DATA ANALYTICS USING DEEP LEARNING

GT 8803 // FALL 2018 // JENNIFER MA

LECTURE #13: FOCUS: QUERYING LARGE VIDEO DATASETS WITH LOW LATENCY AND LOW COST

CREATING THE NEXT®

TODAY'S PAPER

- Focus: Querying Large Video Datasets with Low Latency and Low Cost

TODAY'S AGENDA

- Problem Overview
- Key Idea
- Technical Details
- Experiments
- Discussion

PROBLEM OVERVIEW

- Querying camera recordings
- Traffic intersections, retail stores, offices, etc.
- Slow and costly

PROBLEM OVERVIEW

- Querying a month-long video would requires 280 GPU hours and $250
- To run the query in 1 minute requires 10000s of GPUs
- Traffic jurisdictions and retails may only have 10s or 100s

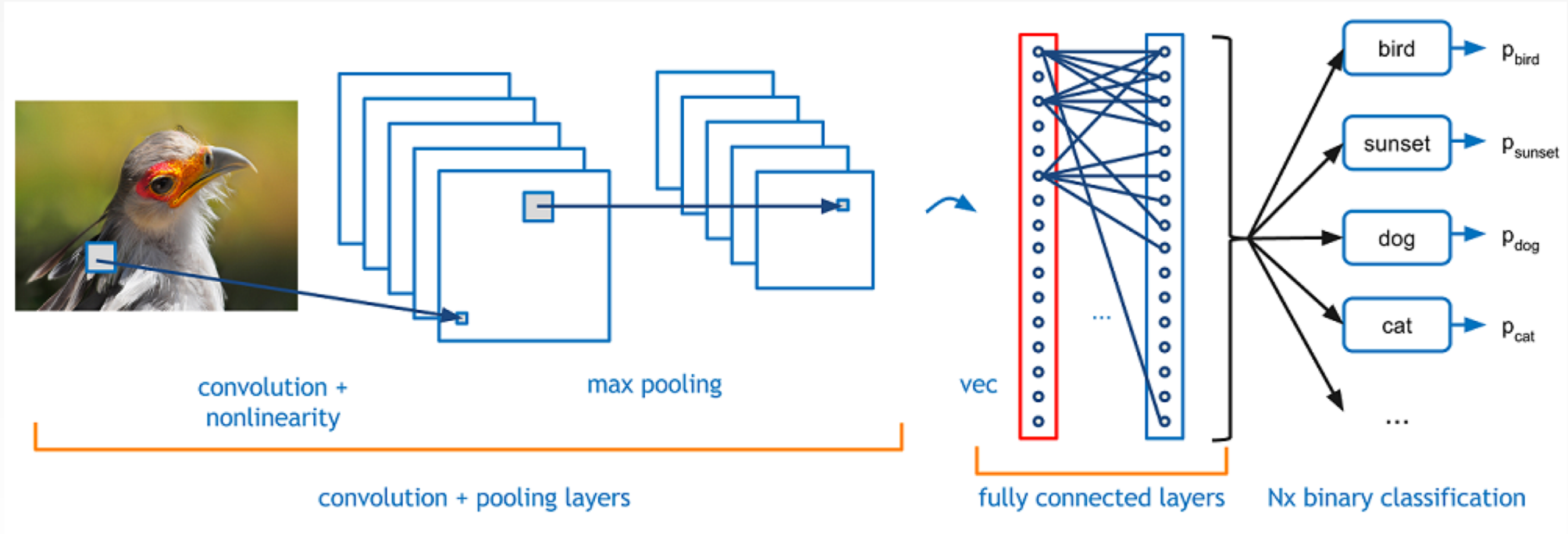
KEY IDEAS

- Classify before query time
- Smaller and specialized CNN's
  - Fewer layers
  - Take in smaller images
  - Specialized: For each video domain, train the CNN's only on the classes that appear in those videos

