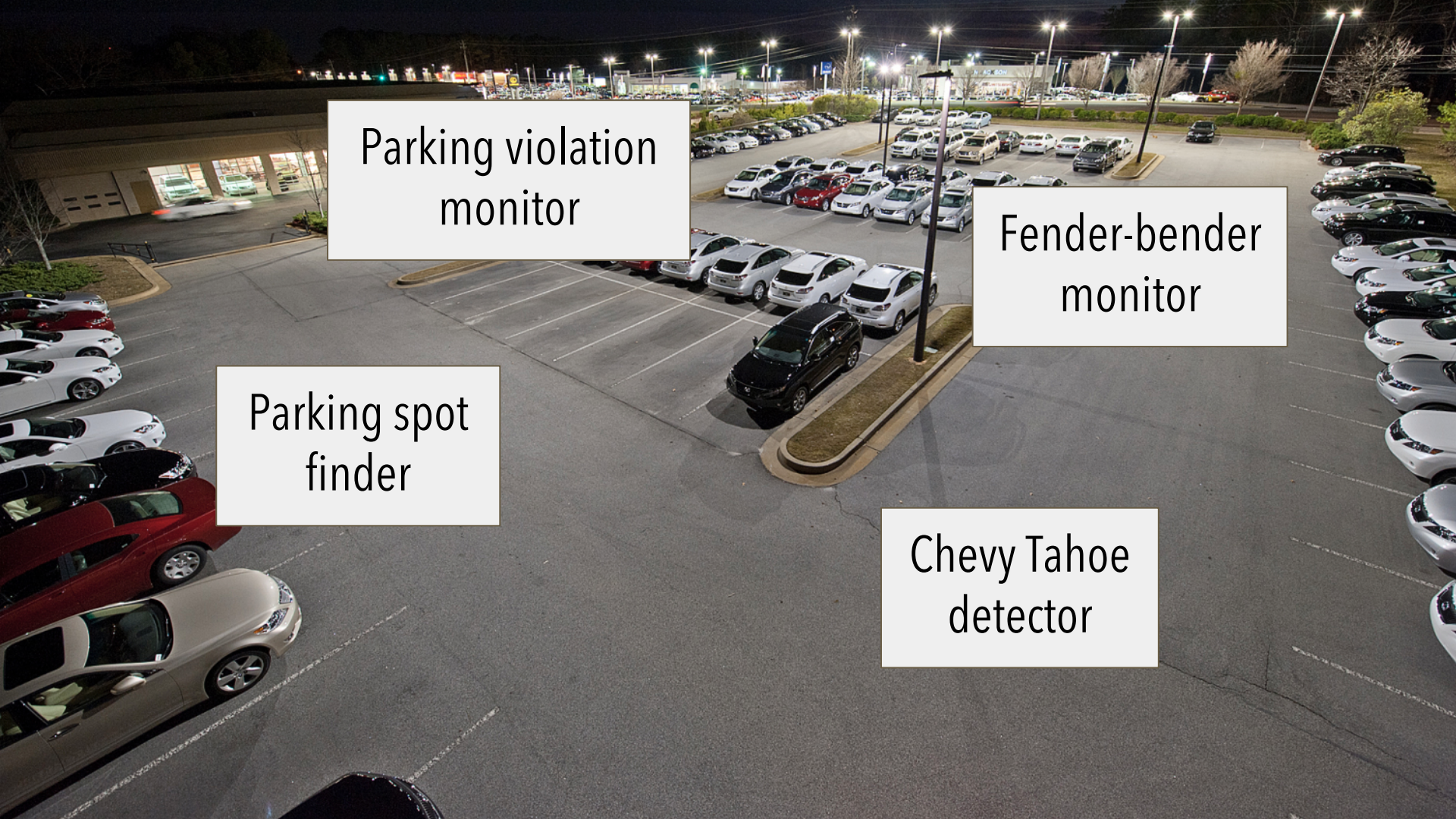
Mainstream: Dynamic Stem-Sharing for Multi-Tenant Video Processing

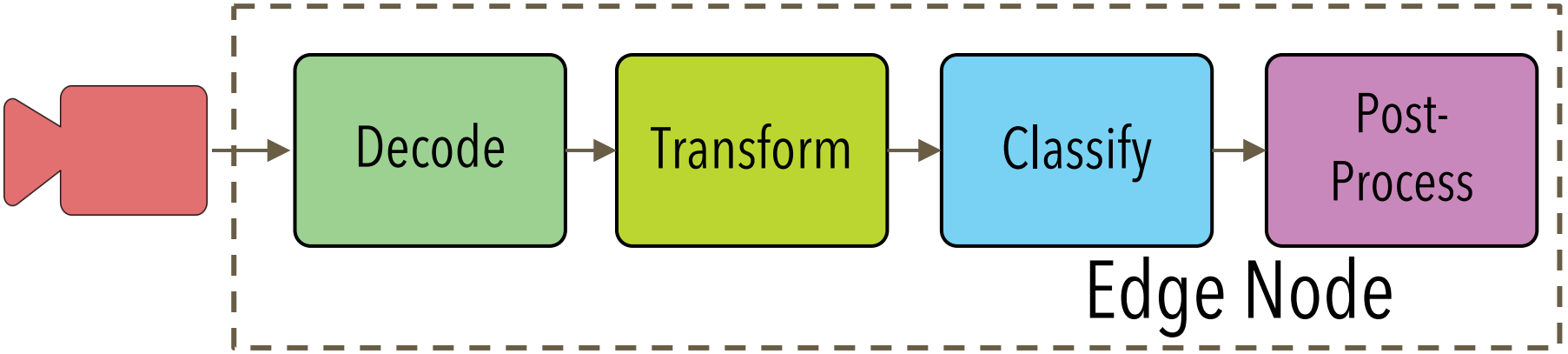
Angela Jiang
Daniel L.-K. Wong, Christopher Canel, Ishan Misra,
Michael Kaminsky, Michael Kozuch, Babu Pillai,
David G. Andersen, Greg Ganger


Parking violation monitor
Fender-bender monitor
Parking spot finder
Chevy Tahoe detector

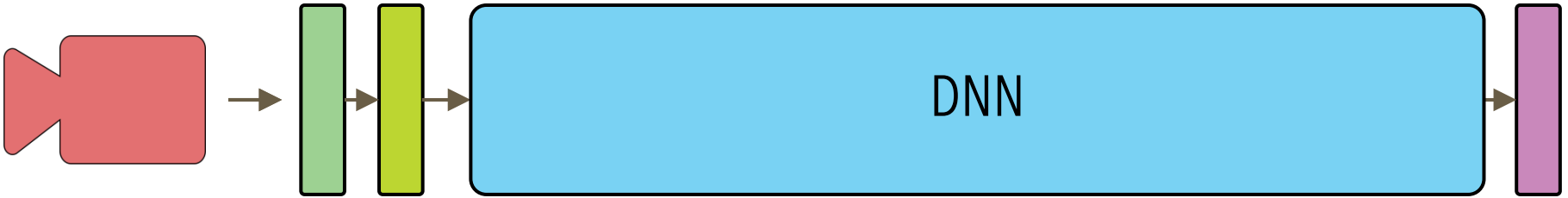
Mainstream: runtime optimization of applications

- Problem: Video analysis applications do DNN *inference*
  - Running several full DNNs becomes very slow
- Currently: Applications developed in *isolation* and *offline*
  - Without understanding of the resources available
- Mainstream reduces contention by sharing redundant computation
- Mainstream *dynamically* tunes DNNs at *runtime*

