

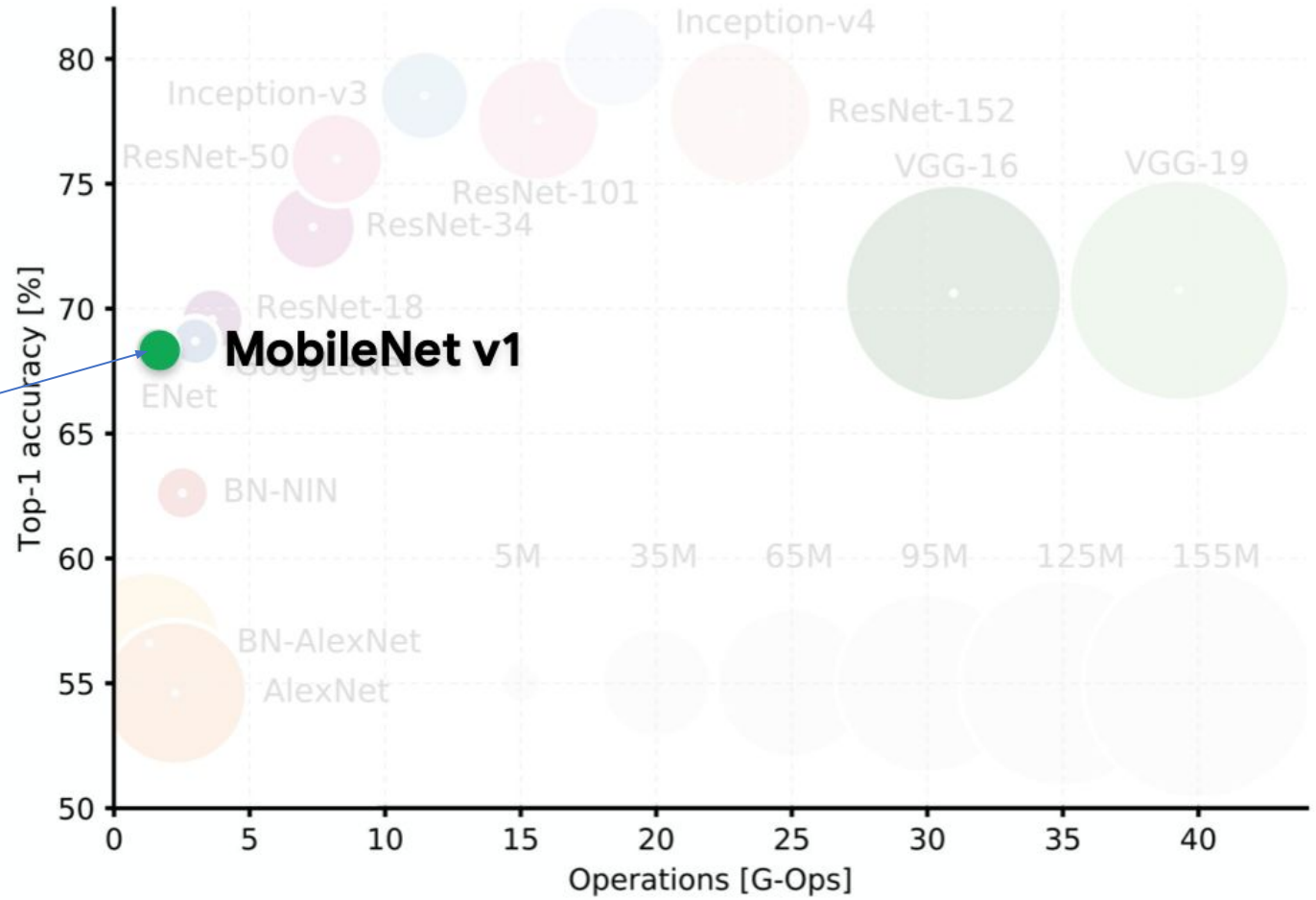
Model **Evolution**



(2012)

Model **Evolution**

(2017)



MobileNet v1

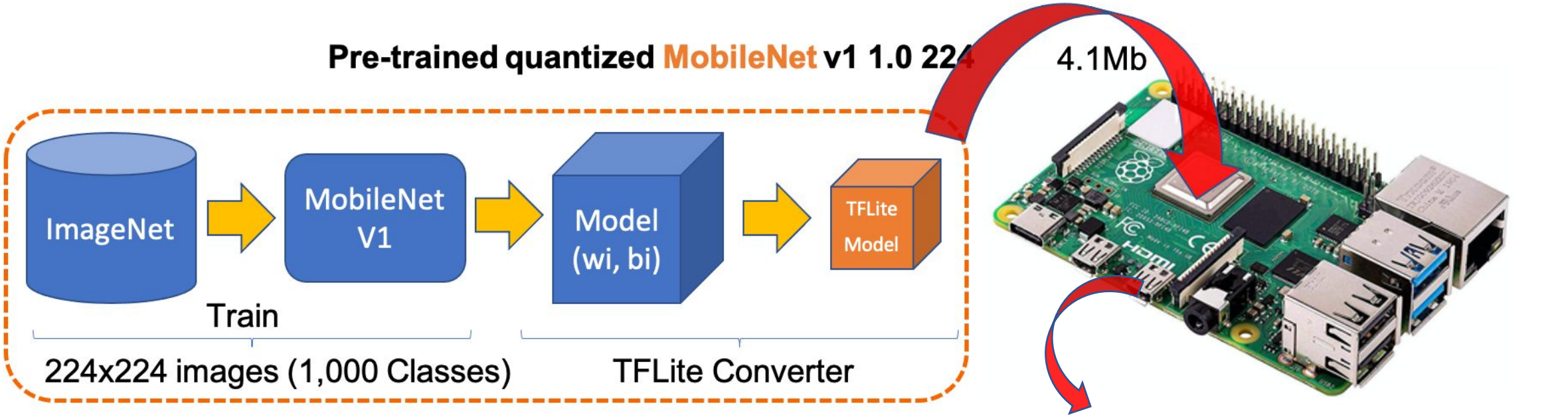
MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