  - Video domains: traffic cameras, surveillance cameras, and news channels

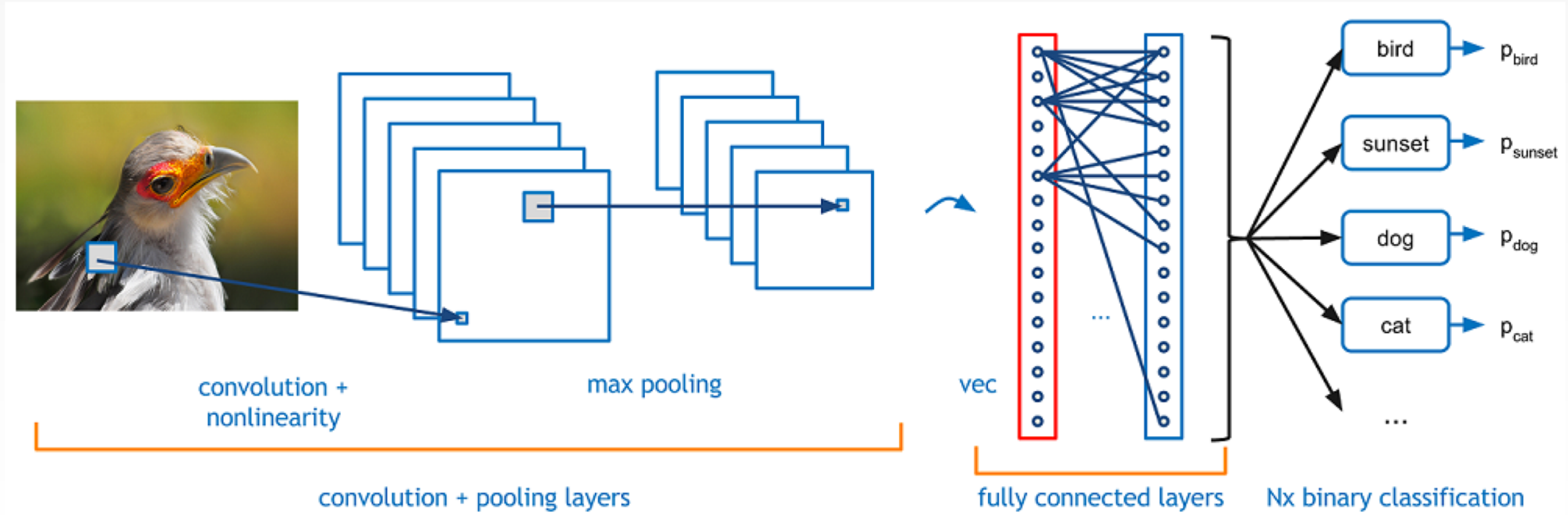
TECHNICAL DETAILS

- Convolutional neural networks (CNN's)



CONVOLUTIONAL NEURAL NETWORKS

- Types of Layers:
  - Convolutional and Rectification Layers
  - Pooling Layers
  - Fully-Connected Layers



CONVOLUTIONAL NEURAL NETWORKS

- Slow and costly
- ResNet152
  - 152 layers
  - Won ImageNet competition of 2015
  - Processed only 77 images/sec with a GPU

TECHNICAL DETAILS

- Compressed CNN's
  - Remove layers
  - Matrix pruning
  - Other

  - Results: smaller cnn's, so faster to train, but lower accuracy

TECHNICAL DETAILS

- Specialized CNN's
  - Smaller set of classes
  - Higher accuracy

TECHNICAL DETAILS

- Recall – percentage of correct frames returned
- Precision – percentage of frames classified correctly
- Predict top-k classes to increase recall
- Use full CNN on objects to increase precision

CHARACTERISTICS OF REAL-WORLD VIDEOS

- Many frames contain no objects
  - 0.01% on average
  - 16% - 43% for the most frequent object classes

- Optimization:
  - Filter these out, to speed up training time

CHARACTERISTICS OF REAL-WORLD VIDEOS

- Each video domain has only a subset of object classes
  - In less busy videos, only 22-33% of the 1000 object classes appeared.
  - In busy videos, only 50-69% of them appear.