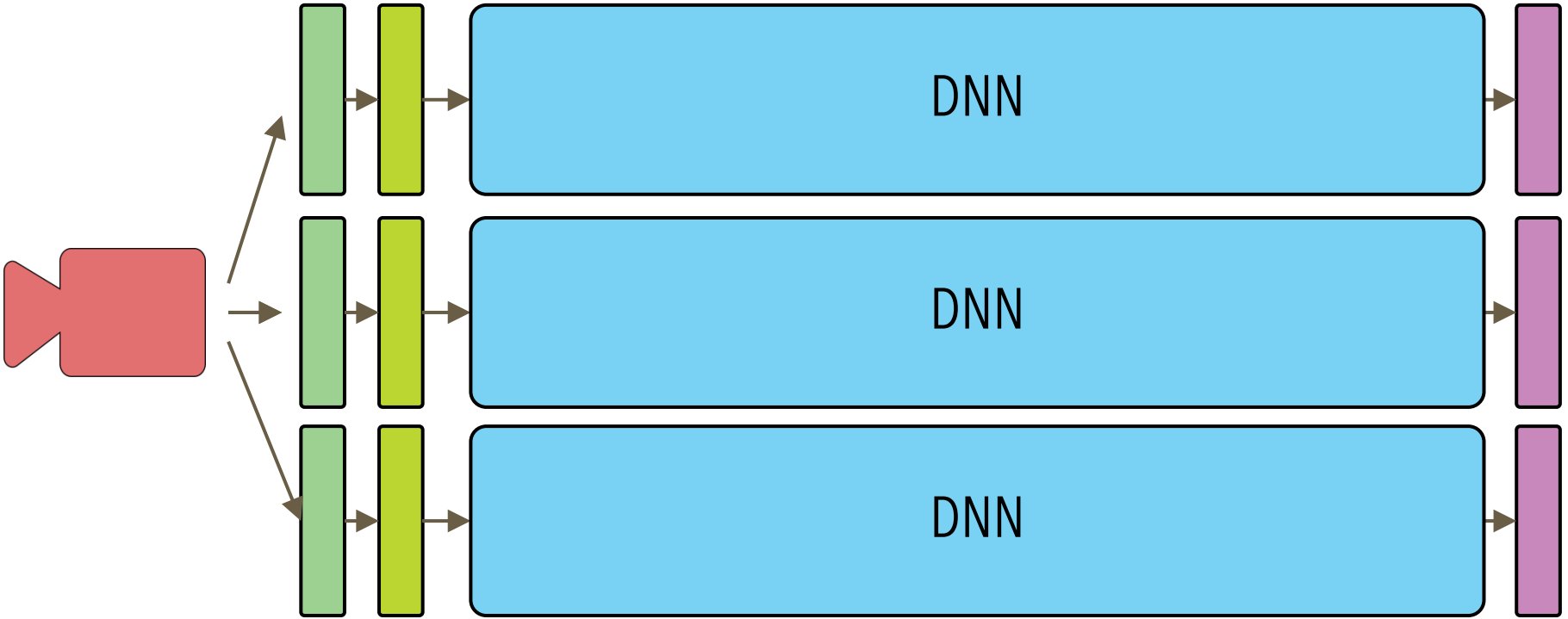
Example video analysis pipeline



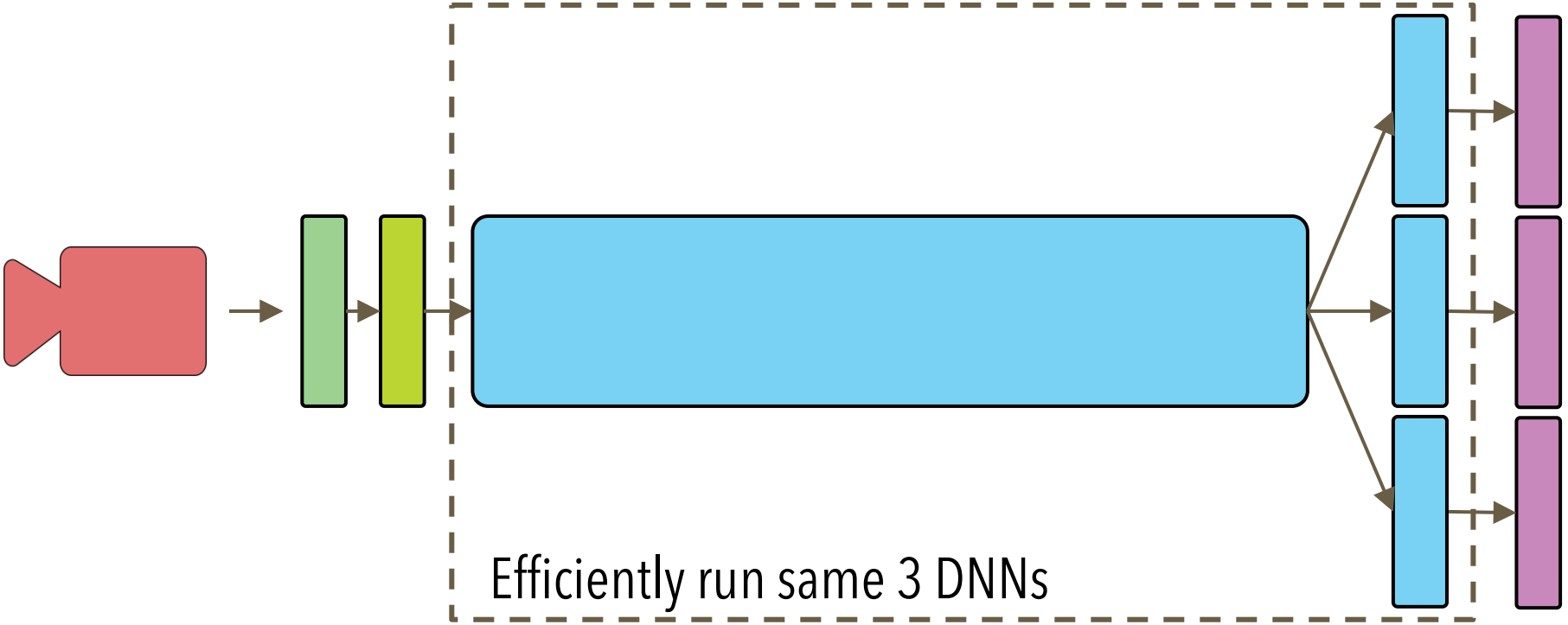
DNN is primary cost of pipeline



DNN is primary cost of pipeline

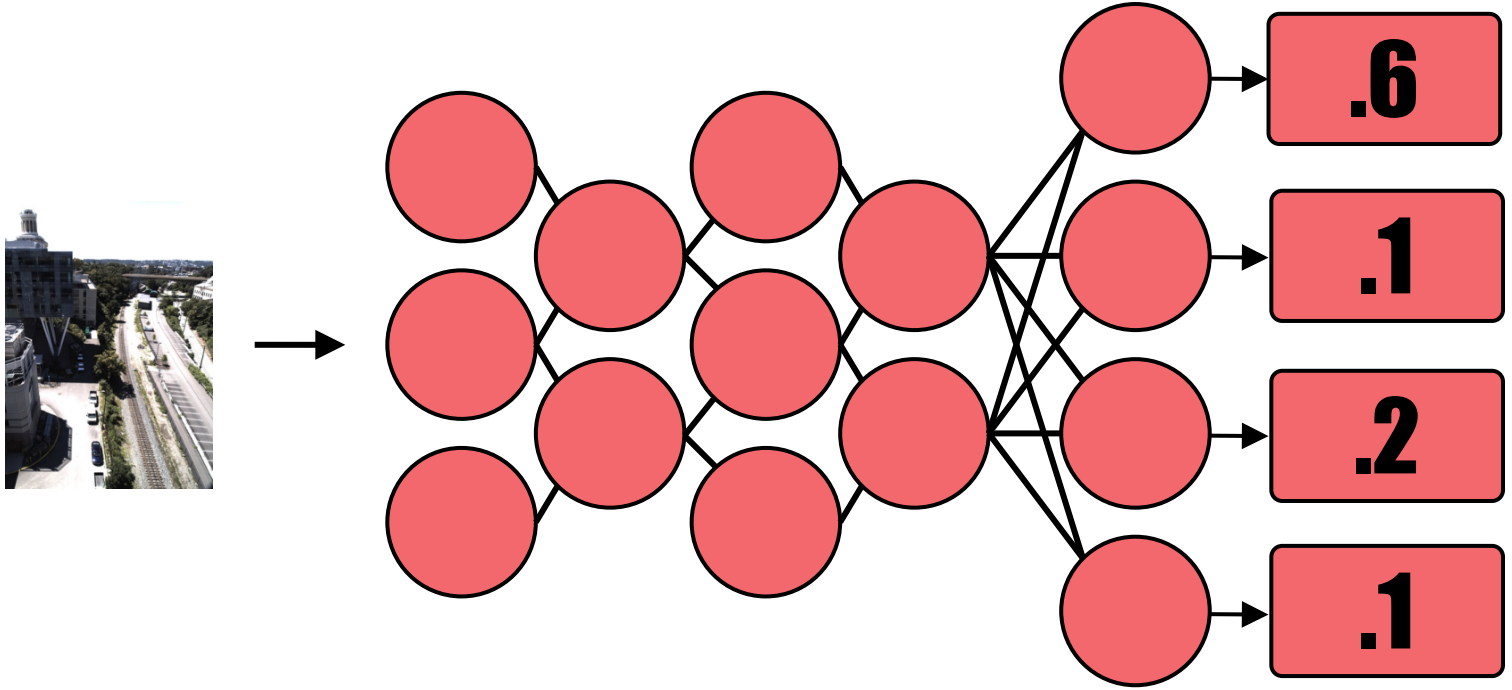


In Mainstream: Partial sharing of DNN

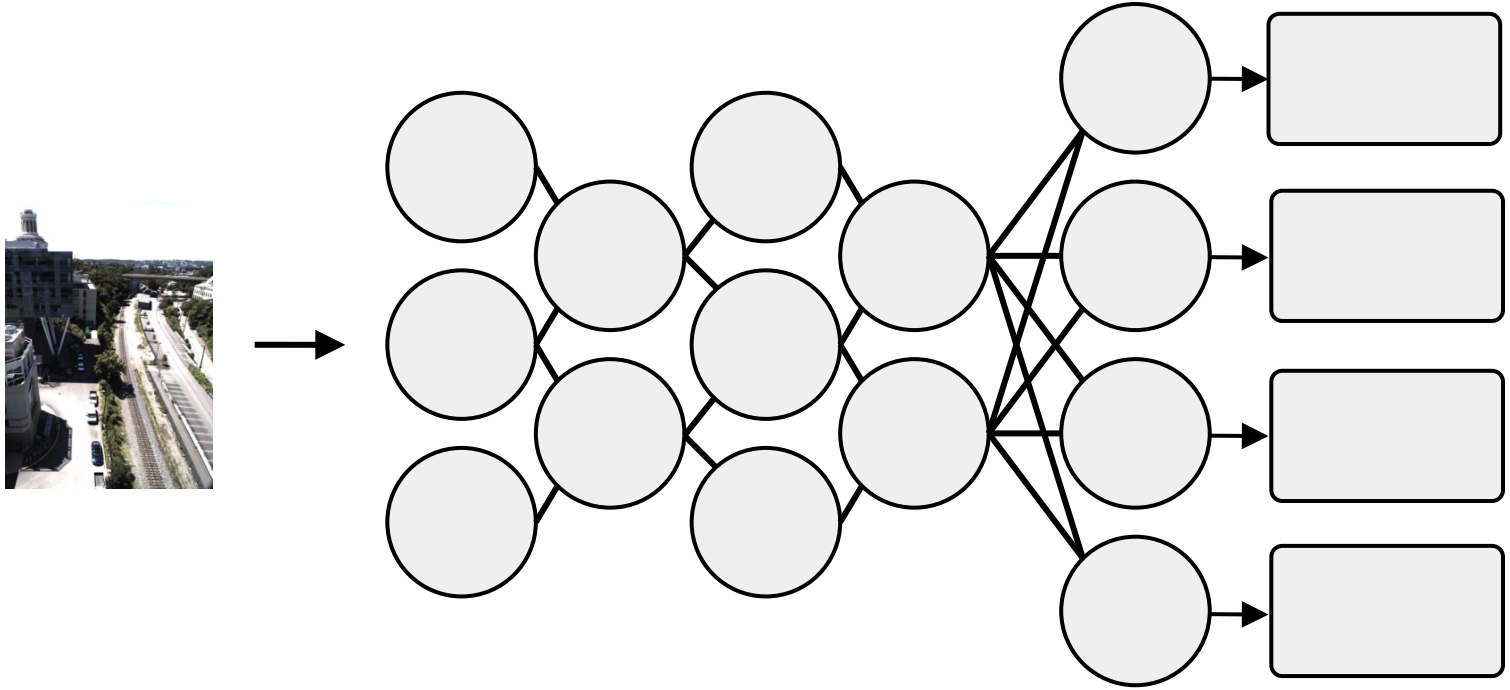


How to find redundancy between DNNs?

