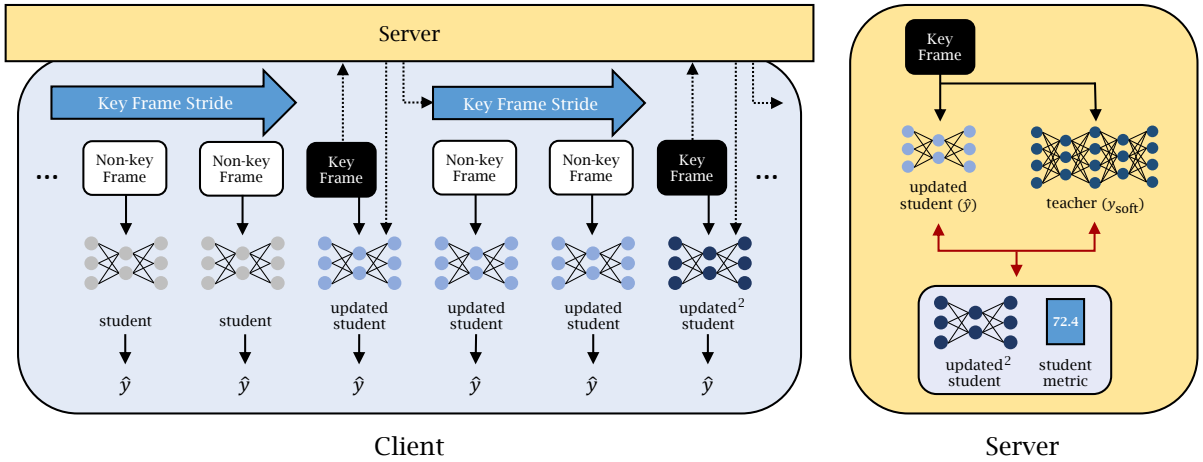
ShadowTutor: Distributed Partial Distillation for Mobile Video DNN Inference

Jae-Won Chung, Jae-Yun Kim, Soo-Mook Moon
Seoul National University, South Korea

Virtual Machine & Optimization Laboratory
Dept. of Electrical and Computer Engineering
Seoul National University

In a Nutshell

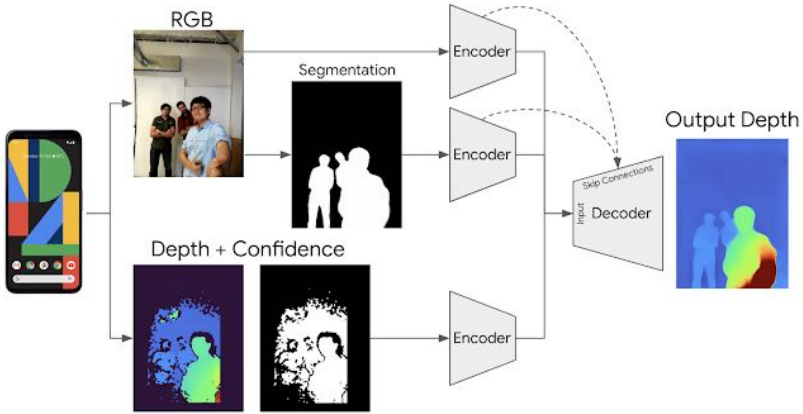
CONTEXT	The server helps an edge device perform DNN inference on videos
PROBLEM	Significant network traffic incurred by naïve computation offloading
THROUGH	Knowledge distillation across the network
NOVELTY	Distributed partial KD, Key frame selection, Analytic modelling
EFFECTS	95% reduction in network data transfer, 3x throughput

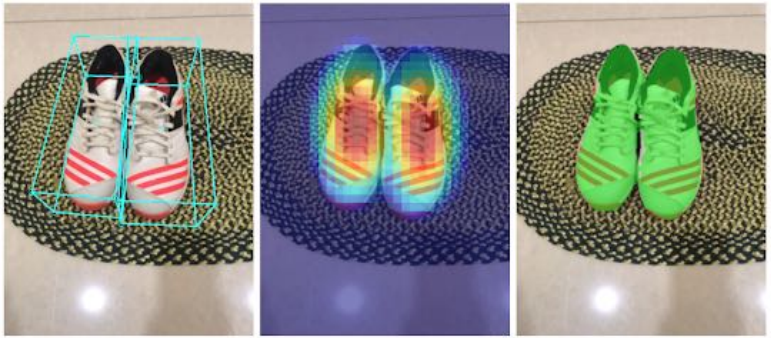


Outline

1. **Introduction**
   - Mobile DNN inference and two major approaches
2. **Motivation**
   - Temporal coherence in videos
3. **Background**
   - Knowledge distillation
4. **ShadowTutor**
   - High-level overview
   - Proposed algorithms
   - Analytic models
5. **Evaluation**
   - Setup and implementation
   - Experimental results
6. **Conclusion**