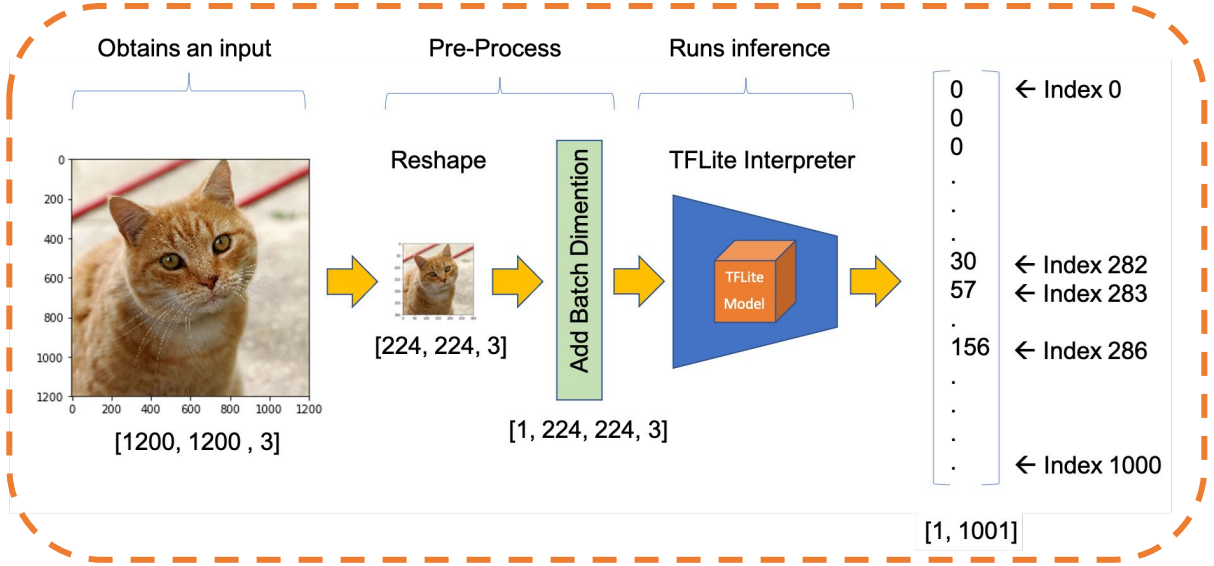
Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko
Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam

Google Inc.

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

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





Exploring IA at the Edge!

EdgeML with TF-Lite - RPi)

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

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

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

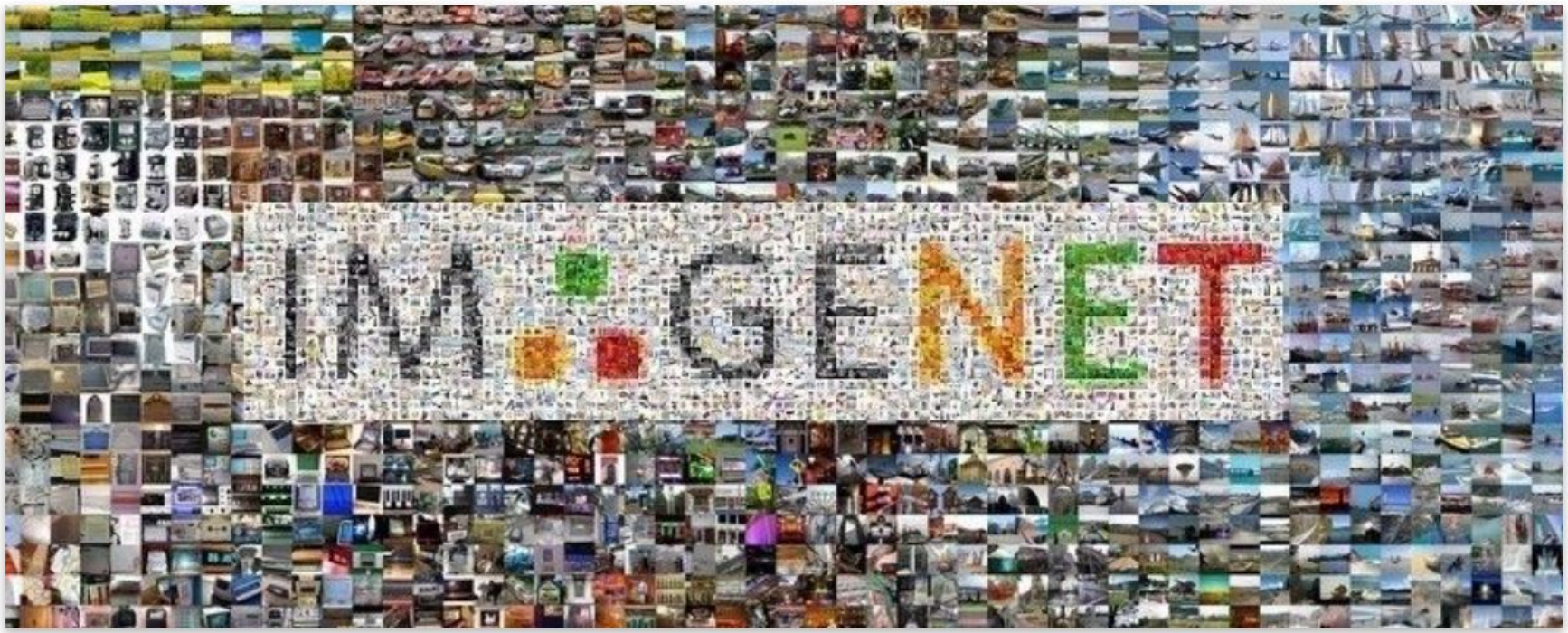
Uses around 53.2K RAM and 101K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

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.

Image Classification,
Training a Model


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro

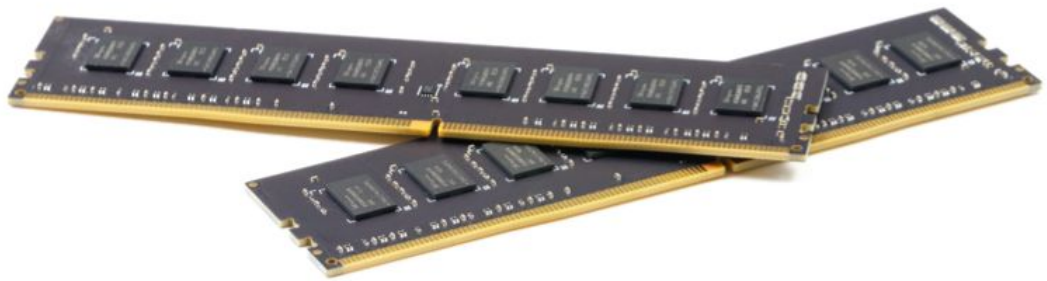
Training Pipeline: **Need Lots of Data**