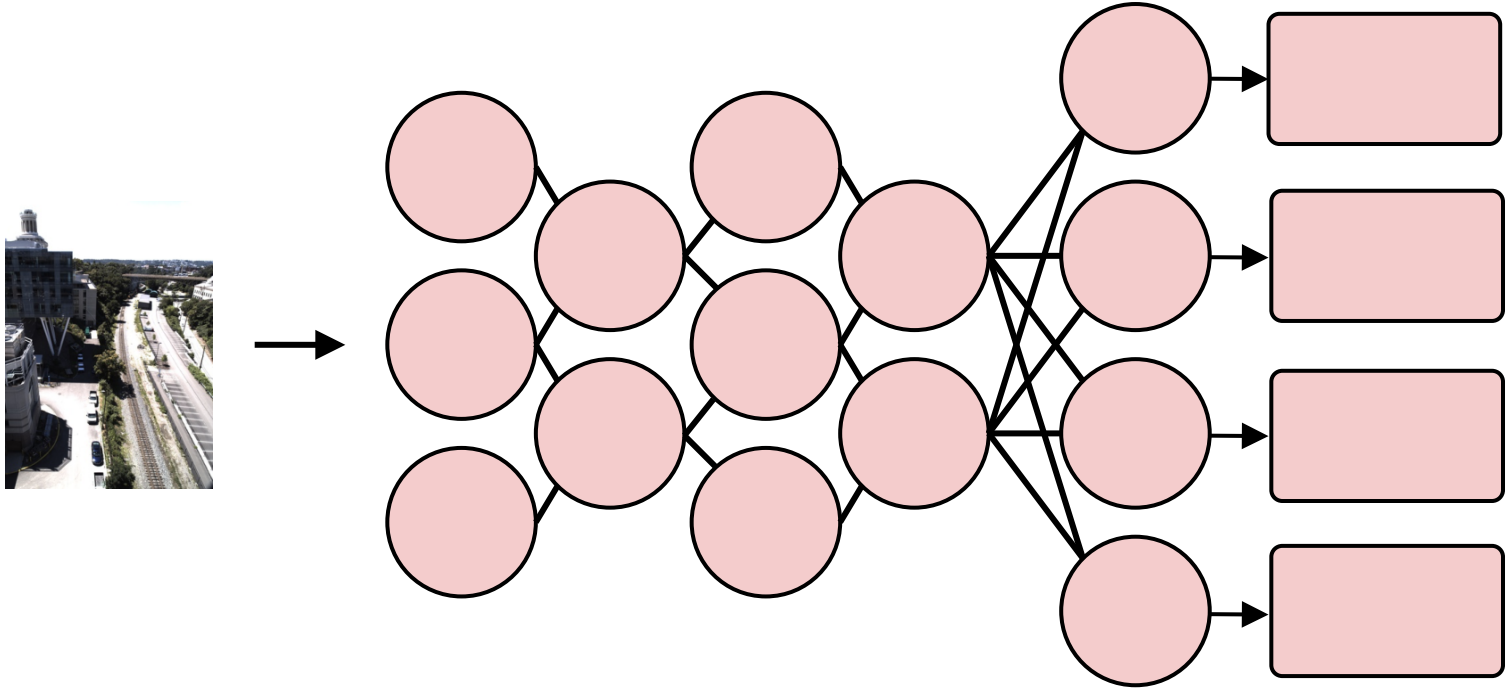
Basic DNN



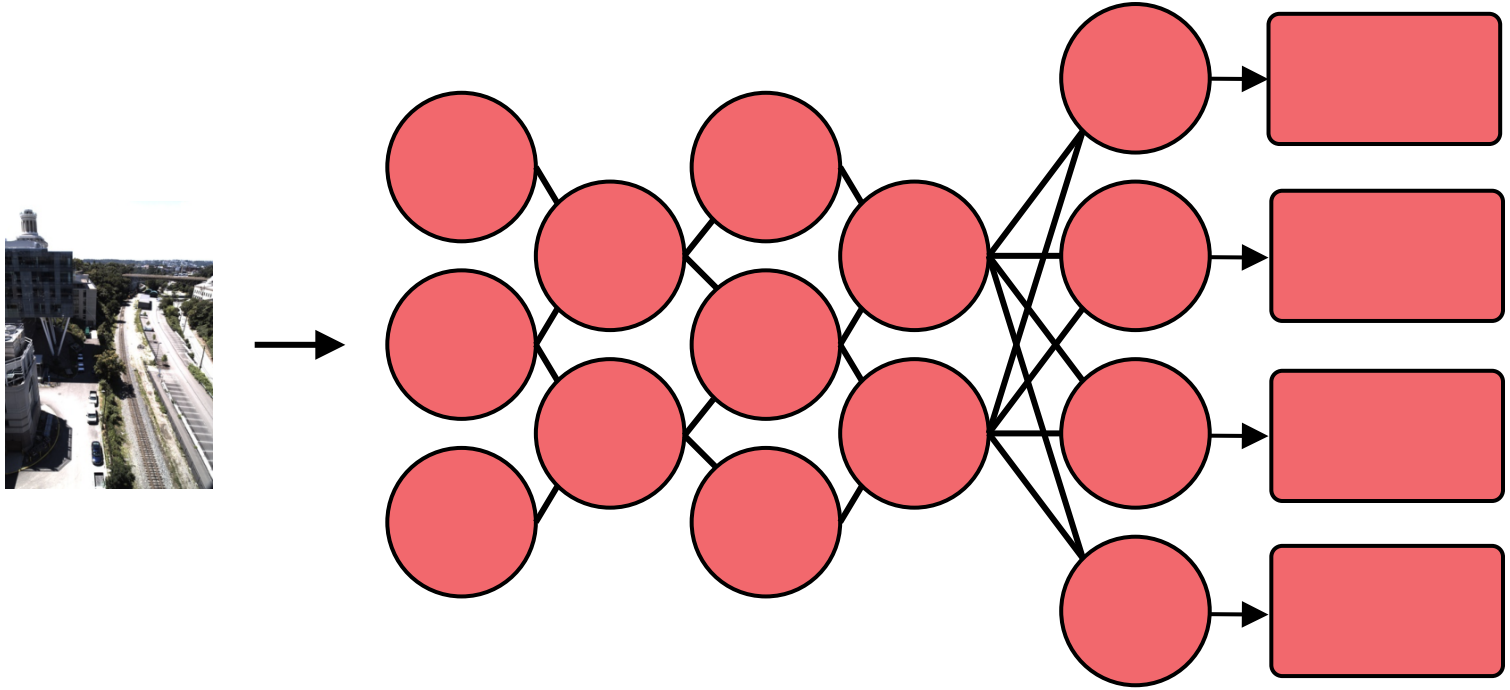
Training a DNN



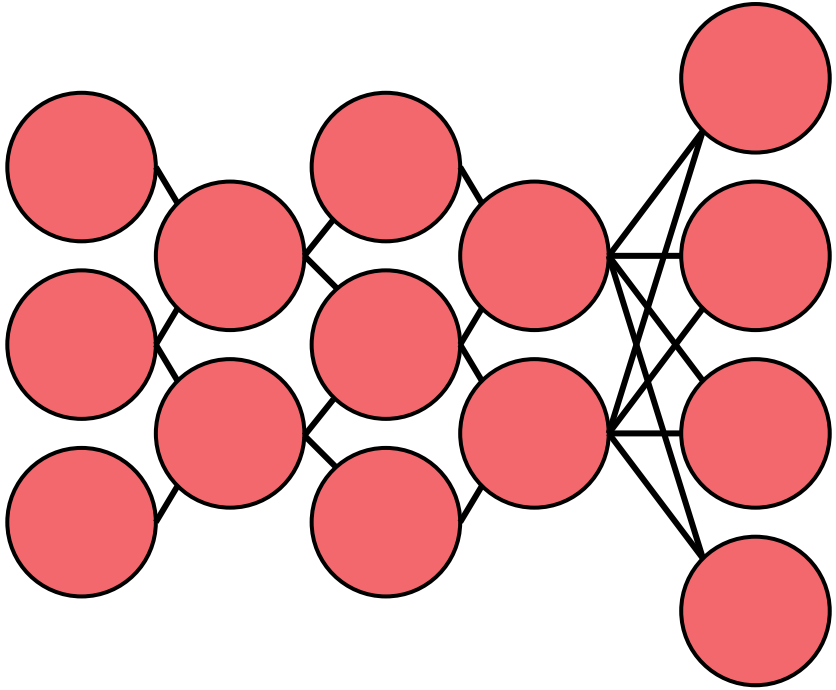
Training a DNN



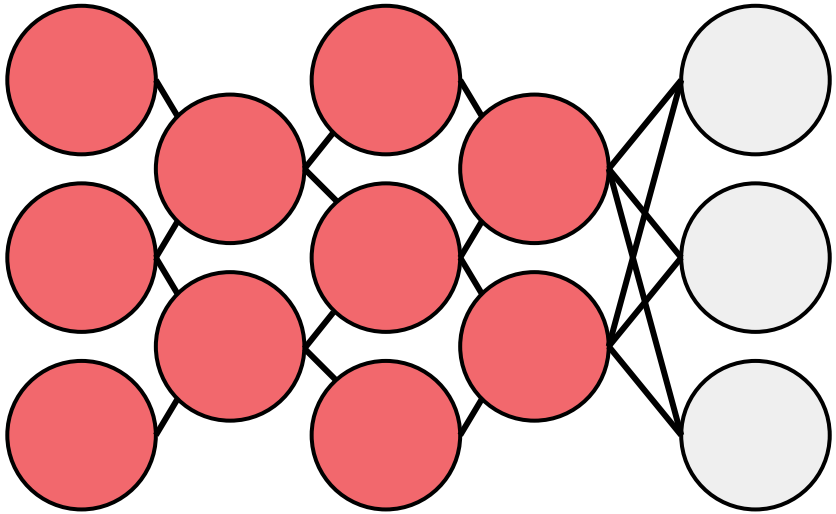
Training a DNN



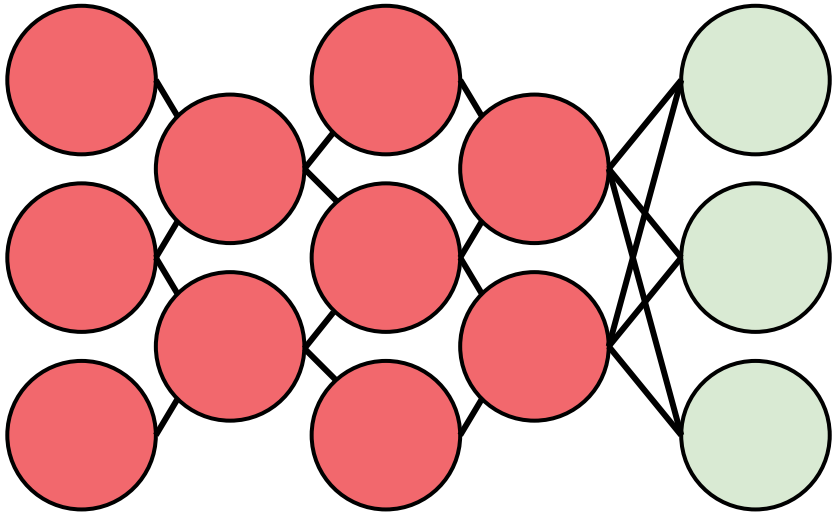
Fine-tuning DNN #1



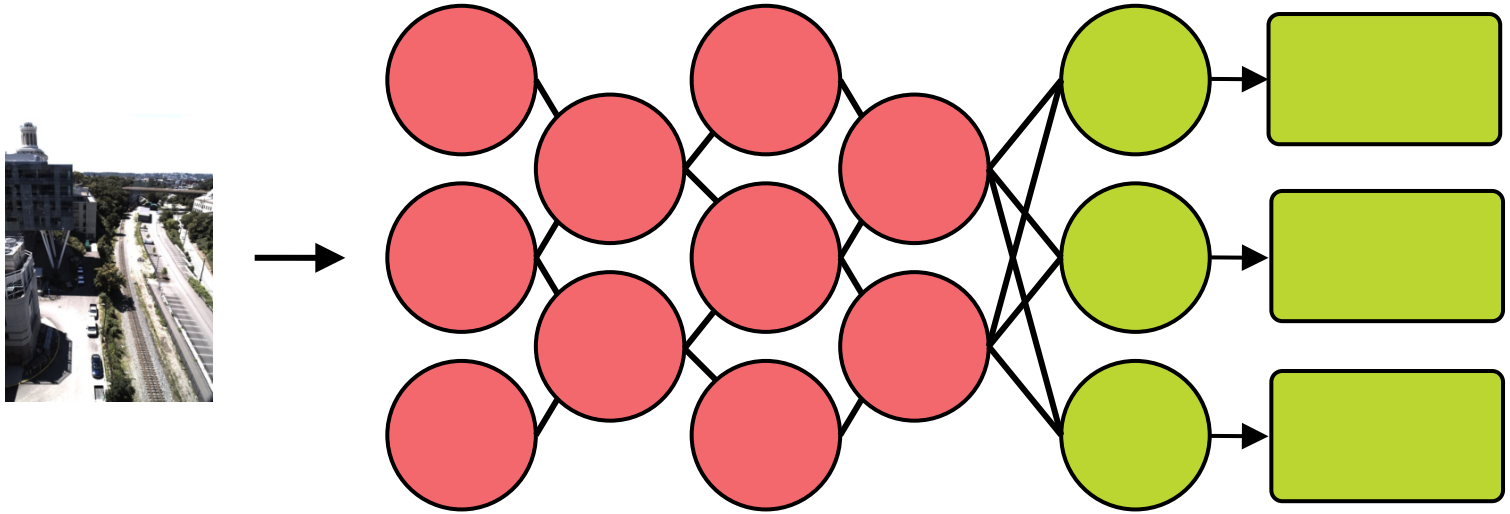
Fine-tuning DNN #1



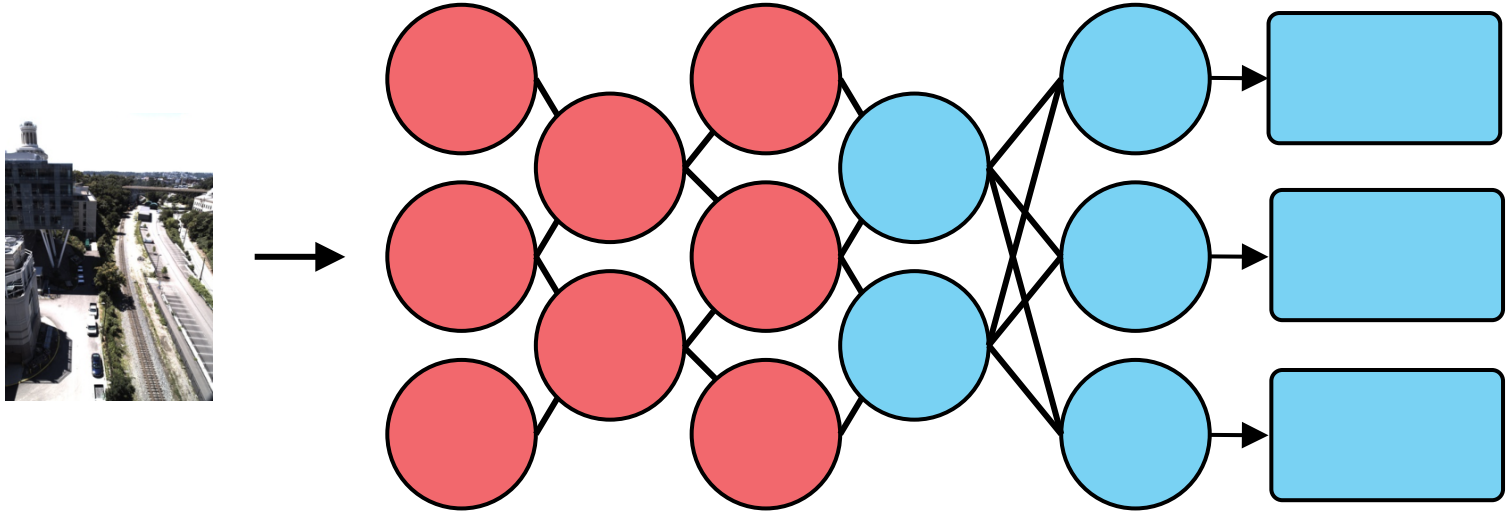
Fine-tuning DNN #1



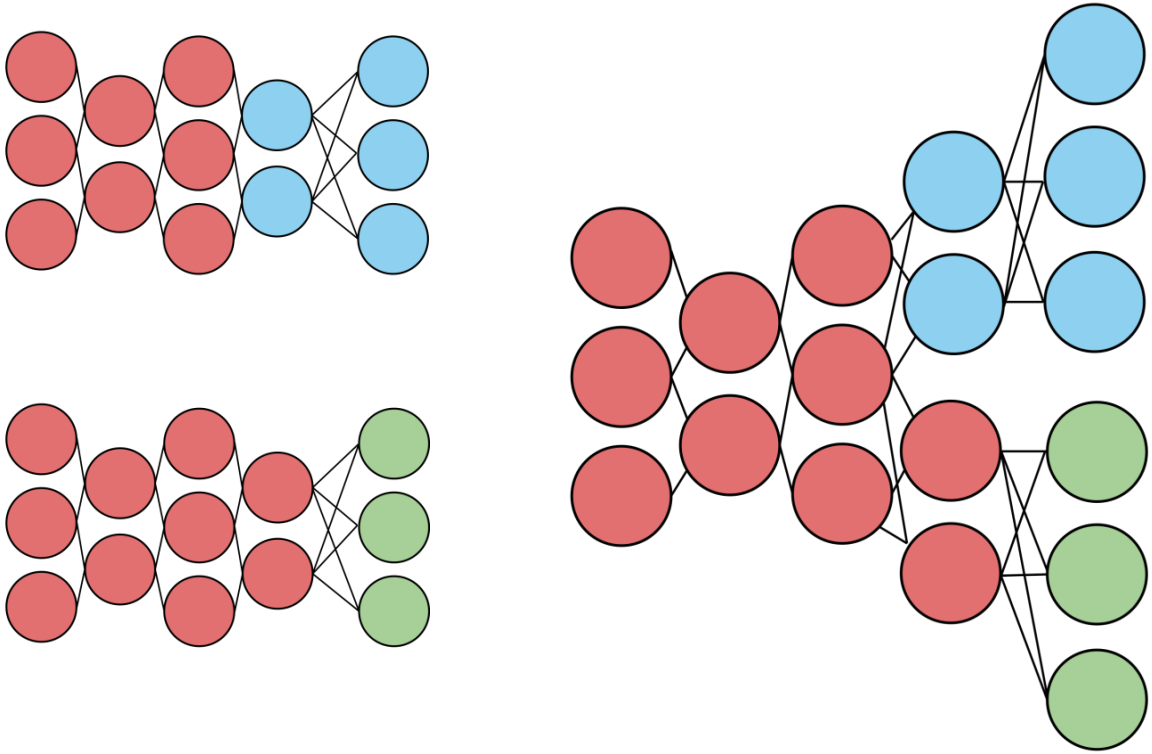
