Lecture 10:
Specialization for Efficient Inference

Visual Computing Systems
Stanford CS348K, Spring 2020
Video processing applications
Thought experiment

Imagine we wanted to detect people/cars/bikes in a video stream
Thought experiment

Imagine we wanted to detect people/cars/bikes in a video stream
Interest in processing video efficiently

- Benefits to datacenter applications:
  - Lower cost/frame enables processing of more streams (e.g., thousands of webcams)

- Benefits to edge devices:
  - Cheaper per frame costs, real-time performance on cheaper/lower energy computing hardware
  - Lower latency per frame
    - Example: automated breaking systems target ~40ms sense to brake
Trick 0: video stream subsampling

- Spatial downsampling: run detector on low-resolution image
- Temporal subsampling: run detector at low frame rate
Trick 1: exploit temporal coherence
Temporal differencing

- Idea: use labels from empty frame image if similar to background image

(a) empty frame  (b) frame with a car  (c) subtracted frames

- Idea: use same result as previous frame if two frames are sufficiently similar
  - How to define sufficiently similar? (thresholds?)
  - Differences in feature space more robust than over pixels
Tracking
Evaluate expensive detector sparsely in time (e.g., every 1/2 second), then use more efficient tracking algorithm to update annotations over sequence of frames
Tracking
Evaluate expensive detector sparsely in time (e.g., every 1/2 second), then use more efficient tracking algorithm to update annotations over sequence of frames
Leveraging motion in the network

Given features (or final segmentation result from prior frame) use flow between prior and current frame to advect features (or segmentation) to new frame.

In other words: it’s easier to produce the result for the current frame if you have the result from the prior frame.

[Zhu CVPR 2017]
Leveraging motion in the network

In practice: despite “intellectual appeal” of advecting features, paper results show advecting segmentation is as good as advecting features.
Trick 3: specialize to content
Model specialization

- Common principle in DNN design/training is to learn most general model (via large datasets, regularization, etc.) to perform well across all instances of a task

- But many cameras see a very specific distribution of images
  - Only certain types of object classes
  - Always from the same/similar viewpoint
  - Objects appear in same regions of screen

- Specialization has been a major theme in this class w.r.t hardware design. Now we wish to specialize models to the contents of a video stream
  - “A model can be much simpler if it only needs to work for a single camera”
Model distillation

- Accurate, but expensive, model: trained on full training set
  - “The teacher”

- Smaller model (cheaper), trained to mimic the output of the teacher
  - “The student”
**Noscope**

- Apply model distillation, but constrain training set to a specific video feed: Given an expensive network that performs a specified detection* task well on a wide range of videos, distill a highly optimized implementation for **this video stream**.

- Example: binary classification (car/no car) for a single traffic camera video stream.

![Image](https://example.com/image.png)

(a) empty frame  (b) frame with a car  (c) subtracted frames

* Noscope actually performs a simpler classification task on a pre-cropped region of the viewport (not detection, which involves object location)
Three Noscope optimizations

- **Statically specialize model to video feed**
  - Teacher network: Yolo object detection network
  - Student network: compact specialized network (2-4 conv layers)
  - Low cost student “learns” to mimic the teacher

- **Dynamic: utilize frame-to-frame difference detectors with learned thresholds**
  - “Same as background”, “same as previous frame”
  - Learn thresholds for how often to check for differences (in frames), and what the magnitude of a meaningful difference is

- **Dynamic: cascades**
  - Run cheap specialized model (student) on frame first, then run teacher model if student does not make a confident prediction
Noscope results *

Noscope actually performs a simpler classification task on a pre-cropped region of the viewport (not detection, which involves object location)
Another take on specialization
Follow on from last time

- Alternative (more traditional) specialization strategy: choose among set of pretrained models to find cheapest (sufficiently accurate) model for the job
  - “Knobs” to configure:
    - Input image resolution
    - Input image frame rate
    - DNN to use (Resnet101, Resnet50, Inception, MobileNet, etc.)
    - Thresholds on frame-to-frame difference detectors, etc.
Simple example

Appropriate frame-rate sampling depends on whether cars are moving
Challenge of distribution shift

- If distribution in video stream is non-stationary, cheap model determined via up-front profiling looses accuracy as contents of video change
- Implication: choice of specialized model needs to be periodically changed

Results from object detection task on traffic camera video

Challenge: cost of profiling to adaptively determine which model to run eliminates potential benefits of model specialization
Reducing the cost of profiling

- The cost of profiling is running the candidate models at points in search space (profiling different values for all knobs)

- Idea 1: set of most-likely-to-be-good models changes slowly over time

- Idea 2: visually similar streams are likely to have similar set of most-likely-to-be-good candidate models
Employing idea 1

- Assume model can change every video “segment” (e.g., 4 seconds)
- Profile all C model configurations for time segment 1
  - Retain top-K configurations
- Profile only top-K configurations in future segments
- Reset after N segments

Let $S$ be number of segments before reset ($\sim 4$)
Let $K$ be size of candidate set ($K \ll C$)

profiling cost $= C + (N-1) \times K \ll C \times N$

Assumption: bad model configurations tend to remain bad for longer periods of time
Employing idea 2

- Say there are many video cameras throughout a city
- Cluster cameras by how visually similar their video streams are
- Only one camera per cluster needs to perform full profiling of the configuration to identify top-K candidate set
  - Other cameras just perform top-K profiling
Intelligent profiling makes adaptive specialization profitable

Across dataset of multiple streetlight cameras, when keeping accuracy similar, adaptive profiling over the 150 second test video yields 2-3X speedup compared to profiling once up front.

But really the problem with profiling once is that accuracy is highly variable (see accuracy variance of blue crosses)
Summary

- An increasing number of cameras across the world will be capturing near continuous video

- Many applications will seek to extract value from these data streams
  - Implications for efficiency of cities (transportation, infrastructure monitoring), brick-and-mortar commerce, security, health-care, robotics, human-robot interactions, autonomous vehicles

- Need significant efficiency gains to process this worldwide visual signal
  - We’ve already talked about hardware specialization
  - Today’s focus: specialization of model to video stream or scene context
Discussion: privacy and ethics in a world with always-on video

Amazon’s Rekognition messes up, matches 28 lawmakers to mugshots

ACLU: "And running the entire test cost us $12.33—less than a large pizza."

CYRUS FARIVAR - 7/25/2018, 5:00 AM