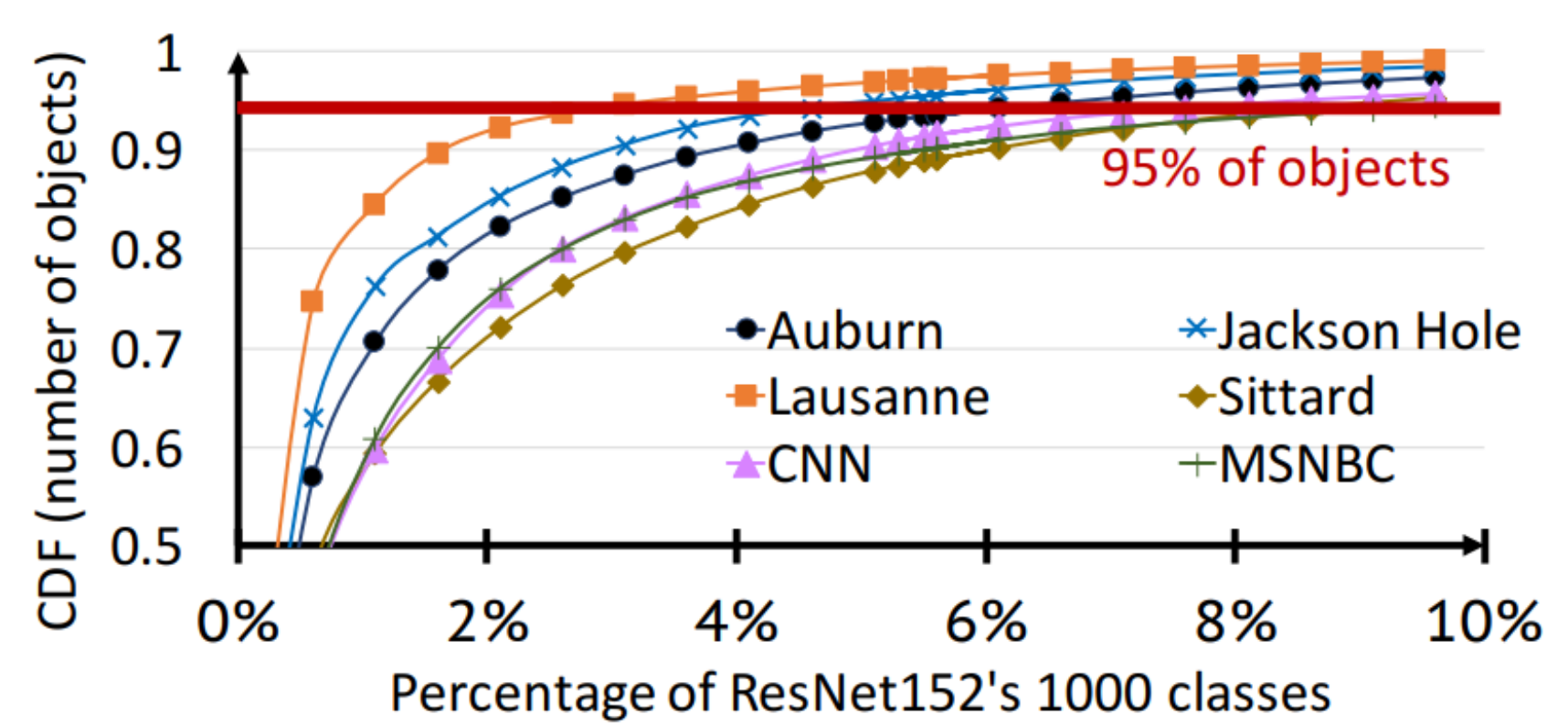
- Optimization:
  - Train specialized CNN's, for higher accuracy

CHARACTERISTICS OF REAL-WORLD VIDEOS

- Each video domain has only a subset of object classes
  - Little overlap between objects in different video domains
    - Different specialized cnn's for each domain
  - Interesting: 3-10% of the most frequent objects cover 95% of appearances