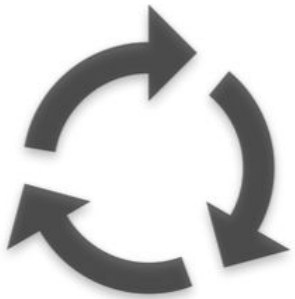
1000 Classes

1000 Images / Class

Training Pipeline: **Need Compute Resources**



Memory

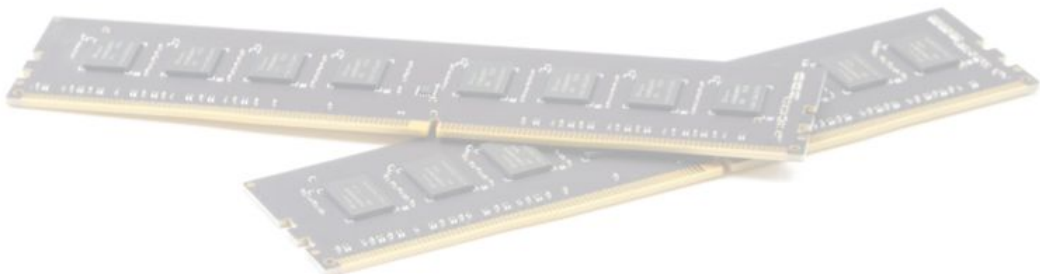


Compute

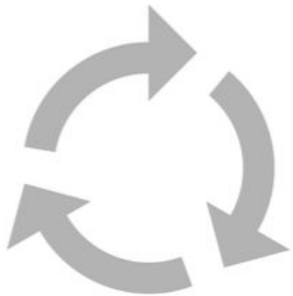
GPU and
Accelerators

Training Pipeline: **Need Compute Resources**

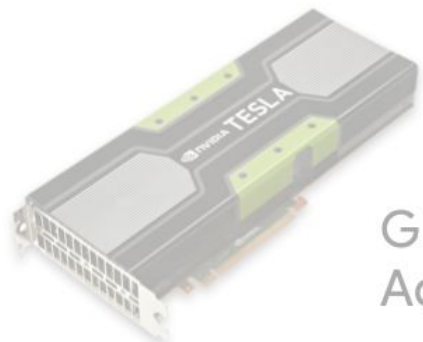
Computationally Intensive
Repeated Many Times (Epochs)