Fine-tuning DNN #1



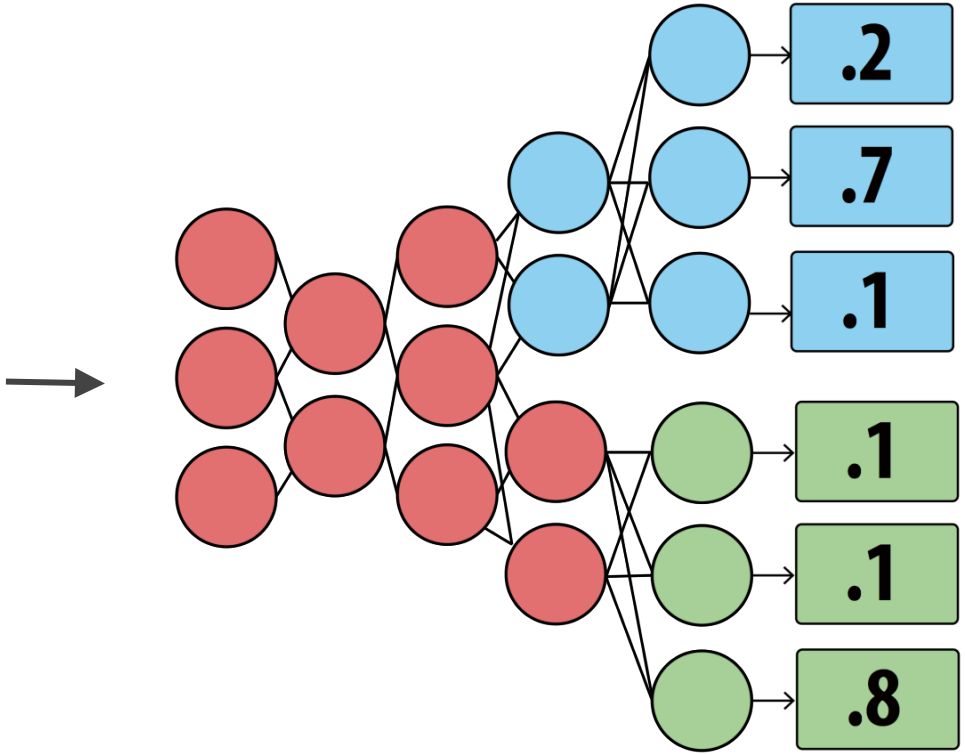
Fine-tuning DNN #2



Two DNNs as run in Mainstream



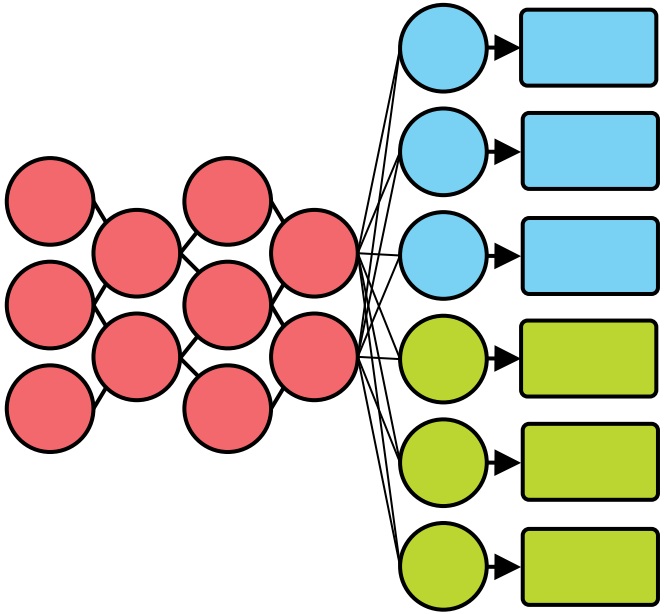
Two DNNs as run in Mainstream



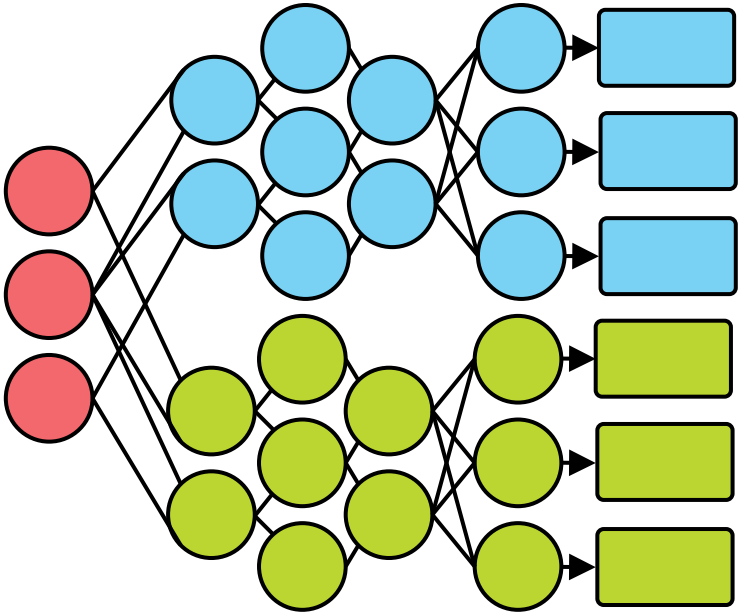
Mainstream's Approach

- Merge the video stream processing of concurrent applications
- Dynamically determine how much processing to share vs. specialize
  - Based on available resources, other applications
  - Maximize application effectiveness