Introduction





https://ai.googleblog.com/2020/04/udepth-real-time-3d-depth-sensing-on.html (left)
https://ai.googleblog.com/2020/03/real-time-3d-object-detection-on-mobile.html (right)

Introduction

Two major approaches to mobile DNN inference

- Architecting on-device models, devising efficient operations
- Computation offloading to cloud/edge servers

Both have advantages

- On-device: No server workload. Data is kept private.
- Offloading: Large but high-quality models can be employed.

But not without limitations

- On-device: Model is less accurate and tailored to specific devices. No adaptation to context.
- Offloading: High network traffic and server load. The mobile device does nothing.

Motivation

When it comes to video data, offloading fails hard

- Frame-by-frame communication overhead
- Directly affected by adverse network conditions
- Results in lag in inference result, and fluctuations in FPS

Video frames bear temporal coherence

- Frames nearby share object, movement speed, ambience, etc
- If we can do well on one frame, doing well on nearby frames is free

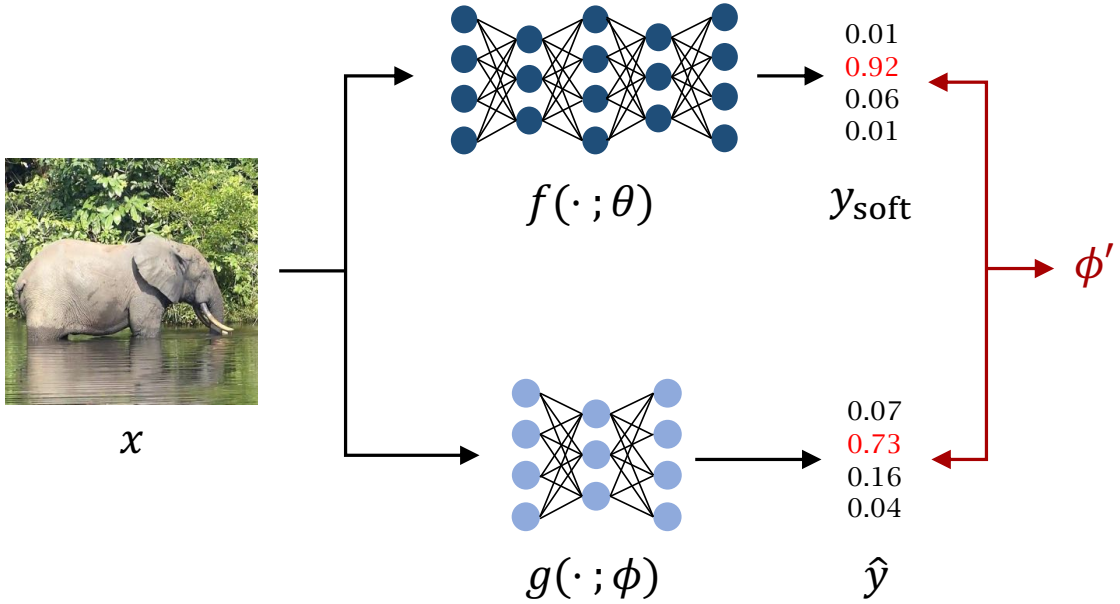
Motivation

Can we enjoy the best of both worlds?

- Little network traffic and server load
- The performance and generality of a large model
- Model adaptation to current data
- Computation power of mobile devices is reasonably leveraged

