June 2020

# CS293B Course Project

## Deep Learning and Object Classification at the Edge

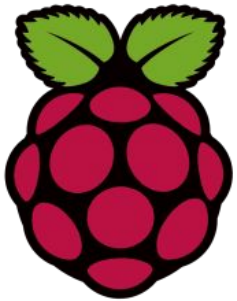
Department of Computer Science

UC SANTA BARBARA

# Project Plan

- Configure a system that uses a robot's camera feed and a trained deep learning model to perform object classification at the Edge.
  - Architecture similar to *Where's The Bear?*

IoT

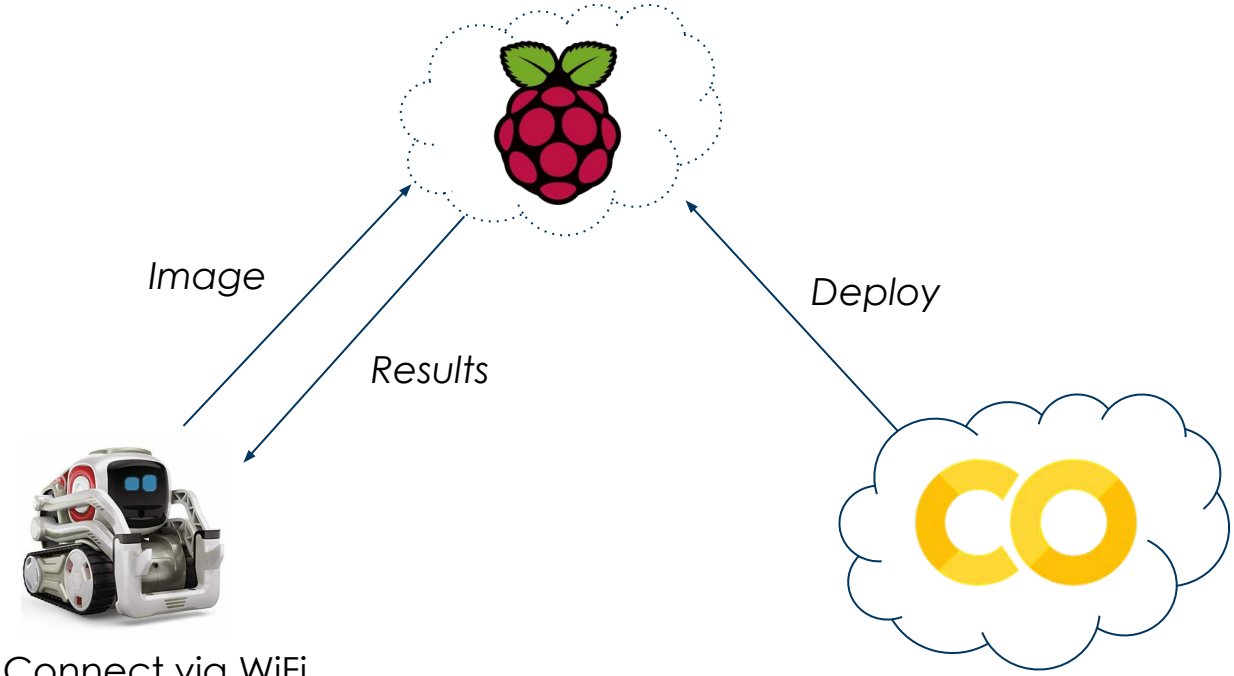


Edge

Cloud

# Architecture

- Model & Labels
- TensorFlow Lite Interpreter




- Connect via WiFi
- SDK on PC

- TensorFlow Lite
- Training Datasets

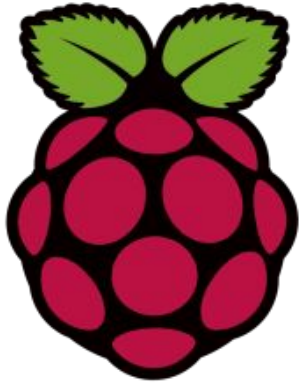
# **Cloud:** Model Training and Data Storage

- **Google Colaboratory**
  - Hosted Jupyter notebook service
  - Free access to computing resources (GPUs)

- **TensorFlow Lite**
  - Set of tools to help developers run TensorFlow models on mobile, embedded, and IoT devices.
    - TF Lite Converter
      - Converts TensorFlow models into an efficient form for use by the interpreter.
    - TF Lite Interpreter
      - Runs specially optimized models on mobile phones, embedded Linux devices and microcontrollers.