Performance trade-offs from specialization



More sharing = Higher throughput



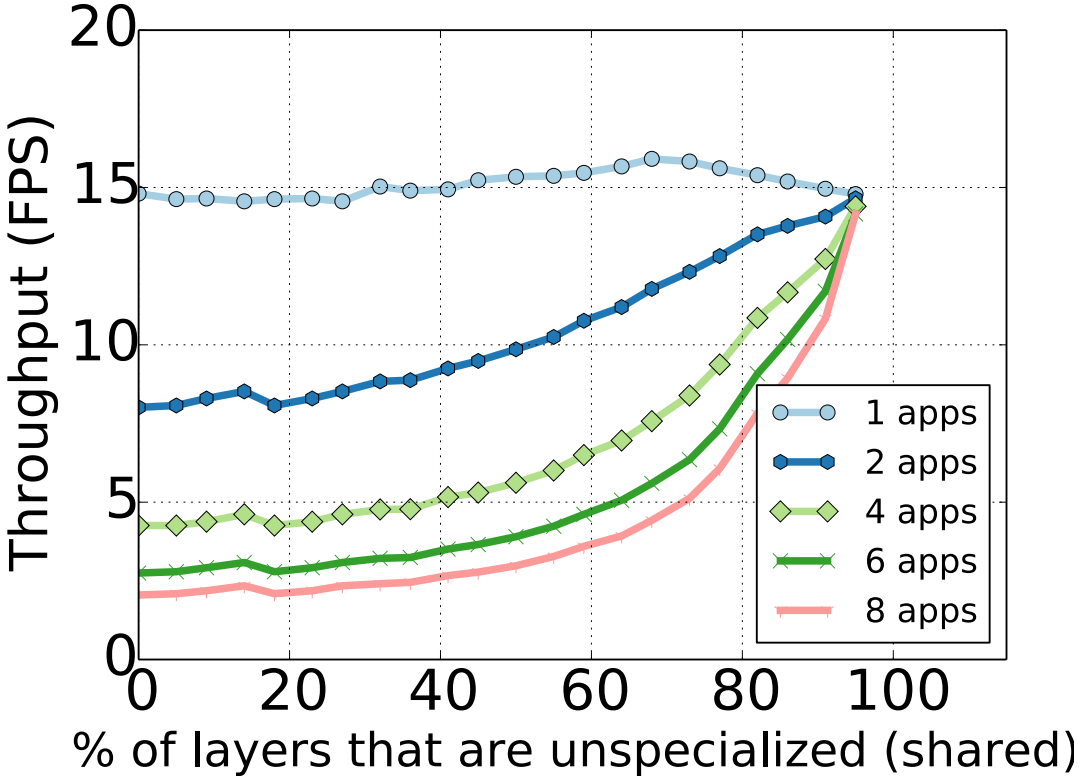
More specialization = Higher per-frame acc.

Experimental setup

- Run concurrent image classification pipelines
  - Performed with InceptionV3 and MobileNets-224 DNNs
- Hardware: Intel NUC
- Video stream: 20FPS, 640x480

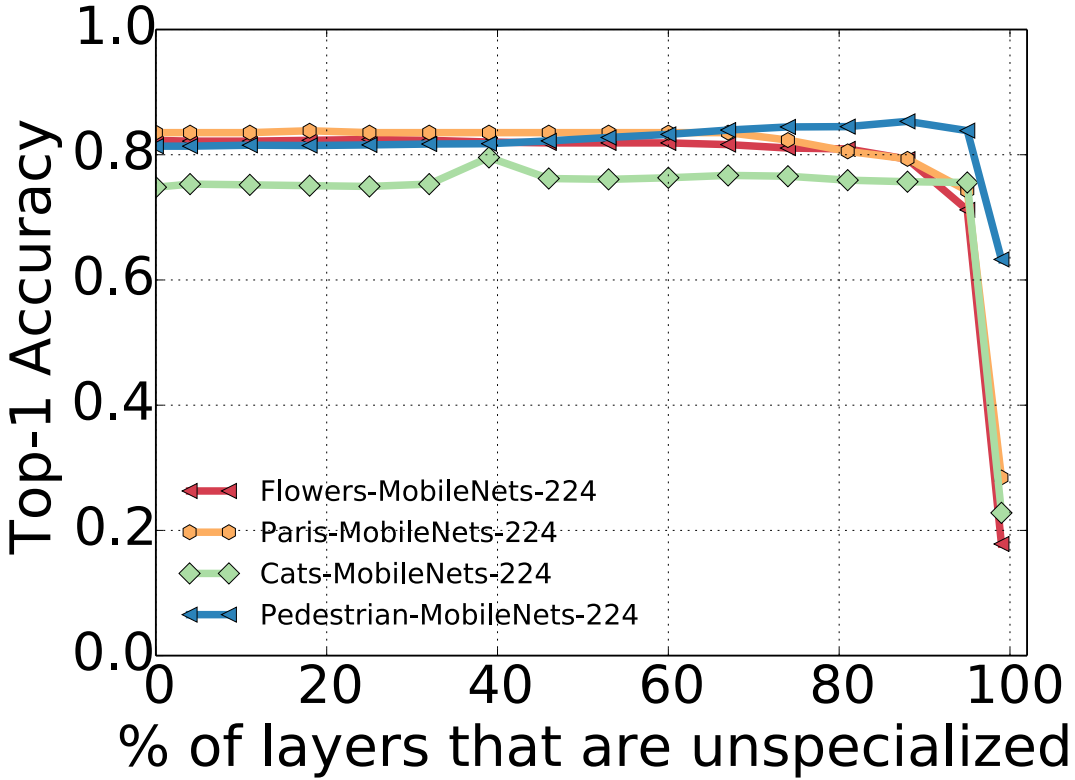
Less Specialized → Faster processing (FPS)

Running image classifiers with InceptionV3

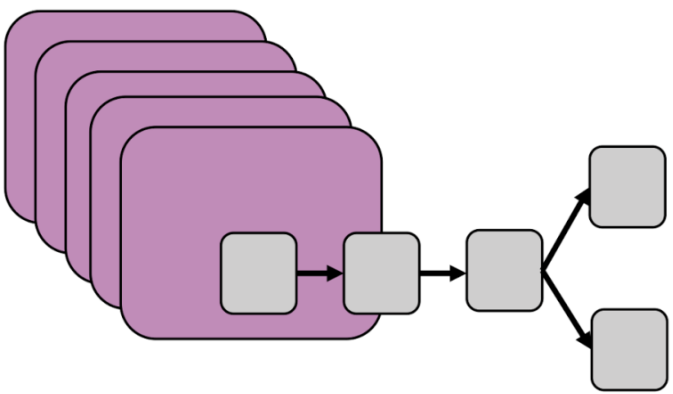


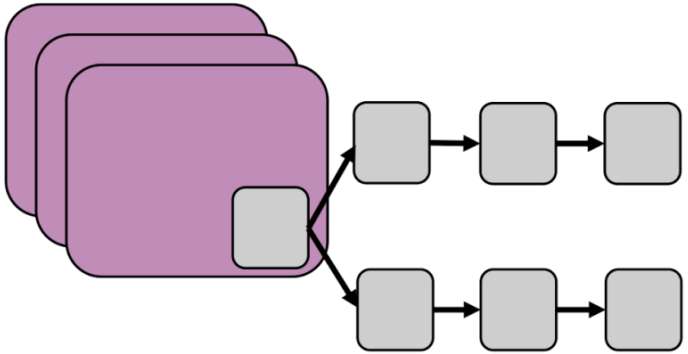
Less specialized → Lower per-frame acc.

Running image classifiers with MobileNets



Sample more or specialize more?





How to measure application performance?

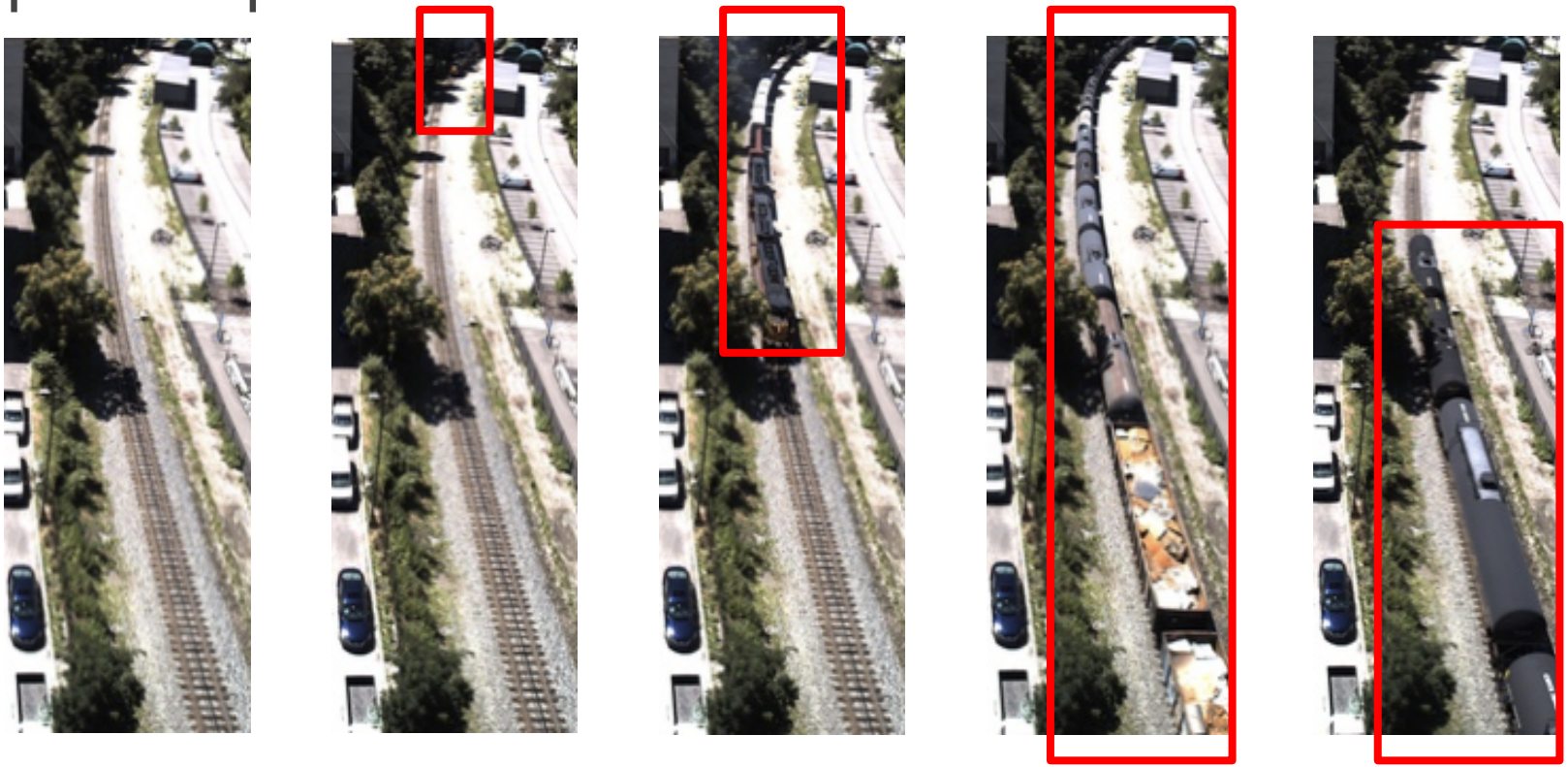
App example: Event detection

Goal: Maximize Event-F1 score

- Event-Recall: How many relevant events are detected?
- Event-Precision: How many detected events are relevant?
- Event-F1 score: Harmonic mean between precision and recall