CHARACTERISTICS OF REAL-WORLD VIDEOS

- The 10% most frequent classes account for 95% of object appearances



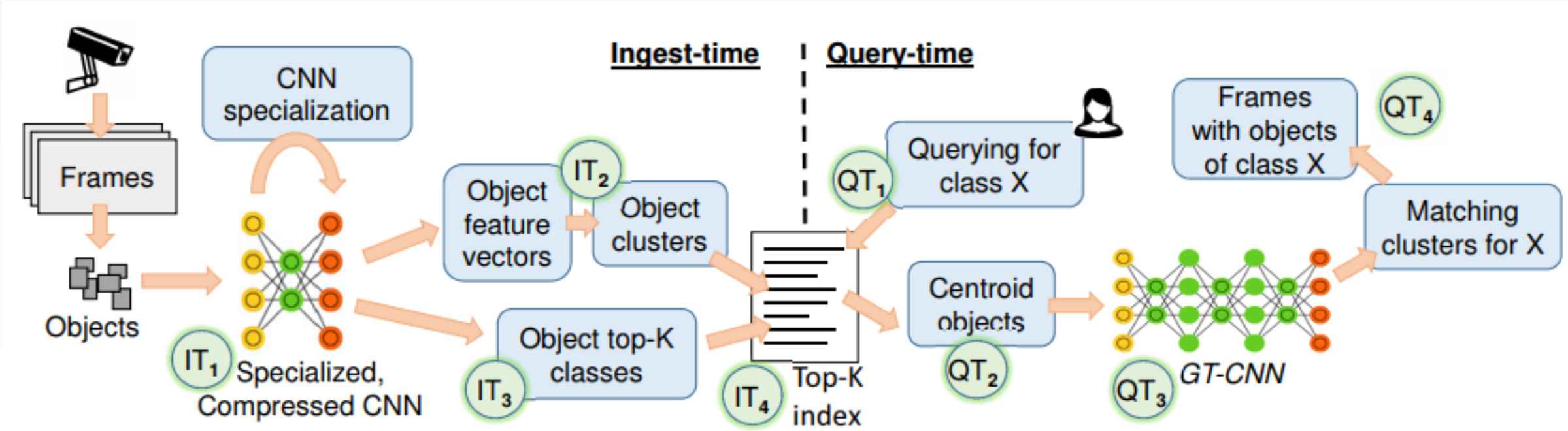
CHARACTERISTICS OF REAL-WORLD VIDEOS

- Many objects appear in several frames
  - Several seconds, several frames

- Optimization:
  - Extract feature vectors for the objects, cluster them, get the centroid, and classify only this one with the cnn

OVERVIEW OF FOCUS

- Query-time – user queries, Focus returns frames
- Ingest-time – Focus runs during recording, creating index from object classes to frame clusters



OVERVIEW OF FOCUS

- Query-time –
  - 1. Get class from query
  - 2. Pass class to index to get the clusters
  - 3. Use ground-truth CNN on each cluster to get predicted class
  - 4. Return frames matching class asked for

OVERVIEW OF FOCUS

- Ingest-time –
  - 1. For each frame, for each object, extract its feature vector
  - 2. Cluster these
  - 3. Assign the top k most likely classes to each cluster
  - 4. Put the cluster in index for each object class

TECHNIQUES: CHEAP INGESTION

- Classify objects at ingest-time to reduce query latency
- Use cheap cnn's to reduce ingest cost
- Take ground truth cnn and apply compression
- Produce set of cheap cnn's to pick from

TECHNIQUES: TOP-K INGEST INDEX

- Cheap cnn's have lower accuracies
- To keep recall high, pick top K classes
- Higher K -> lower precision, so use ground truth cnn

TECHNIQUES: REDUNDANCY ELIMINATION

- To reduce query latency, use GT-CNN to classify object class once
- Assign the prediction to all similar object appearances
- Identify same objects by clustering their feature vectors
- Assign clusters top-k classes, index clusters, and at query time, run GT-CNN on all clusters, return ones matching object class in question

TECHNIQUES: CLUSTERING HEURISTIC

- O(Mn), M constant, n = number of objects
- Single pass, does not need number of clusters as parameter
- Algorithm:
  - For each new object, assign to closest cluster
  - If no closest cluster within T distance, assign it to new cluster
  - If # of clusters > M, put smallest in index

TECHNIQUES: CLUSTERING AT INGEST VS QUERY TIME

- Clustering at ingest time:
  - Store all feature vectors
- Query time:
  - Store only cluster centroids
  - Faster

TECHNIQUES: PIXEL DIFFERENCING OF OBJECTS

- Reduce ingest cost
- For objects with similar pixel values, assign to same cluster instead of rerunning CNN

SPECIALIZED CNNS

- Higher accuracy due to
  - Videos have only a few object classes
  - The objects look similar -> less image features needed -> simpler model -> more accuracy
- 10x Faster because
  - 1/3 less layers
  - Input image 4x smaller
- Higher accuracy -> smaller K -> lower query latency

MODEL RETRAINING

- Keep models up to date
- Resample frames regularly
- Use ground truth CNN to get new class distribution
- Select new classes to train specialized models on
- Power law

THE OTHER CLASSES

- Classes not selected for specialized are grouped into one class: “Other”
- Smaller Ls leads to bigger “Other”

PARAMETERS

- K
  - Number of top classes to assign to each cluster
- L_s
  - Number of classes to train specialized model on
- CheapCNN
  - The specialized ingest-time cheap CNN
- T
  - The distance threshold for clustering objects