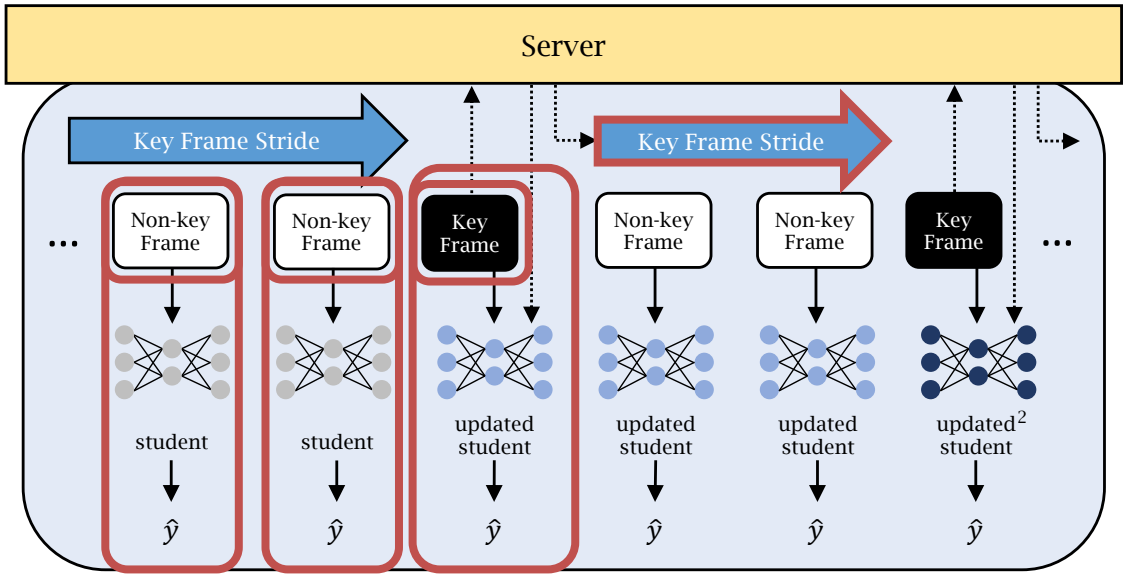
Background

Knowledge Distillation

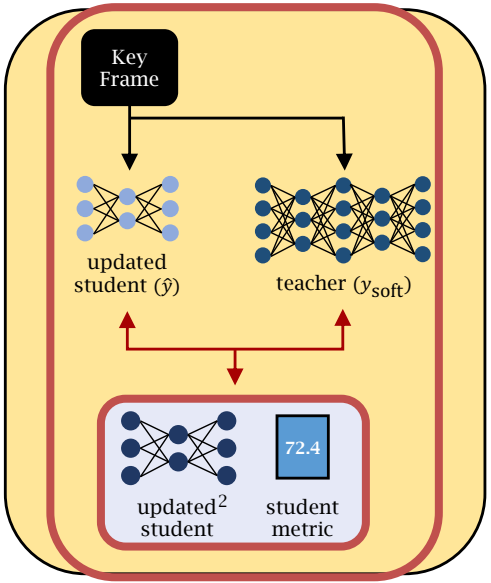


ShadowTutor

A high-level overview



Client



Server

ShadowTutor – Proposed Algorithms

Student training (Alg. 1)

- Thresholded early stopping with maximum number of optimization steps

Key Frame Striding (Alg. 2)

- Determines the distance to the next key frame based on the current stride and student metric
- Longer stride if student performs well, shorter stride otherwise.

Server loop (Alg. 3)

- Partial knowledge distillation: Only update the backend parameters.

Client loop (Alg. 4)

- Asynchronous inference: Do not wait until the updated parameters arrive.

ShadowTutor – Analytic Models

Accurate analytical models for throughput and network traffic

- Aids the configuration of system parameters for the service provider

Throughput upper-bound

$$\frac{\texttt{MAX\_STRIDE}}{(\texttt{MAX\_STRIDE} - \texttt{MIN\_STRIDE})t_{si} + \max(\texttt{MIN\_STRIDE}t_{si}, t_{net} + t_{ti})}$$

Throughput lower-bound

$$\frac{\texttt{MIN\_STRIDE}}{\texttt{MIN\_STRIDE} \times t_{si} + \texttt{MAX\_UPDATES} \times t_{sd} + t_{ti} + t_{net}}$$

Network traffic upper-bound

$$\frac{s_{net}}{\max(\texttt{MIN\_STRIDE} \times t_{si}, t_{net} + t_{ti})}$$

Network traffic upper-bound

$$\frac{s_{net}}{\texttt{MAX\_STRIDE} \times t_{si} + \texttt{MAX\_UPDATES} \times t_{sd} + t_{ti} + t_{net}}$$

Evaluation

Setup

- The server has a powerful GPU (NVIDIA RTX2080ti)
- The client has a weak GPU (NVIDIA Jetson Nano), comparable to recent mobile devices
- Network bandwidth limited to 80Mbps
- Semantic segmentation on HD 25fps videos (LVS dataset)

Implementation

- OpenMPI + PyTorch + Detectron2
- Code open at Github: https://github.com/jaywonchung/ShadowTutor

Evaluation

Experimental results

Throughput (FPS): Increased more than 3x

Camera	Scene	Partial	Full	Naive
fixed	animals	6.55(762.5)	6.21(804.5)	2.09(2391.3)
fixed	people	**6.60(757.4)**	6.43(777.0)	2.09(2391.3)
fixed	street	6.50(768.8)	5.95(840.5)	2.09(2391.3)
moving	animals	6.57(760.5)	6.27(796.5)	2.09(2391.3)
moving	people	**6.59(758.5)**	6.36(785.8)	2.09(2391.3)
moving	street	6.41(780.2)	5.55(901.0)	2.09(2391.3)
egocentric	people	6.57(760.5)	5.89(848.5)	2.09(2391.3)
average		**6.54(764.1)**	6.08(822.0)	2.09(2391.3)