Effect of accuracy and FPS on F1-score

App example: Event detection

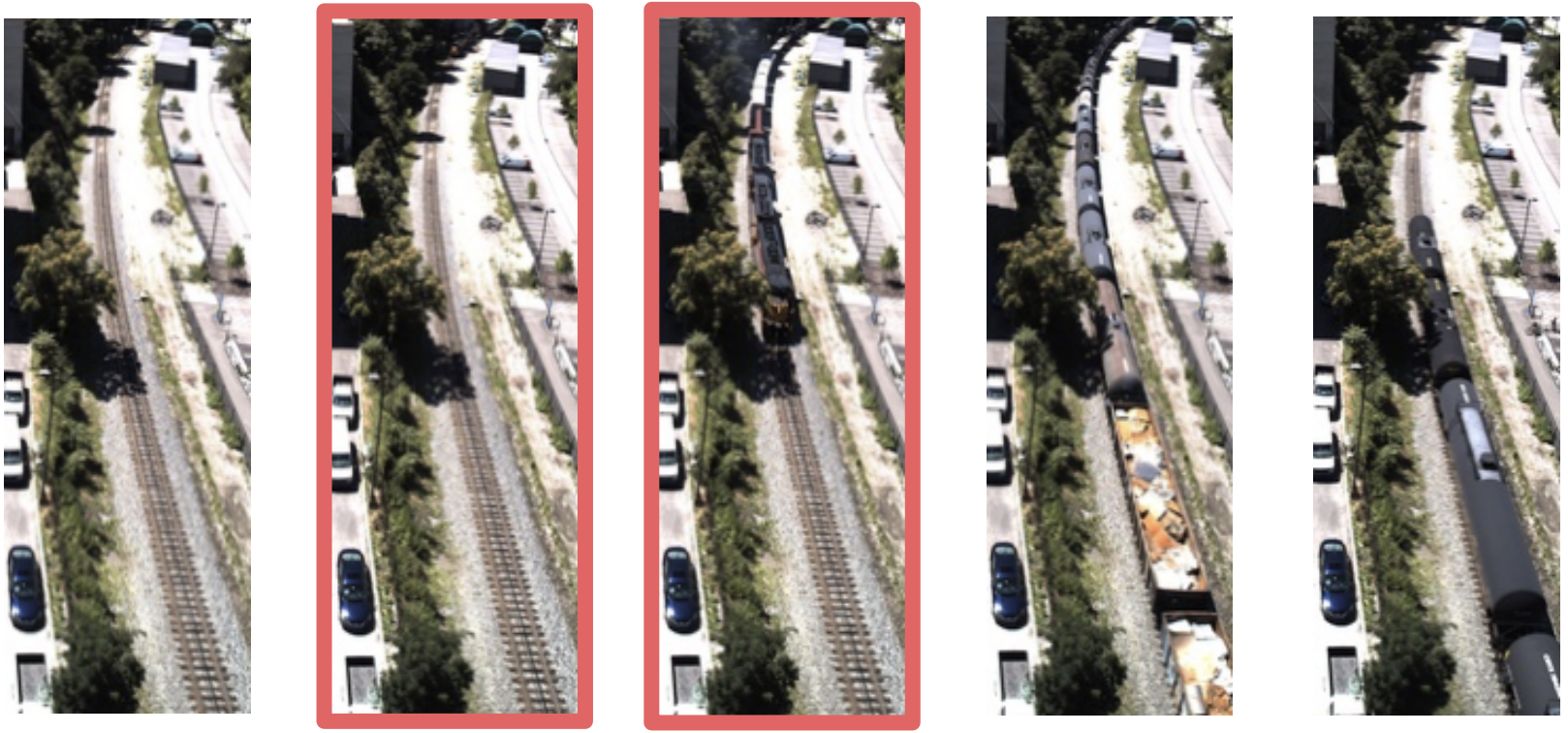


App example: Event detection

App example: Event detection

App example: Event detection

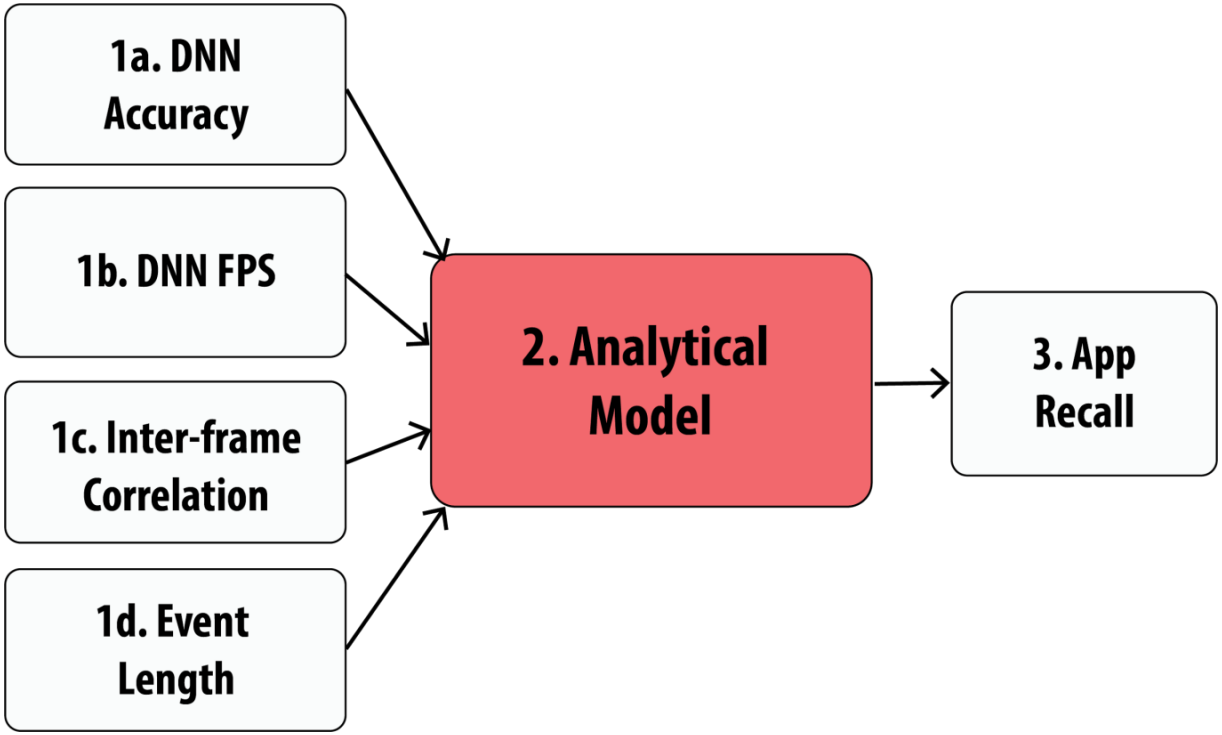
App example: Event detection



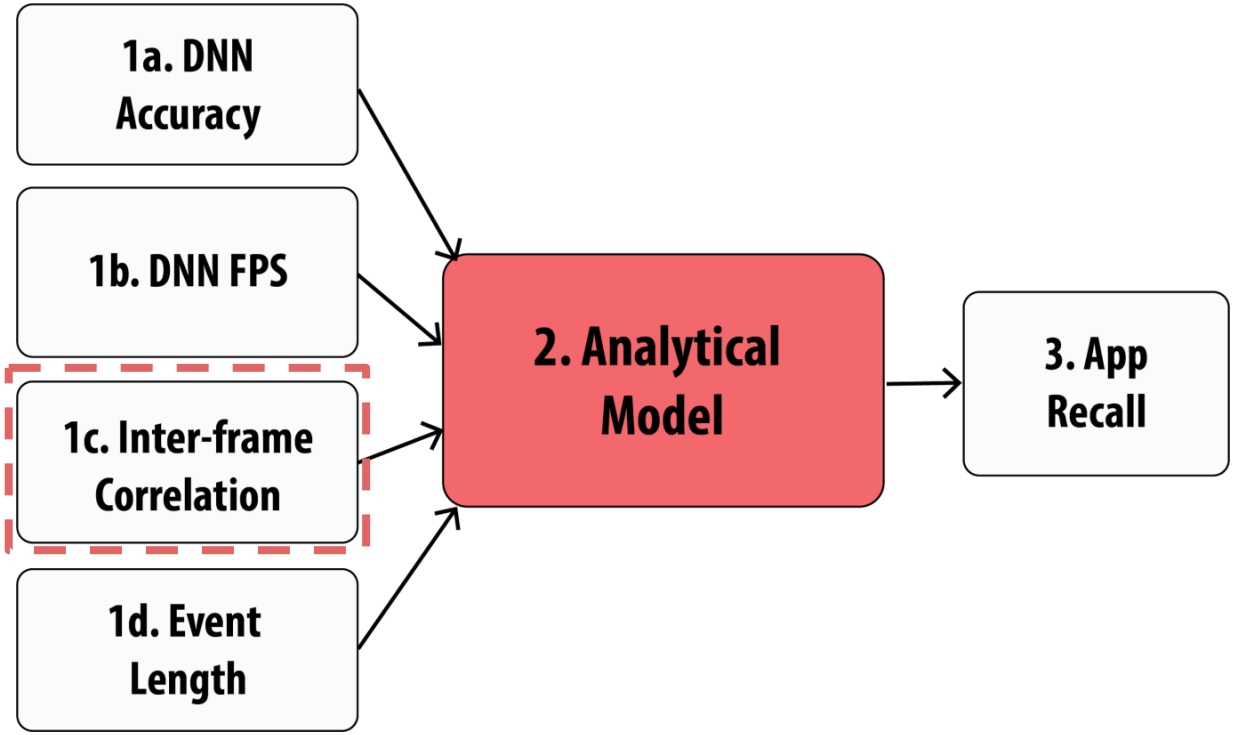
Challenge of searching the state-space

- Requires joint optimization between applications
- Combinatorially large state space
- Analytically model the function between FPS, accuracy and F1-score