PARAMETER SELECTION

- Stage 1:
  - Choose CheapCNN, Ls, and K
  - Recall target
- Stage 2:
  - Choose T
  - Precision target

PARAMETER SELECTION

- Minimal sum of ingest and query costs

- Or:
  - Minimal ingest cost

- Or:
  - Minimal query cost

EXPERIMENTS: DATA

- 13 video streams
- Traffic cameras, surveillance cameras, and news channels
- 12 hours per video
  - Covers day and night time

EXPERIMENTS: BASELINE

- Ground truth:
  - classifications by state-of-the-art CNN, ResNet152
- Default accuracy targets:
  - 95% recall and 95% precision

Baselines:

- Ingest-all
  - classifies all objects at ingest time, and stores in index
- Query-all
  - classifies objects at query time

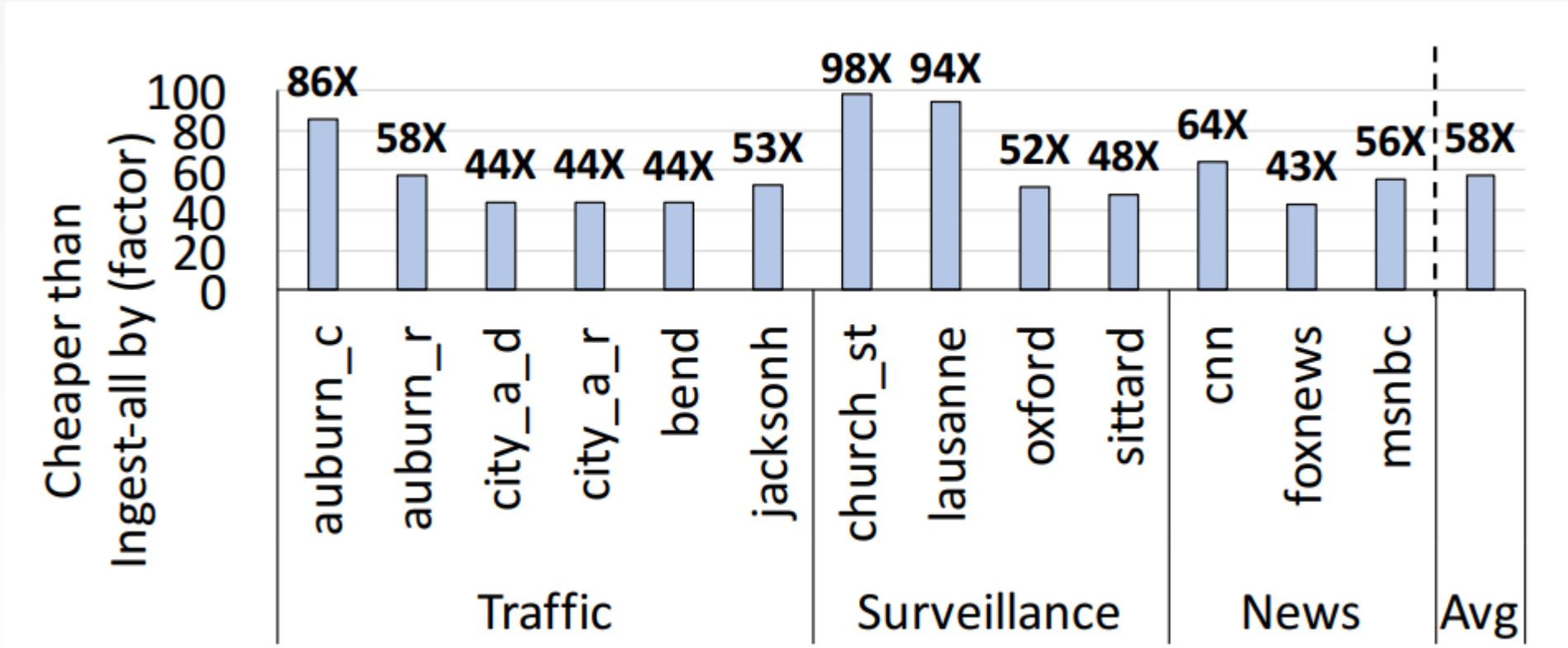
EXPERIMENTS: METRICS

1. Ingest cost
   - GPU time to process each video

2. Query latency
   - Time to query a specific object class
   - Per video, they average the latencies for dominant object classes.