Memory

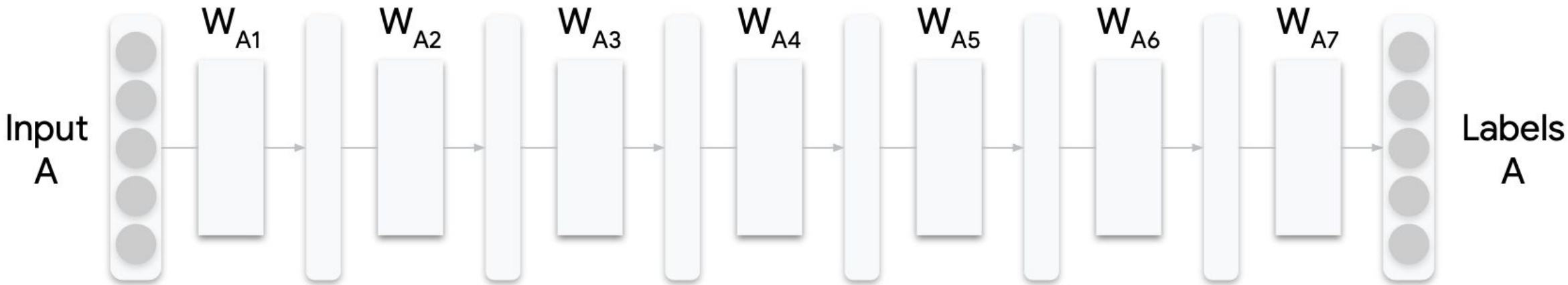


Compute

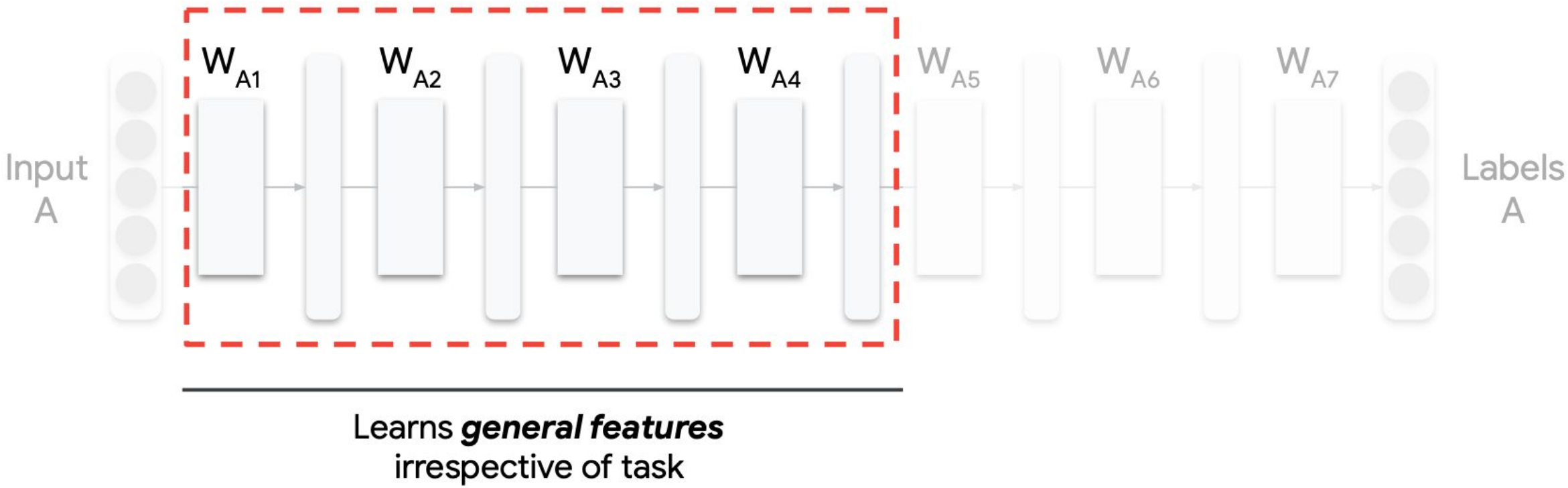


GPU and
Accelerators

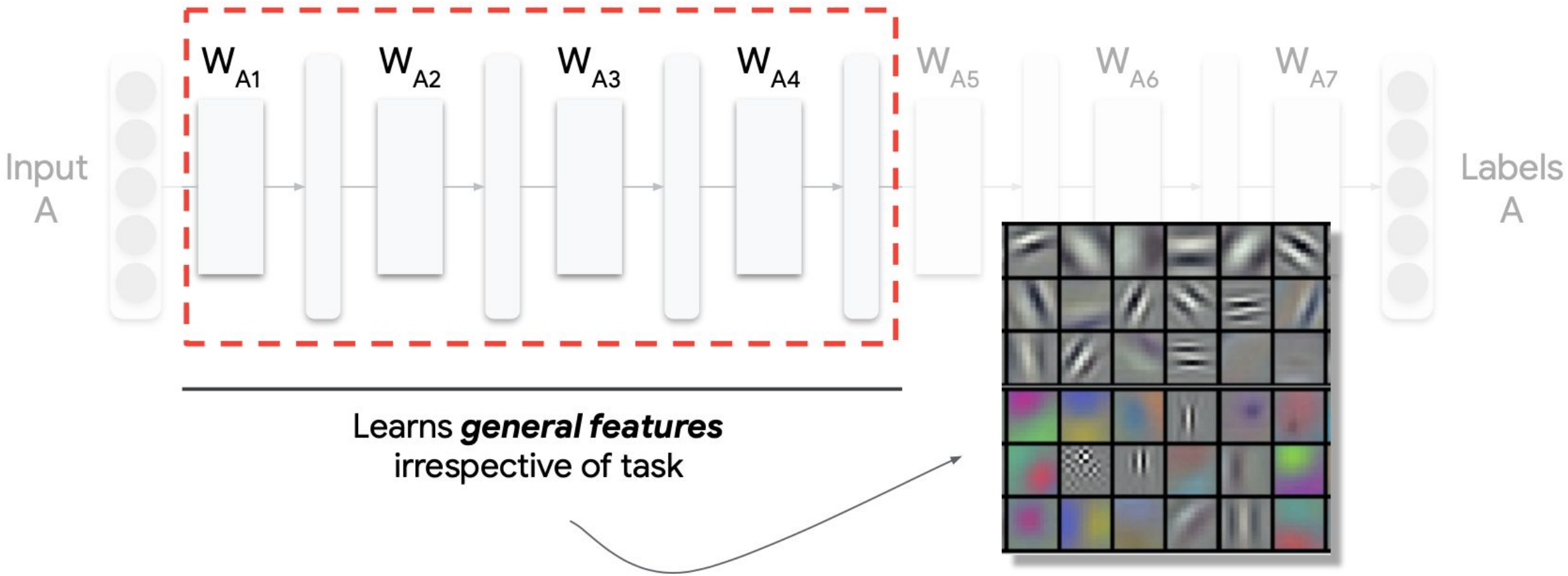
End Result of Training



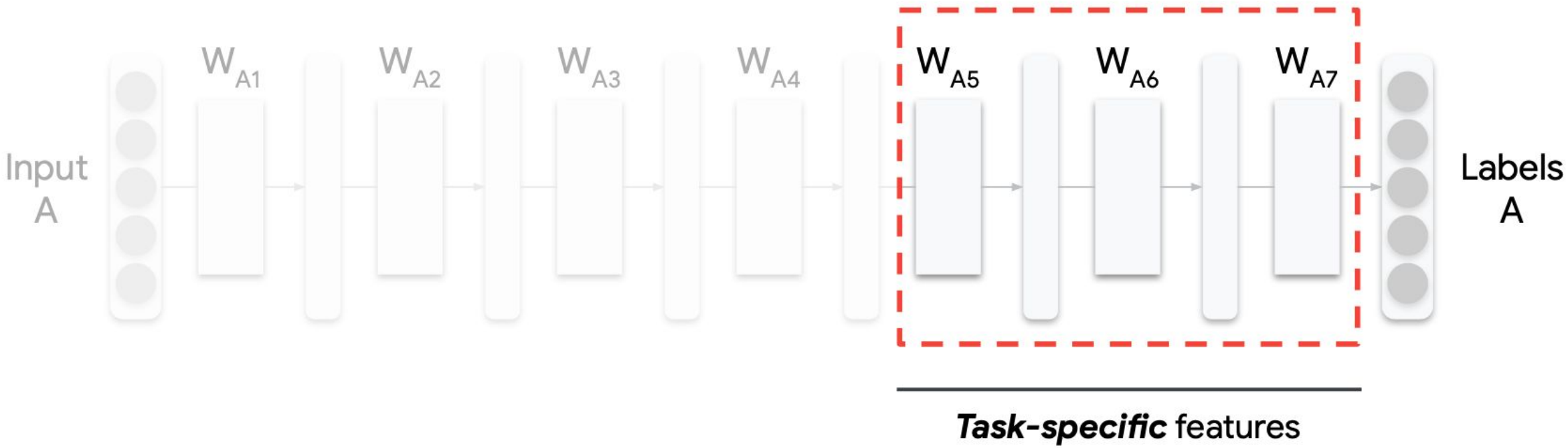
End Result of Training