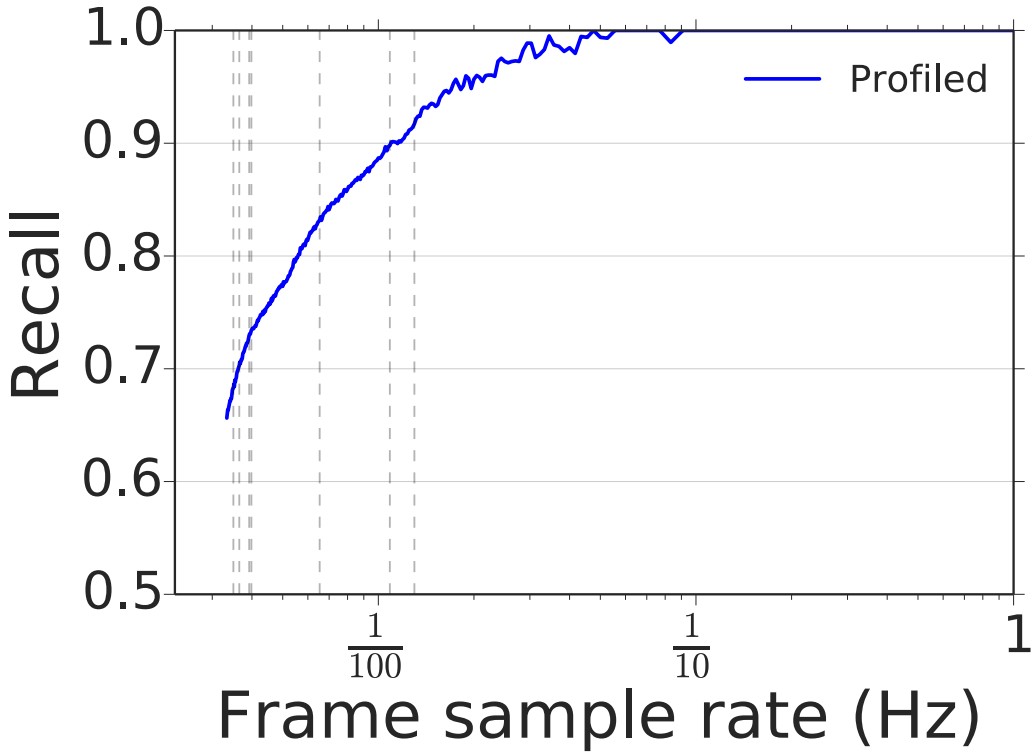
Analytical model for application quality



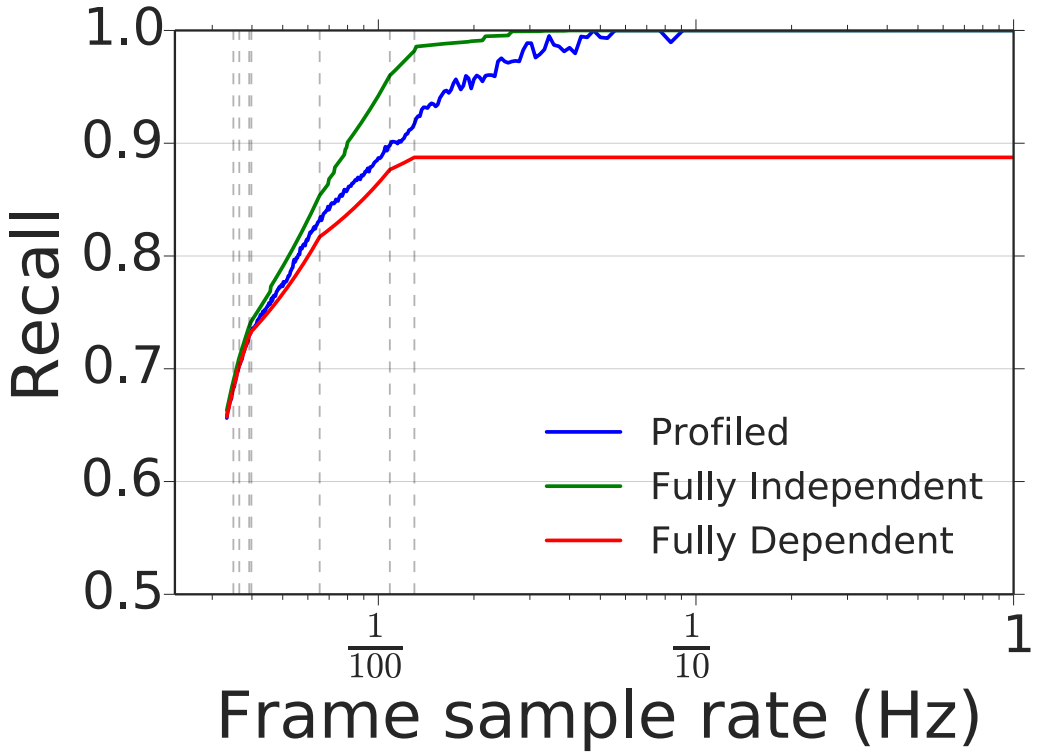
Analytical model for application quality



