Online Continual Learning for Embedded Devices

Tyler L. Hayes and Christopher Kanan
tlh6792@rit.edu; ckanan@cs.rochester.edu

Code Available
https://github.com/tyler-hayes/Embedded-CL

Overview

- Continual learning is needed for making **real-time updates and inferences** on embedded devices
- Embedded continual learning can enable **new applications**: home robots, user personalization on mobile devices, augmented/virtual reality headsets, and more
- Challenge: Embedded devices have **limited memory** and **compute capacity** due to hardware constraints
- Goal: Perform a comprehensive study of how well online continual learners perform with mobile CNNs

Embedded Continual Learning Criteria

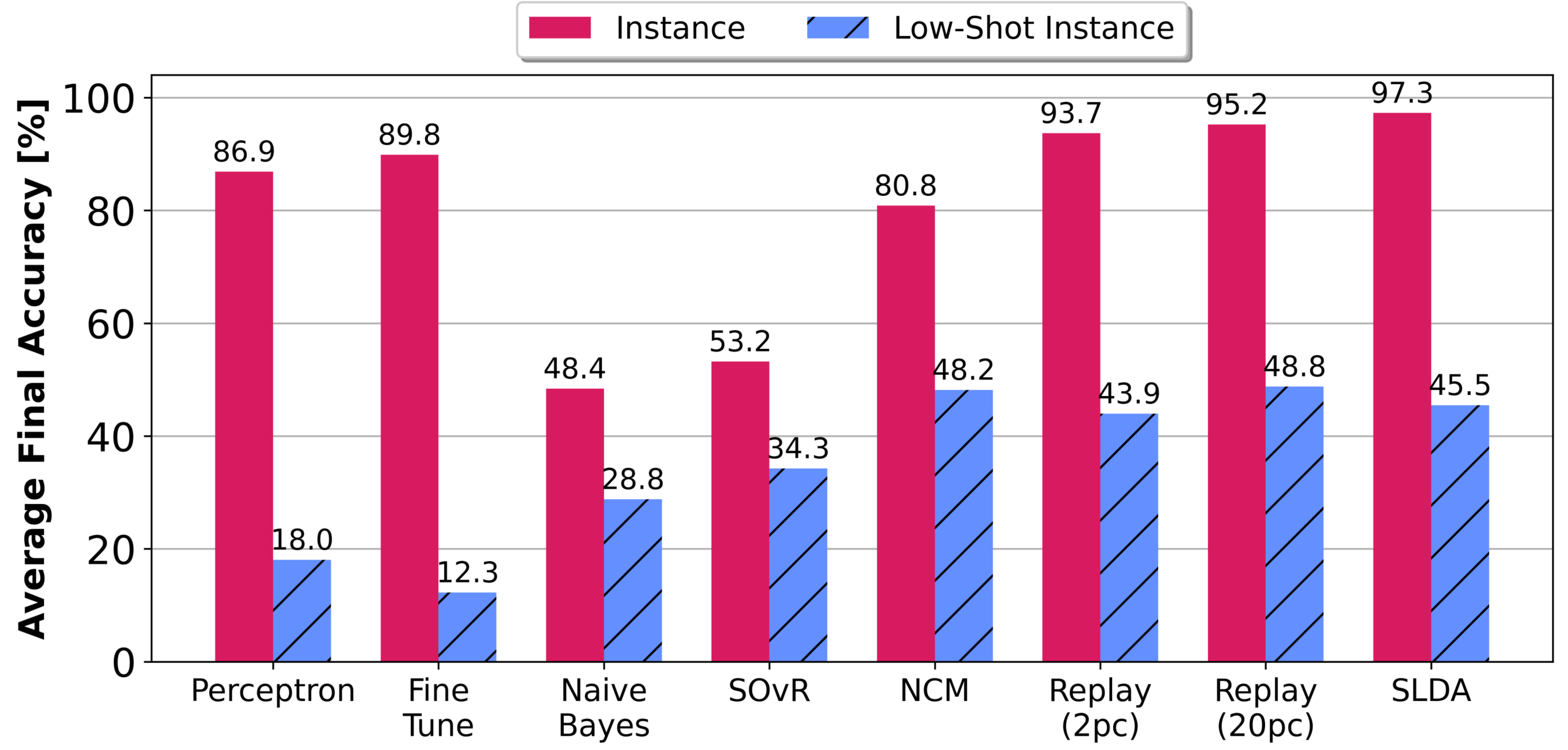
- **Online learning** and inference in a **compute** and **memory constrained** environment
- The ability to learn from **data in any order** without catastrophic forgetting
- Making no assumptions about the availability of **task labels** during inference
- Efficiently learning and generalizing with as **few labeled examples** as possible