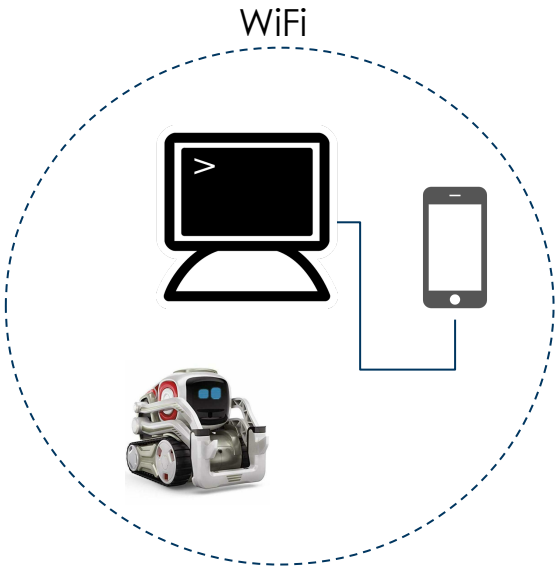
# **Edge:** Handle Requests and Run Inference

- **Files**
  - TensorFlow Lite Model & Labels
    - 13 MB
  - TensorFlow Lite Interpreter
    - 5 MB
  - IoT Image
    - 8 kB
- **Client Requests**
  - Classify
  - Exit



# **IoT:** Cozmo Robot

- **System**
  - Cozmo's engine packaged in iPhone app
  - Cozmo connected to iPhone via WiFi network
  - iPhone connected to PC via USB
    - Cozmo SDK ran on PC
- **Functions**
  - Remote Control & Camera Feed
    - Controlled via Flask web page
  - Send images
    - Send via *scp* to Edge

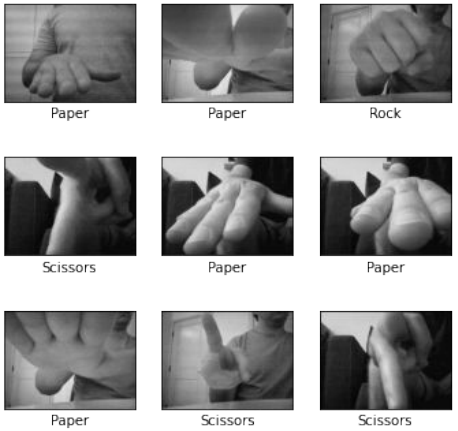


# **Classification:** Off-the-shelf

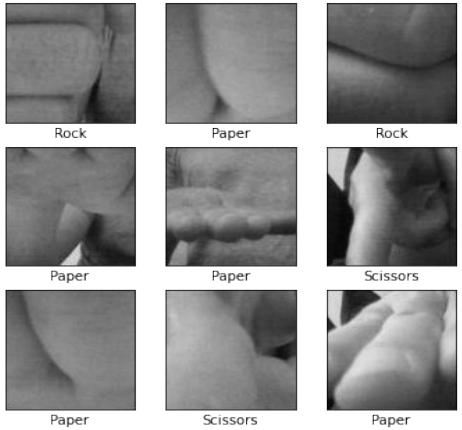
- Pre-trained model and public dataset
- **Model:** Mobilenet
  - 8.9 MB
  - ~300 ms inference time
- **Dataset**: ImageNet 2012
  - 999 classes
    - Cup, Toaster, Stove, Microwave, Candle, etc.
  - ~1,000 training images/class

# **Classification:** Rock, Paper, Scissors

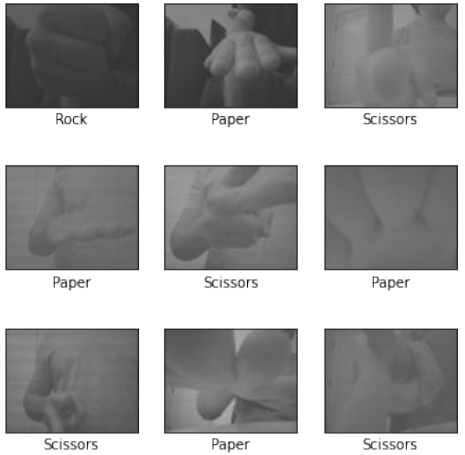
- **Dataset**: Images from Cozmo
  - ~25,000 training images
    - ~4,000 raw images
    - ~21,000 augmented images
      - Random contrast/brightness/crop/resize/etc.