Network Data Transfer: 95% reduction

Camera	Scene	Key frame ratio			Network traffic	
		Partial	Full	Naive	Partial	Naive
fixed	animals	4.73	4.60	100.0	7.51	58.51
fixed	people	**1.96**	2.42	100.0	**3.14**	58.51
fixed	street	7.78	7.43	100.0	12.27	58.51
moving	animals	2.55	2.29	100.0	4.06	58.51
moving	people	**3.45**	4.12	100.0	5.51	58.51
moving	street	11.70	11.48	100.0	18.19	58.51
egocentric	people	**5.46**	9.75	100.0	8.70	58.51
average		**5.38**	6.01	100.0	**6.19**	58.51

Evaluation

Experimental results

Accuracy (mIoU): 72% of teacher with 1% param

Camera	Scene	Wild	P-1	P-8	F-1	Naive
fixed	animals	14.34	74.31	73.27	74.47	100.0
fixed	people	13.91	81.69	81.39	81.36	100.0
fixed	street	17.28	**70.26**	69.01	63.60	100.0
moving	animals	22.31	74.94	73.80	75.21	100.0
moving	people	17.62	74.82	74.06	75.55	100.0
moving	street	18.65	**60.48**	58.61	52.94	100.0
egocentric	people	14.80	**70.42**	68.87	61.41	100.0
average		16.99	**72.42**	71.29	69.22	100.0

7FPS Videos: only 6%p drop in accuracy

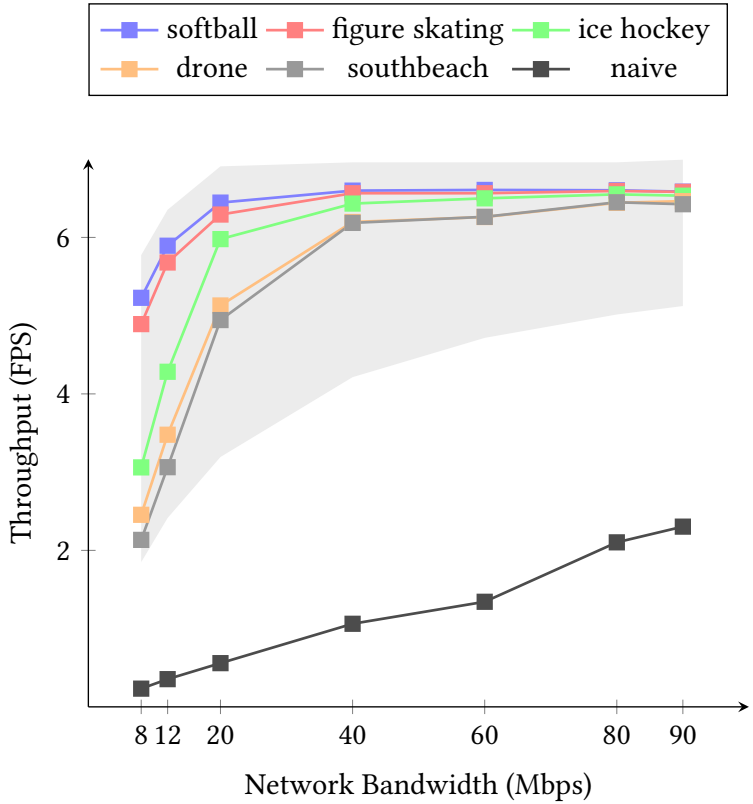
Camera	Scene	Partial-1	Partial-8	Key frame
fixed	animals	62.72	61.86	6.59
fixed	people	80.44	80.08	1.97
fixed	street	63.78	62.51	8.9
moving	animals	68.63	66.78	4.84
moving	people	73.66	72.91	4.15
moving	street	48.92	46.99	12.34
egocentric	people	67.57	66.09	5.44
average		66.53	65.31	6.32

Evaluation

Experimental results

Reducing network bandwidth

- Throughput extremely robust
- Experimental results obey analytic model



Conclusion

Can we enjoy the best of both worlds?

- Little network traffic and server load ✓
- The performance and generality of a large model ✓
- Model adaptation to current data ✓
- Computation power of mobile devices is reasonably leveraged ✓

- Good throughput ✓
- Robustness to adverse network condition ✓