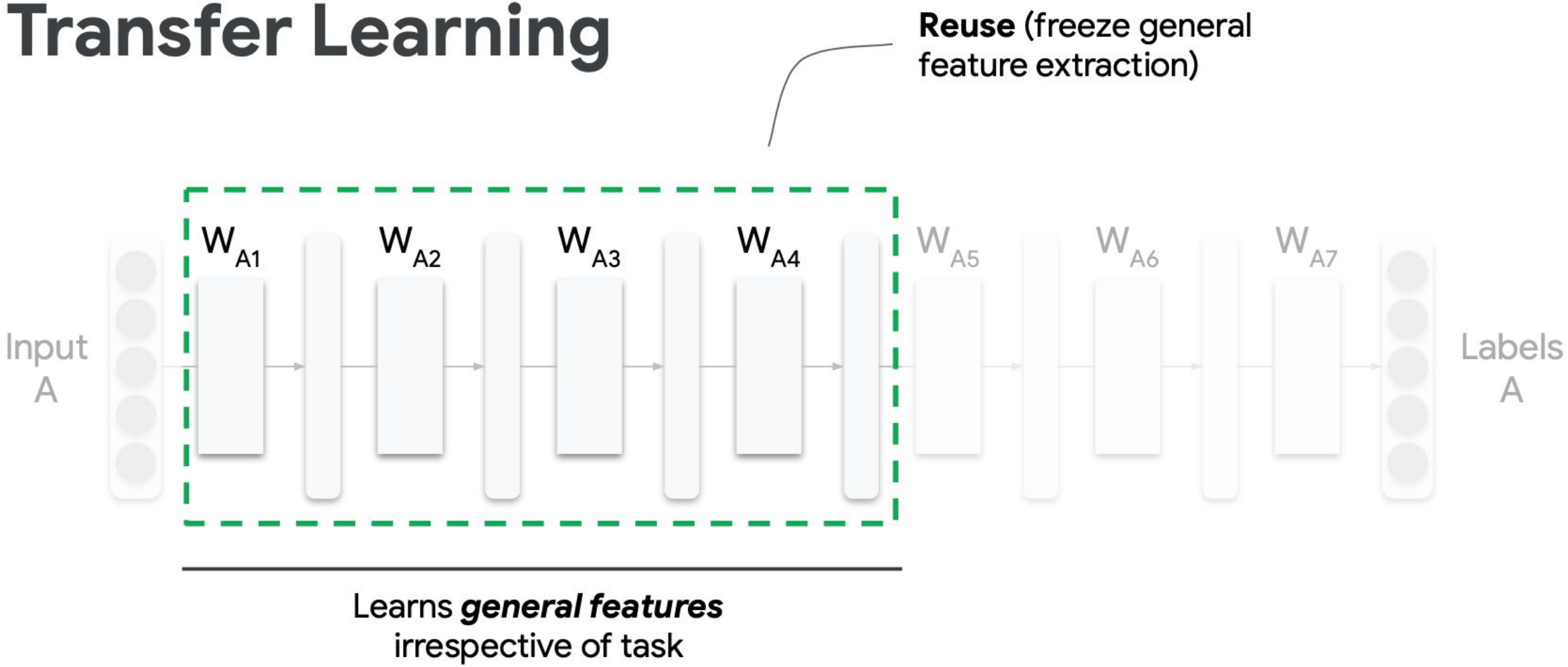
End Result of Training



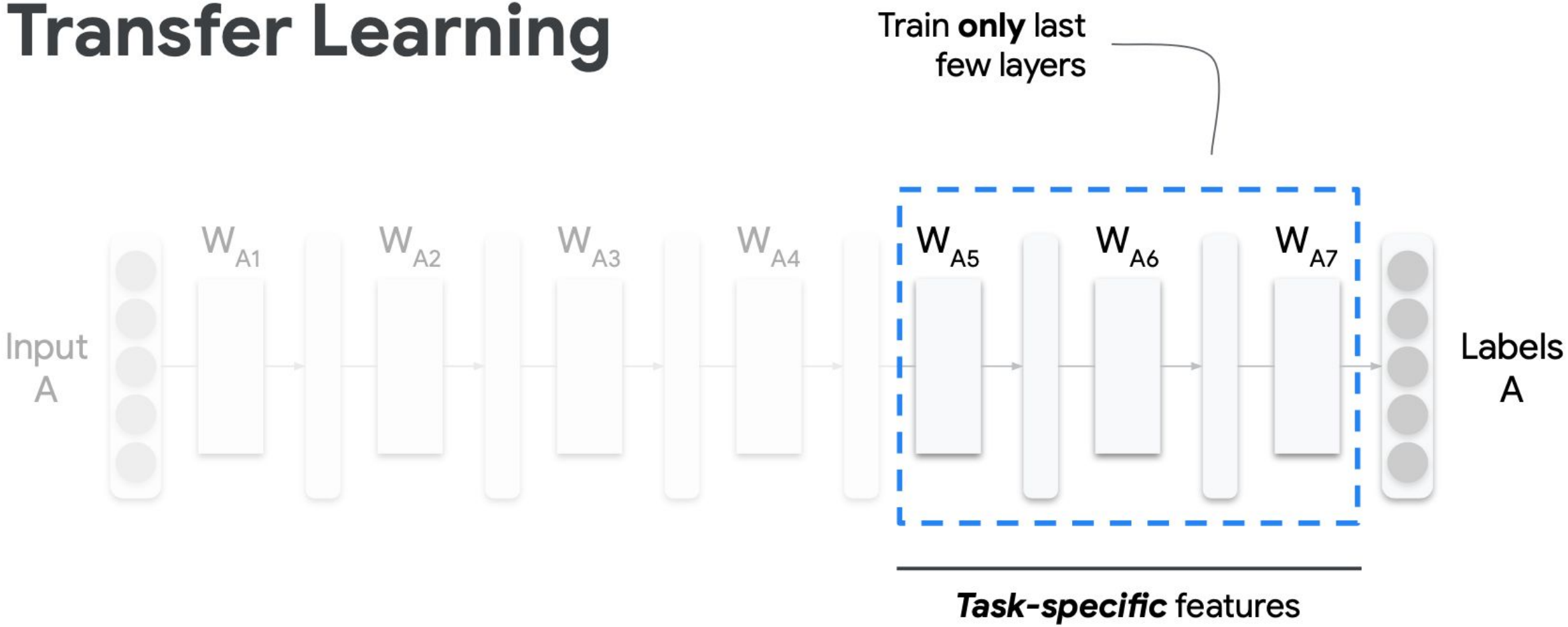
End Result of Training



Transfer Learning



Transfer Learning

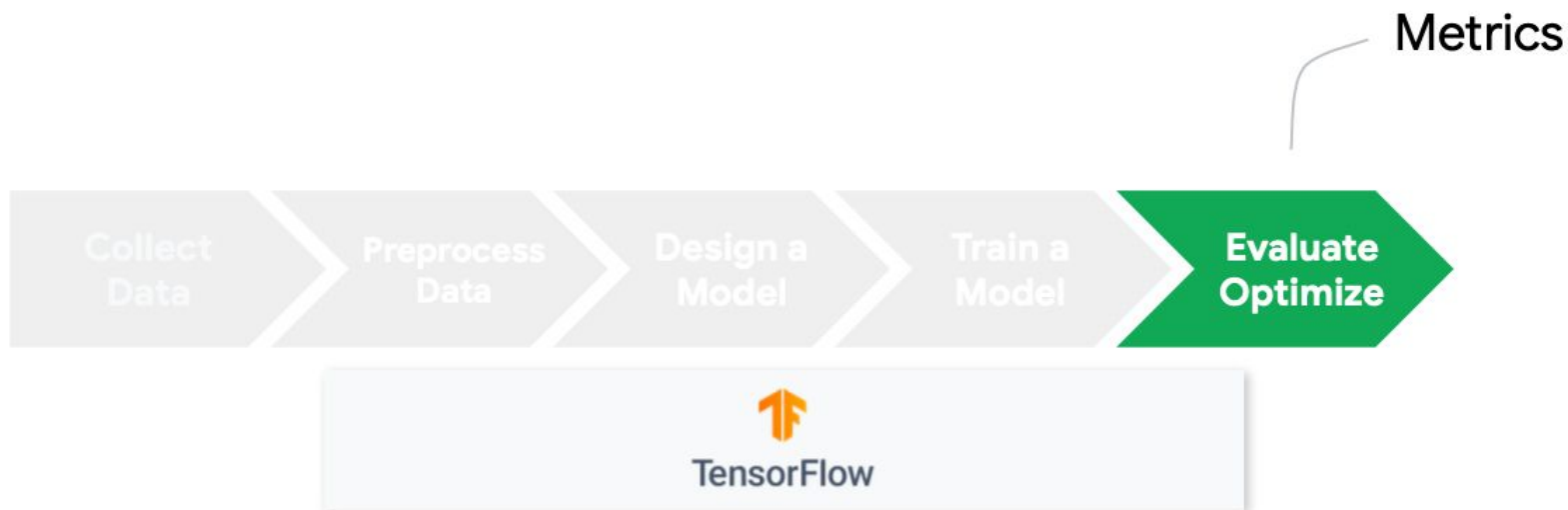


Mask Detection using Transfer Learning

Code Time!

