Odessa: Enabling Interactive Perception Applications on Mobile Devices

Moo-Ryong Ra, Anmol Sheth, Lily Mummert, Padmanabhan Pillai, David Wetherall, Ramesh Govindan
USC ENL, Intel Labs Seattle/Pittsburgh, University of Washington

Introduction

Interactive Perception Applications
- A set of interesting applications for the mobile device are emerging.
- They often use machine-learning or computer-vision based algorithms, which can be naturally expressed as data-flow graphs.
- Example applications include:
  - Realtime face recognition application.
  - Gesture recognition application to control the TV.
  - Object detection and its pose estimation application.

Requirements and Challenges
- Two metrics
  - FPS (Frames Per Second): how many frames we can process within a second.
  - Makespan: how long the system takes for the recognition task.
- We want high FPS and low makespan.
- Requirements are highly coupled with the user experience.
  - Both are important. We need to optimize both metrics simultaneously.

Problem Description

Techniques for Speed-Up
- Offloading computations to the cloud infrastructure.
- Parallelization of the sub-components.
  - Data Parallelism: we can use multiple workers on the single task.
  - Pipeline Parallelism: we can run multiple frames simultaneously by using separate worker threads across each computation stages.

Decision Problem
- Deciding the right partition a priori is hard due to a lot of variability.
  - Input variability.
  - Network variability.
  - Device heterogeneity.
- Determining the level of parallelism makes the problem harder.
  - How many worker threads for the stage x?
  - How many tokens in the pipeline?

Proposed Solution: Odessa – Offloading DEcision System for Streaming Applications

Measurement Study
- Variability
- Data Parallelism
- Pipeline Parallelism

Odessa Runtime Design
- It must simultaneously achieve low makespan and high throughput (FPS).
- It must react quickly to changes in input complexity, device capability, or network conditions.
- It must have low computation and communication overhead.
- Incremental offloading and parallelism decision algorithm focusing on the bottleneck stage.

Evaluation
- End-to-End Performance

Performance Comparison