Experimental Setup

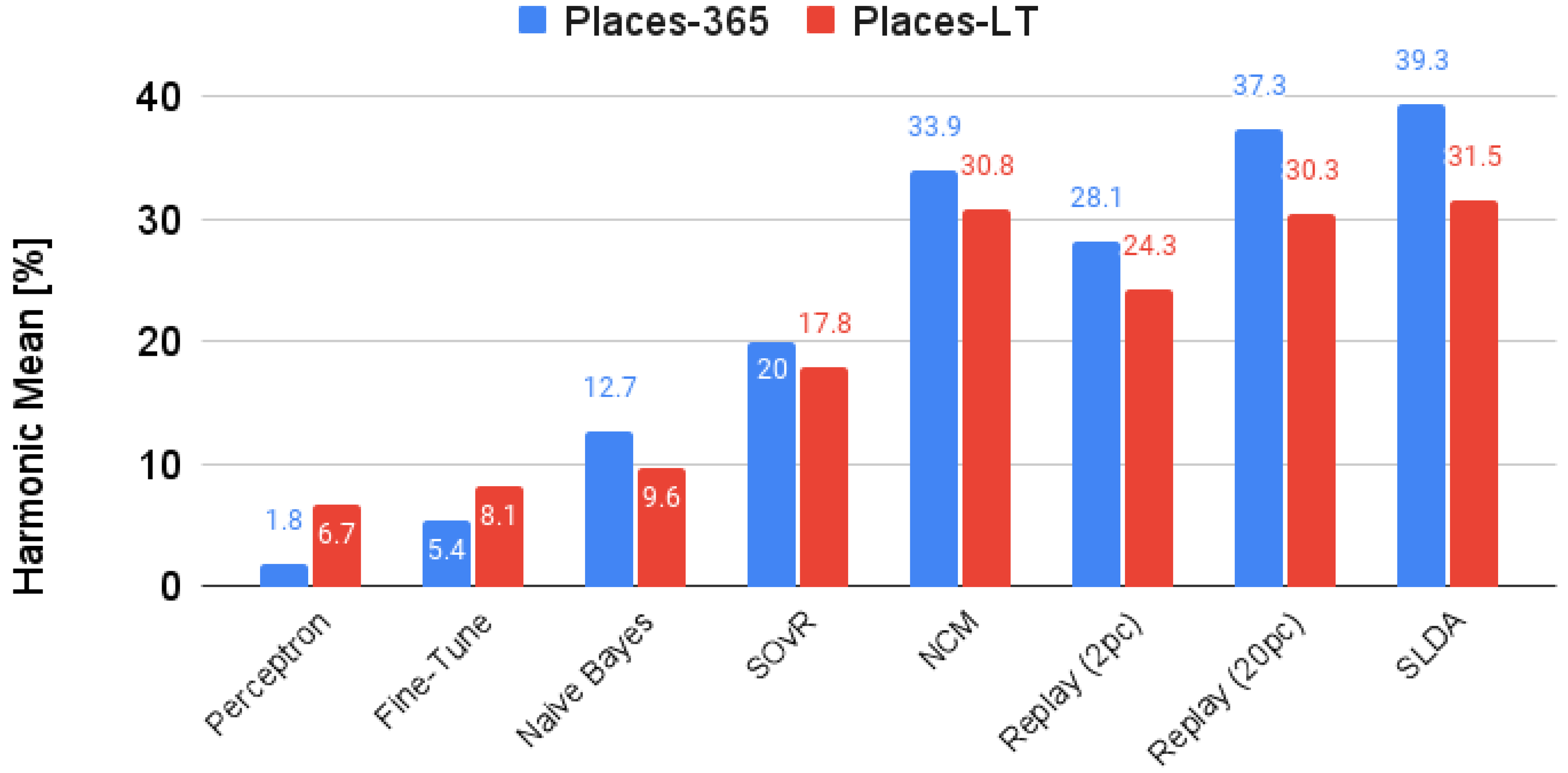
- **Pre-train** mobile CNN backbone with supervised learning on **ImageNet-1k**
- Incrementally update the **output layer** of the CNN
- Mobile CNNs: MobileNet-v3 (Small & Large), EfficientNet (B0 & B1), ResNet-18
- Online Learners: Perceptron, Fine-Tune, Naïve Bayes, Stream One-v-Rest (SOvR), Nearest Class Mean (NCM), Replay, Stream Linear Discriminant Analysis (SLDA)
- Datasets: OpenLORIS (Videos), Places-365, Places-Long-Tail (Places-LT)