Mask_Detection_using_Transfer_Learning.ipynb

Image Classification,
Metrics


Metrics
Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
TensorFlow

Common Metrics

Accuracy

Quantitative

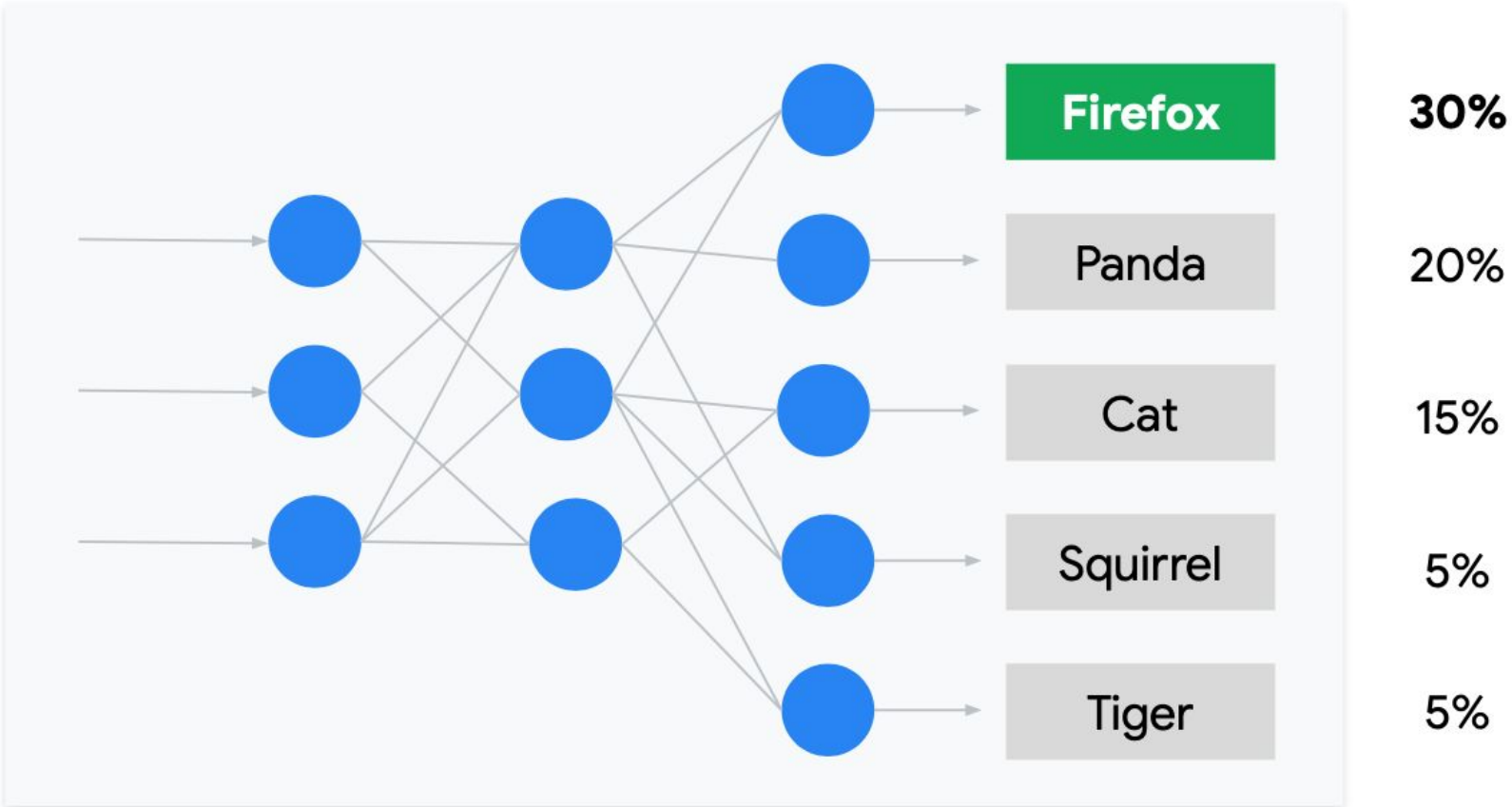
Efficiency

Quantitative

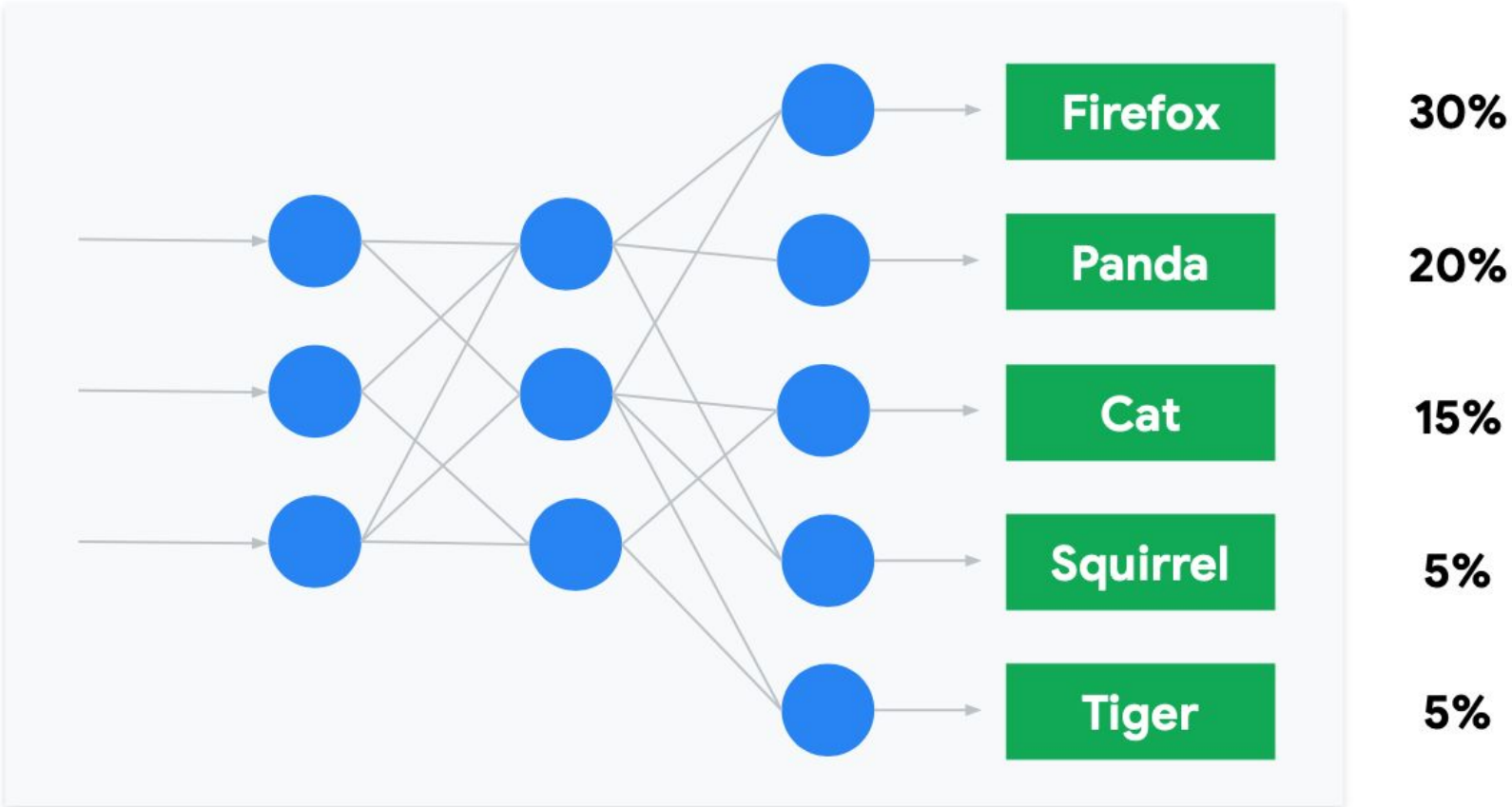
User Experience

Qualitative

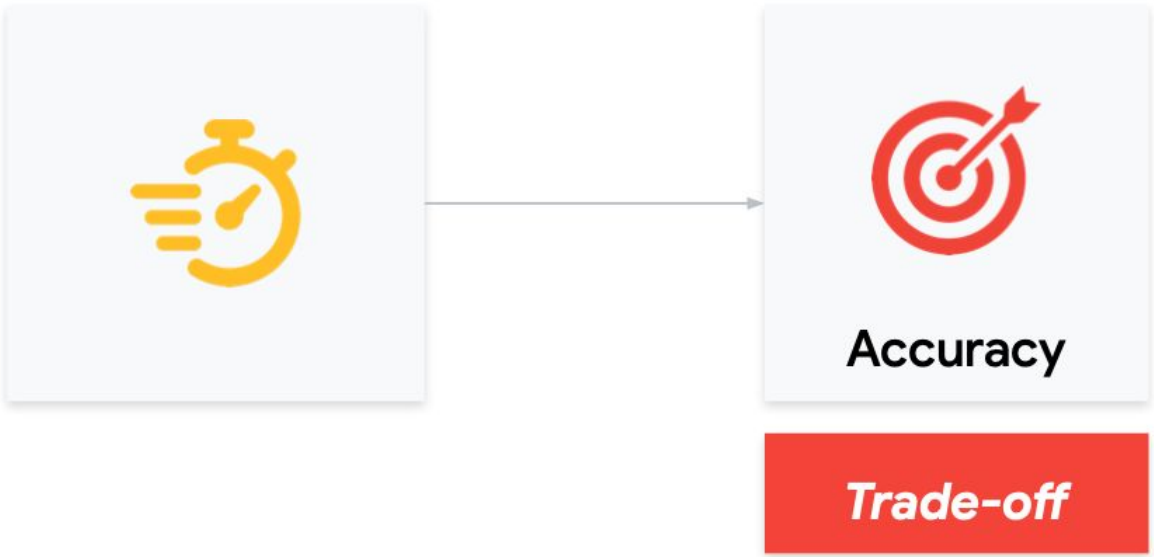
Top-1 Accuracy



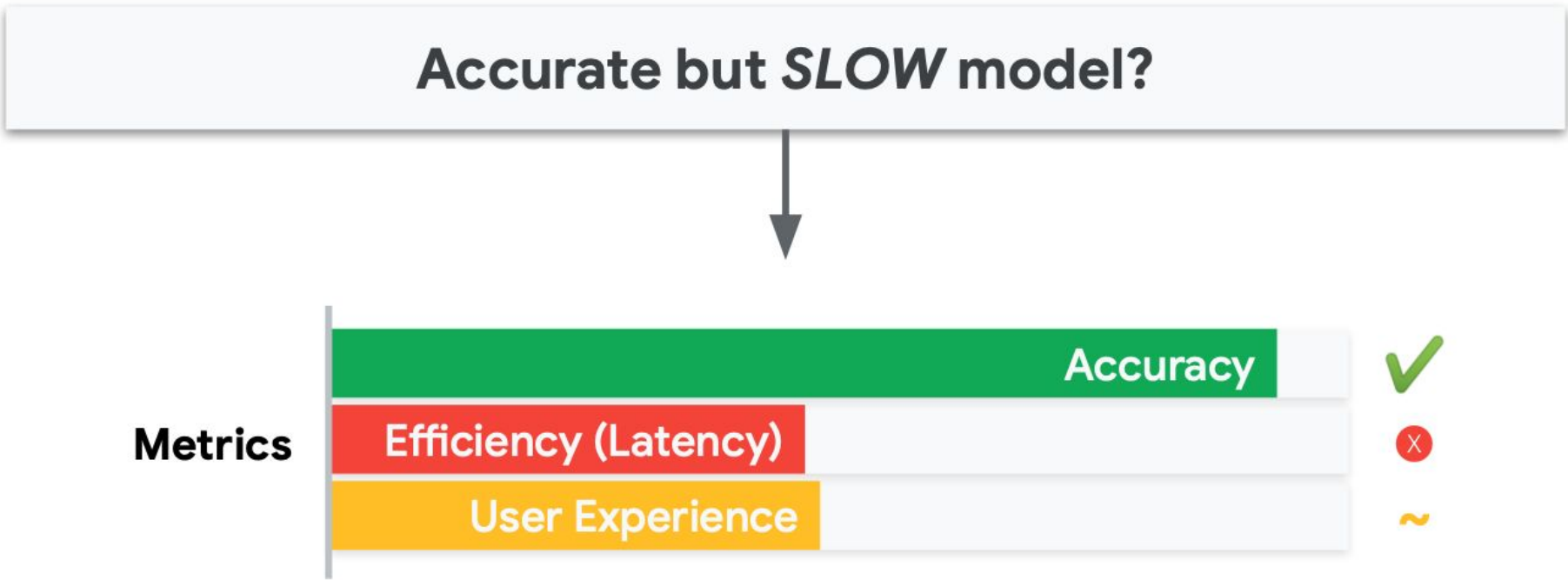
Top-5 Accuracy



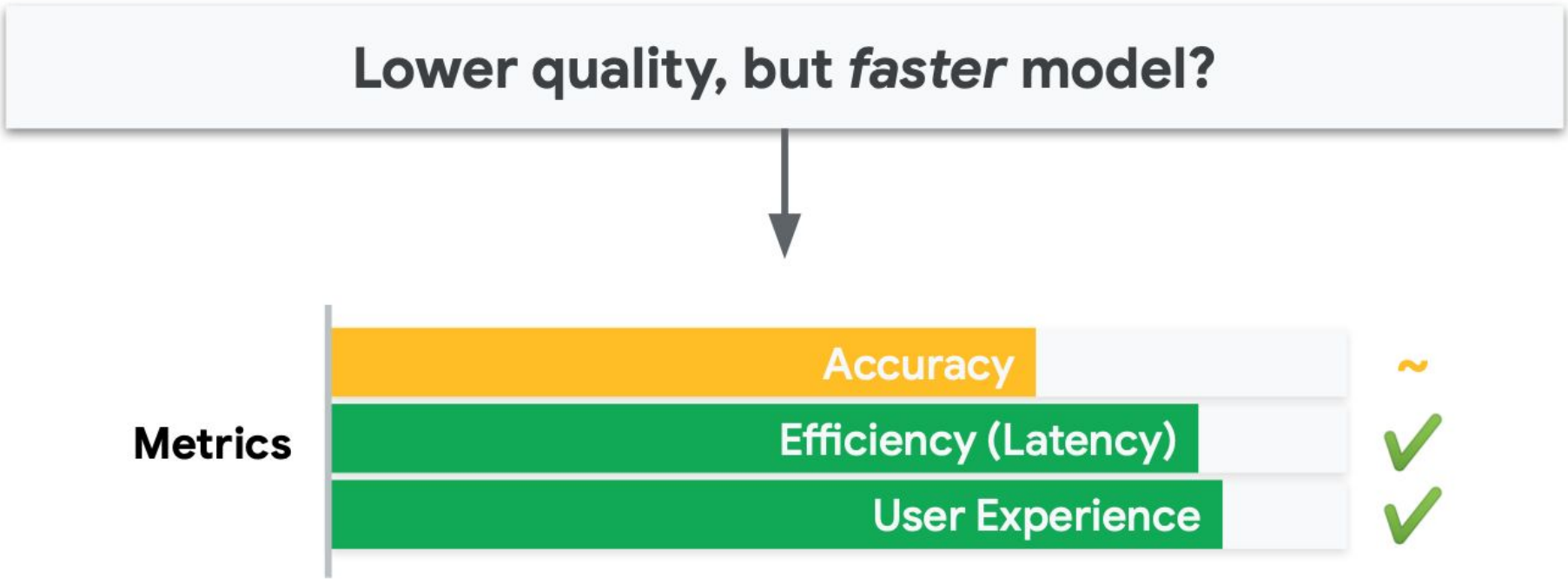
Latency



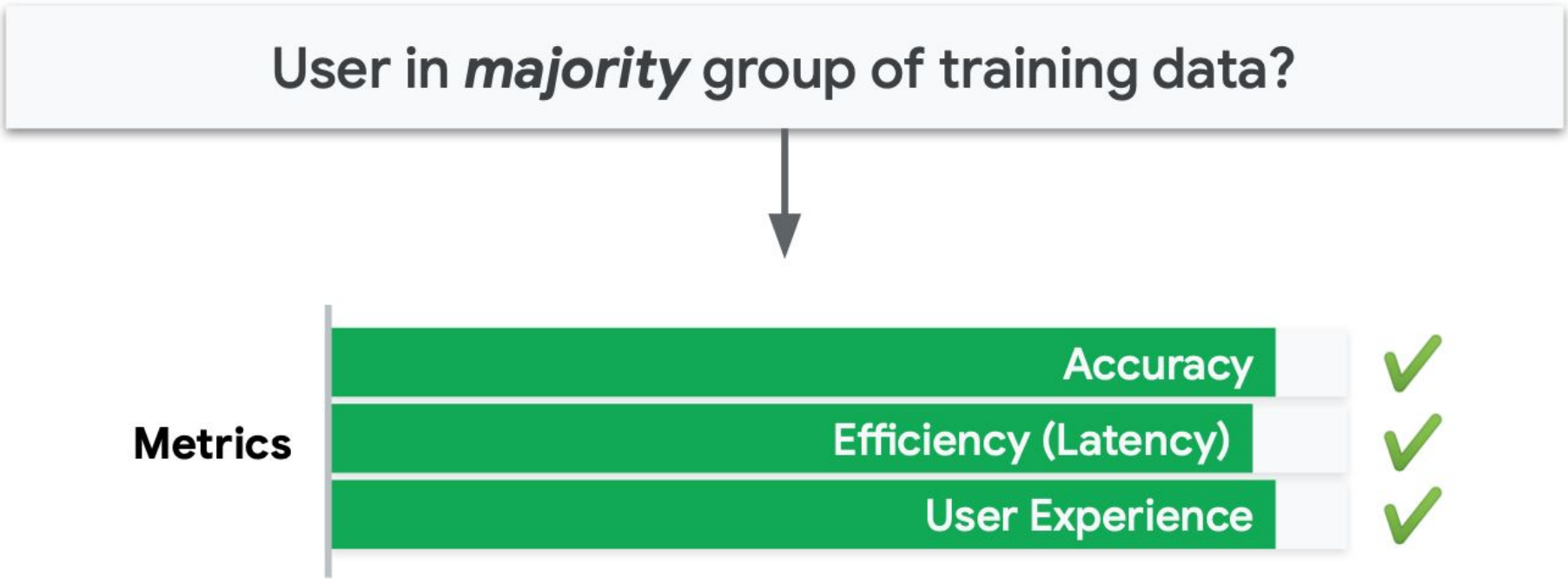