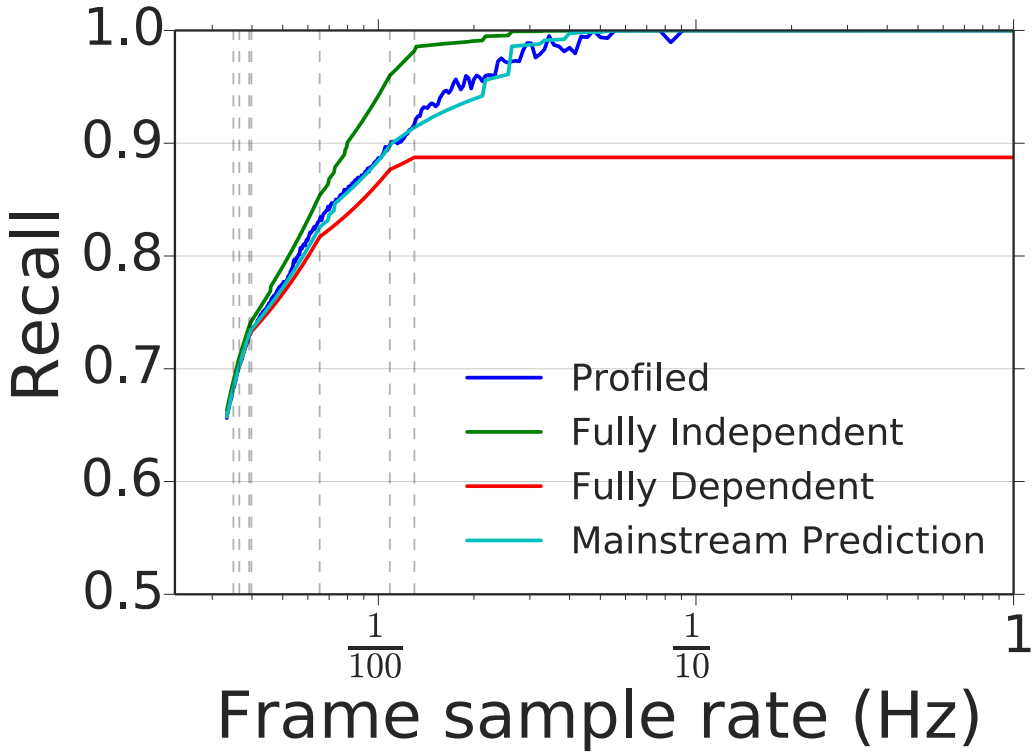
Estimating application quality



Estimating application quality

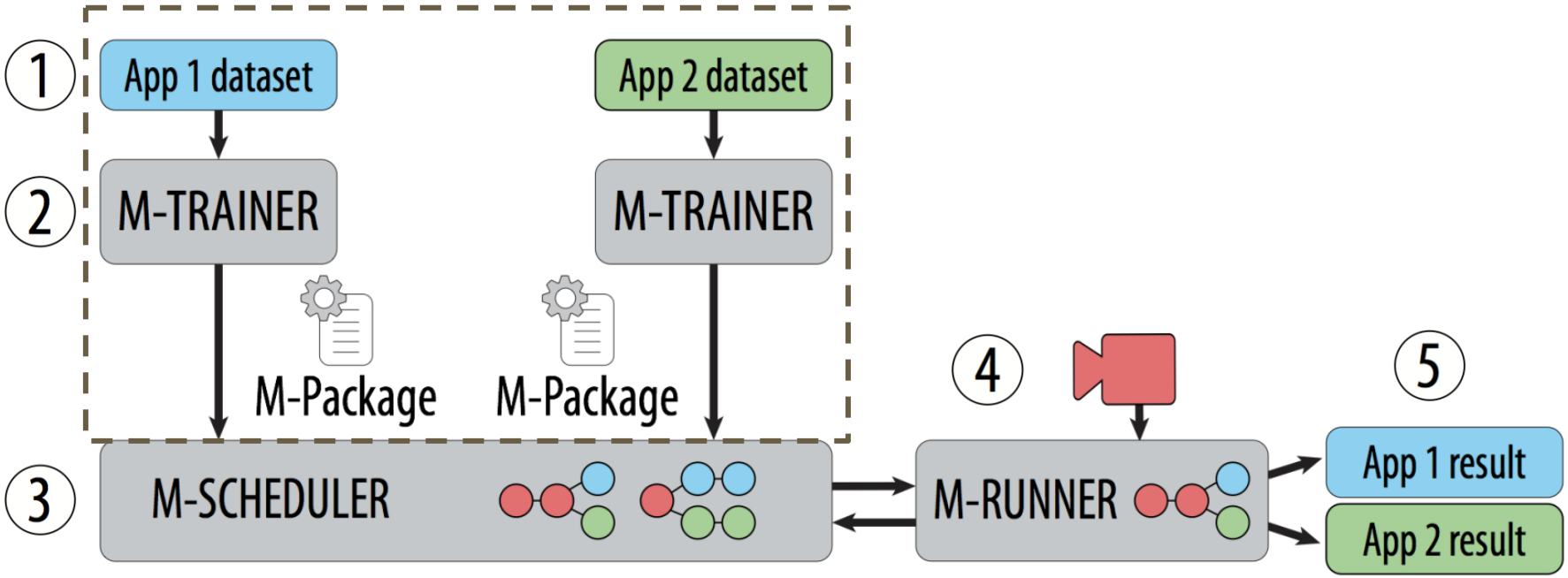


Estimating application quality

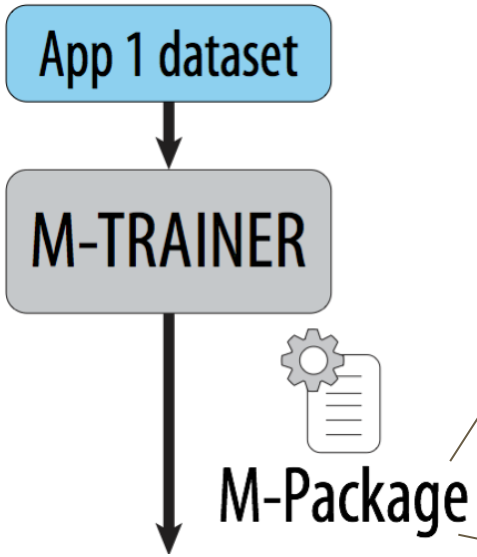


Mainstream’s architecture

Mainstream architecture



Decentralized, sharing-aware, training

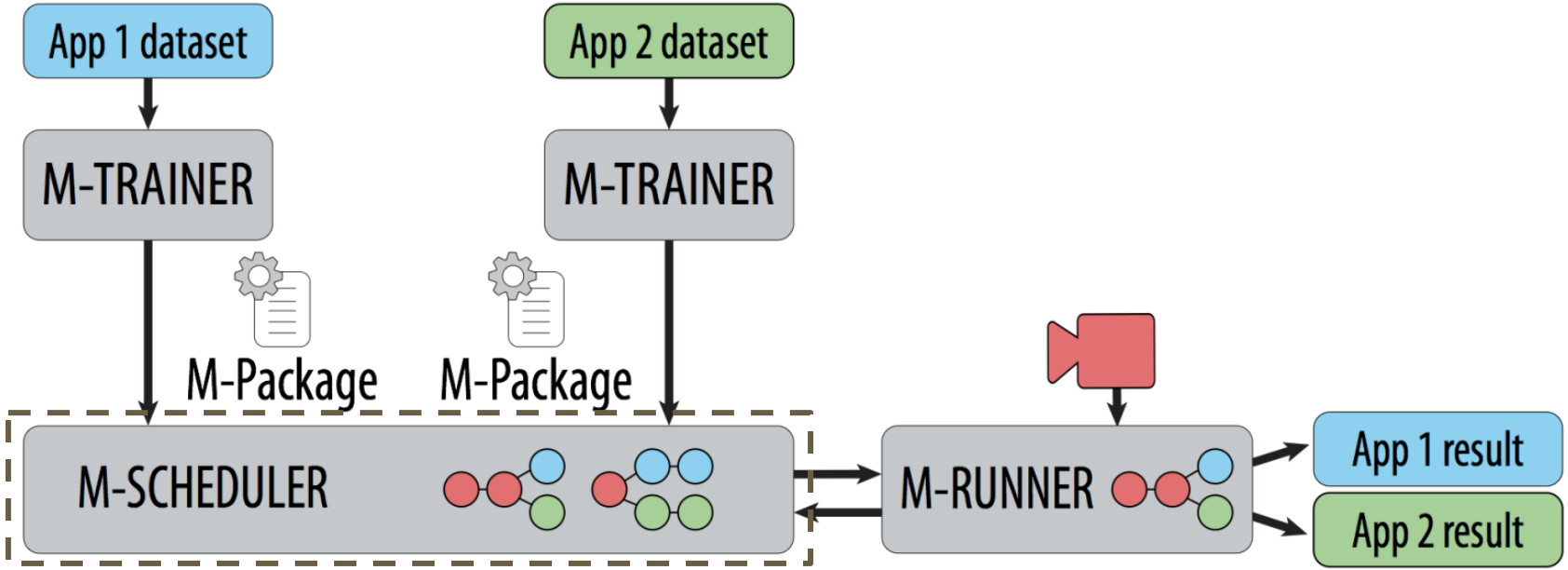


% specialized	Top1 Acc.	Model

Minimum event length = 50ms
Inter-frame correlation = 0.8

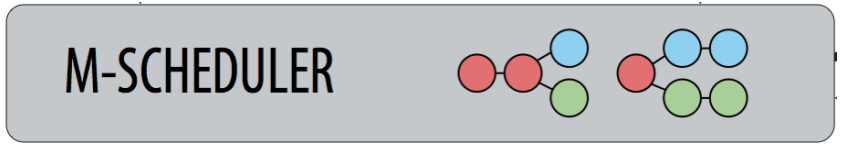
M-TRAINER

Mainstream architecture



Dynamic, sharing-aware, scheduling

- Input: M-Package
  - Model Sets, Min event length, Inter-frame correlation
- Output: Number of task-specific layers and frame-rate (per application)
- Maximize: F1-score (across applications)
- Approach: Greedily choose moves with maximum cost-benefit ratio
  - Benefit: from analytical model
  - Cost: Relative latency-based cost
- Use X-voting to improve precision

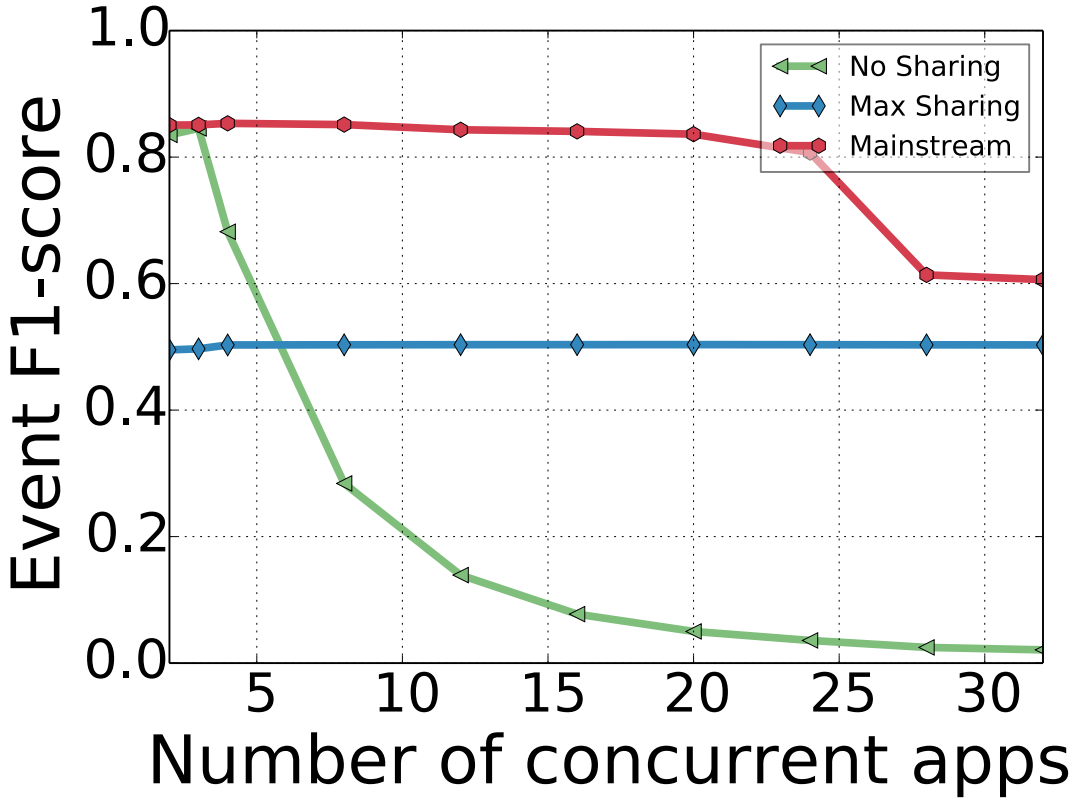


Mainstream’s evaluation

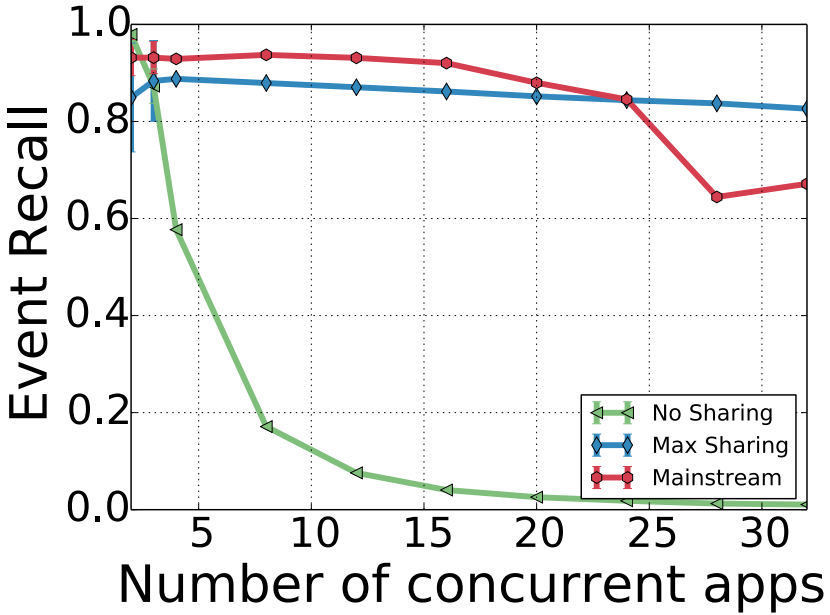
Experimental setup

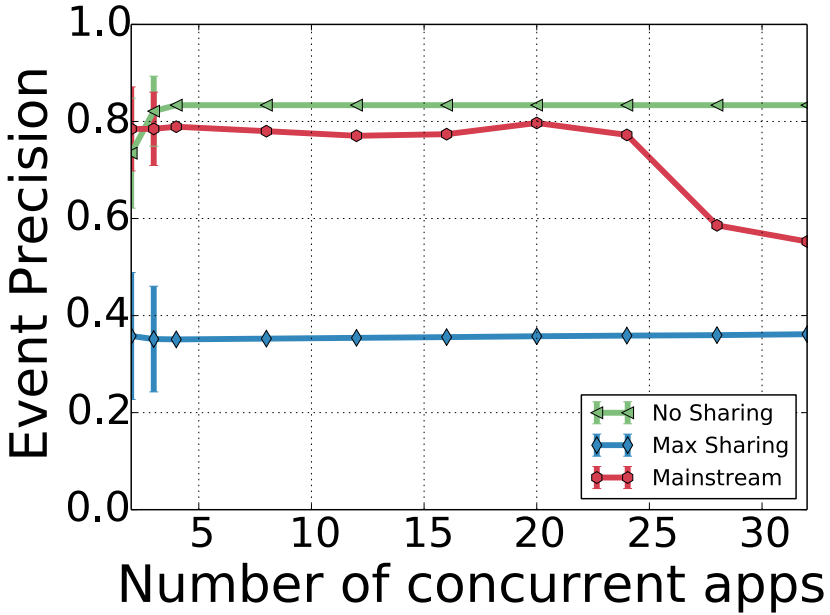
- Run concurrent image classification pipelines
  - Performed MobileNets-224 DNNs
- Applications: Pedestrians, Cars, Flowers, Cats
- Hardware: Intel NUC
- Video stream: 20FPS, 640x480

Mainstream improves F1-score up to 28X



Mainstream balances recall and precision





Mainstream Conclusion

- Enables efficient processing of a set of dynamic applications
  - By sharing DNN computation
- Determines how much of each network to specialize at runtime
- Uses analytical model to perform joint optimization
- Up to 28X improvement in F1-score, compared to No Sharing
- Up to 71% improvement in F1-score, compared to Max Sharing

X-voting must also be tuned dynamically