Results

OpenLORIS Results



Places Results



Overall Results

Factors Evaluated:

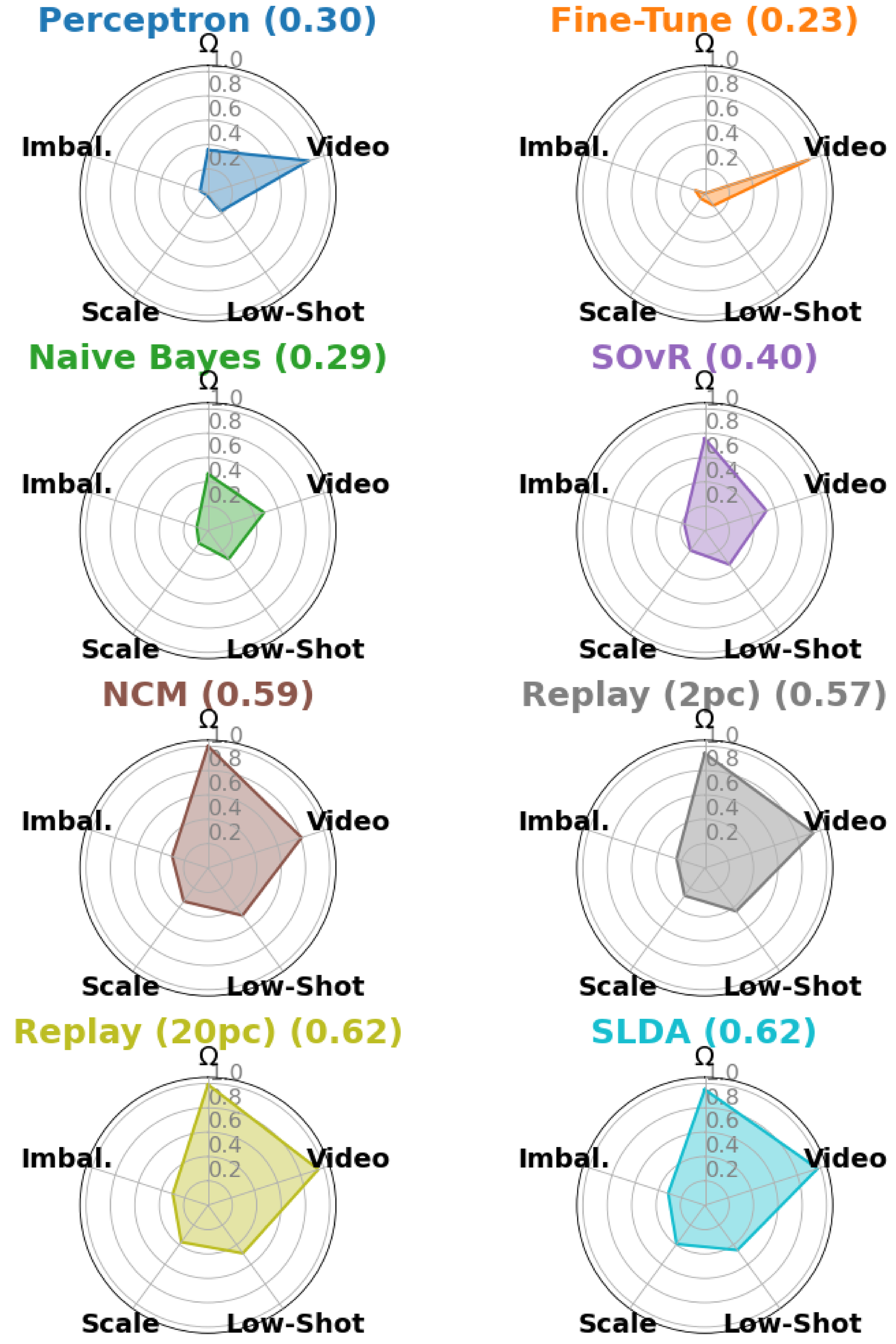
Ω: classification efficacy, compute, memory

Video: ability to learn from temporally correlated videos

Low-Shot: ability to generalize from few inputs

Scale: ability to scale to large-scale data

Imbal: ability to perform well on imbalanced data



Summary

- We established **online continual learning baselines for embedded applications** subject to: imbalanced data streams, large-scale data streams, video streams, and low-shot video streams
- **Replay (20pc)** and **SLDA** performed the best across experiments
- **NCM** also performed well and used less memory and compute
- **EfficientNet CNNs** yielded the best performance across methods

Acknowledgements:

This work was supported in part by NSF awards #1909696, #2047556, and #DGE-2125362.