Latency


Accurate but *SLOW* model?
Metrics
Accuracy
Efficiency (Latency)
User Experience

Latency

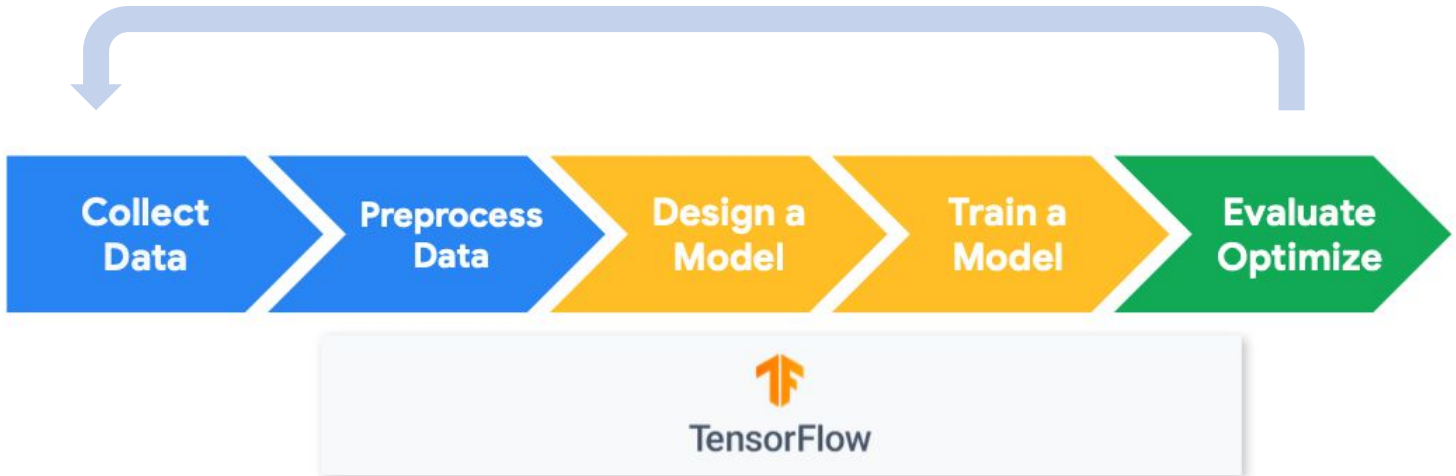

Lower quality, but *faster* model?
Metrics
Accuracy
Efficiency (Latency)
User Experience
~
✓
✓

Fairness



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

Achieving Ideal Metrics: Revisit pipeline



Reading Material

Main references

- Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning - Coursera/Edge Impulse
- Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse
- Fundamentals textbook: "Deep Learning with Python" by François Chollet
- Applications & Deploy textbook: "TinyML" by Pete Warden, Daniel Situnayake
- Deploy textbook "TinyML Cookbook" by Gian Marco Iodice

I want to thank Shawn Hymel and Edge Impulse, Pete Warden and Laurence Moroney from Google, Professor Vijay Janapa Reddi and Brian Plancher from Harvard, and the rest of the TinyMLedu team for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the **TinyML4D**, an initiative to make TinyML education available to everyone globally.

Thanks

UNIFEI