Raw



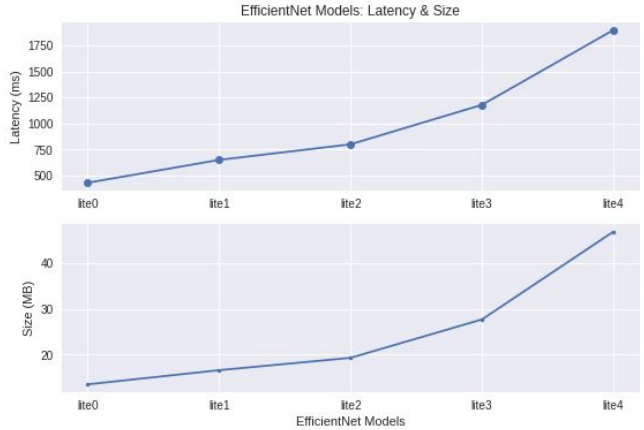

Resize

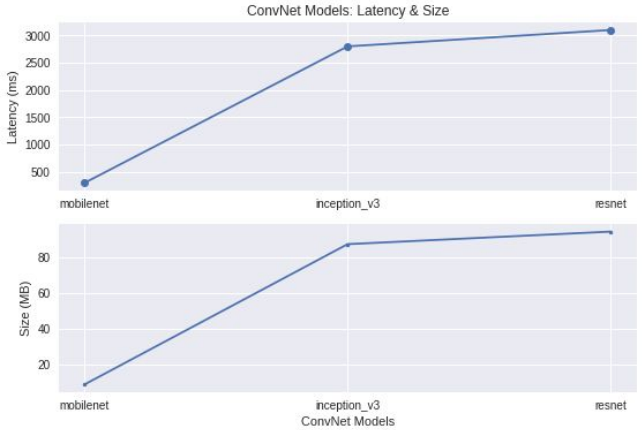



Contrast

# **Classification:** Rock, Paper, Scissors

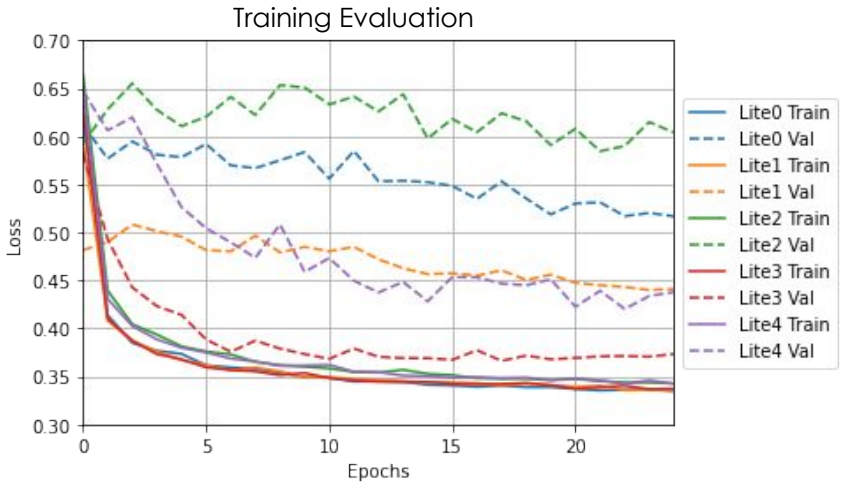
- **Model:** EfficientNets
  - Mobile-size models
  - Quick inference
  - Minimize parameters and FLOPS, maximize accuracy
    - *Compound Scaling Method*

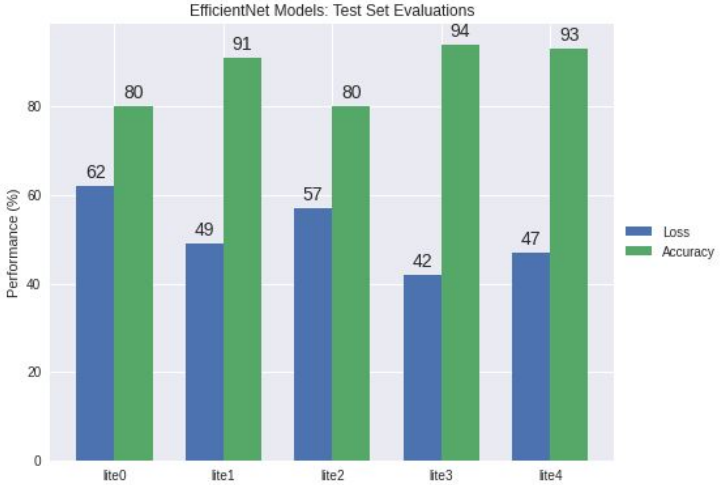


# **Classification:** Rock, Paper, Scissors

- **Model:** EfficientNets
  - Fastest Inference: EfficientNet-lite0 (14 MB)
  - Highest Accuracy: EfficientNet-lite3 (28 MB)



# DEMO