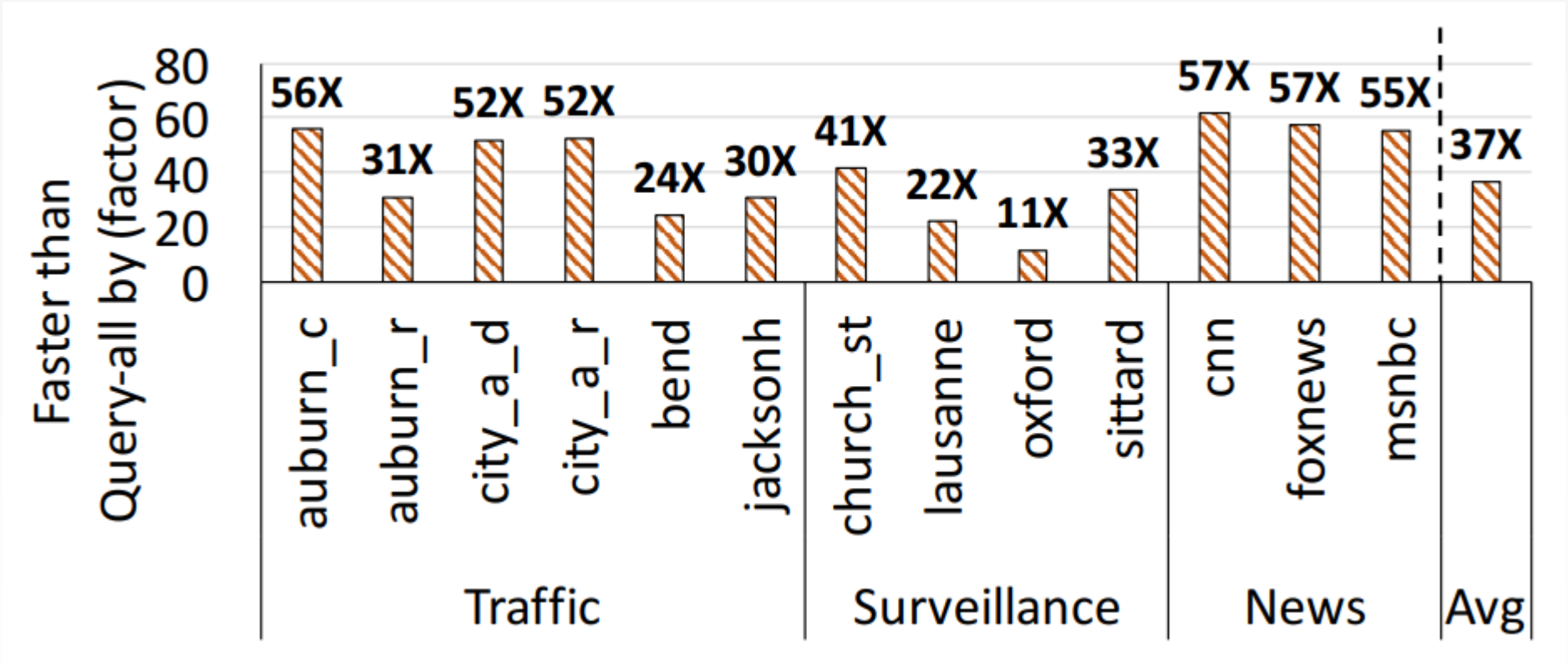
EXPERIMENTS: INGEST COST

- Speedup improvement compared to Ingest-all



EXPERIMENTS: QUERY LATENCY

- Speedup improvement compared to Query-all



EXPERIMENTS: QUERY LATENCY

- Average speedup: 37x
- With 10 GPU's, querying 24-hr video goes from 1 hr to < 2 min
- Cost goes from $250 to $4/month

EXPERIMENTS: QUERY LATENCY

- Query latencies improved for variety of different videos
  - busy intersections,
  - normal intersections or roads,
  - rotating cameras,
  - busy plazas,
  - a university street, and
  - different news channels.

EXPERIMENTS: EFFECT OF COMPONENTS

- Compressed model
- Compressed + Specialized model
- Compressed + Specialized model + Clustering

EXPERIMENTS: COMPRESSED MODEL

- Decreased both ingest and query costs
- Relatively minimally
- Fewer layers -> Lower accuracy
- Need to select more expensive model and larger K -> increases ingest and query times

EXPERIMENTS: COMPRESSED+SPECIALIZED

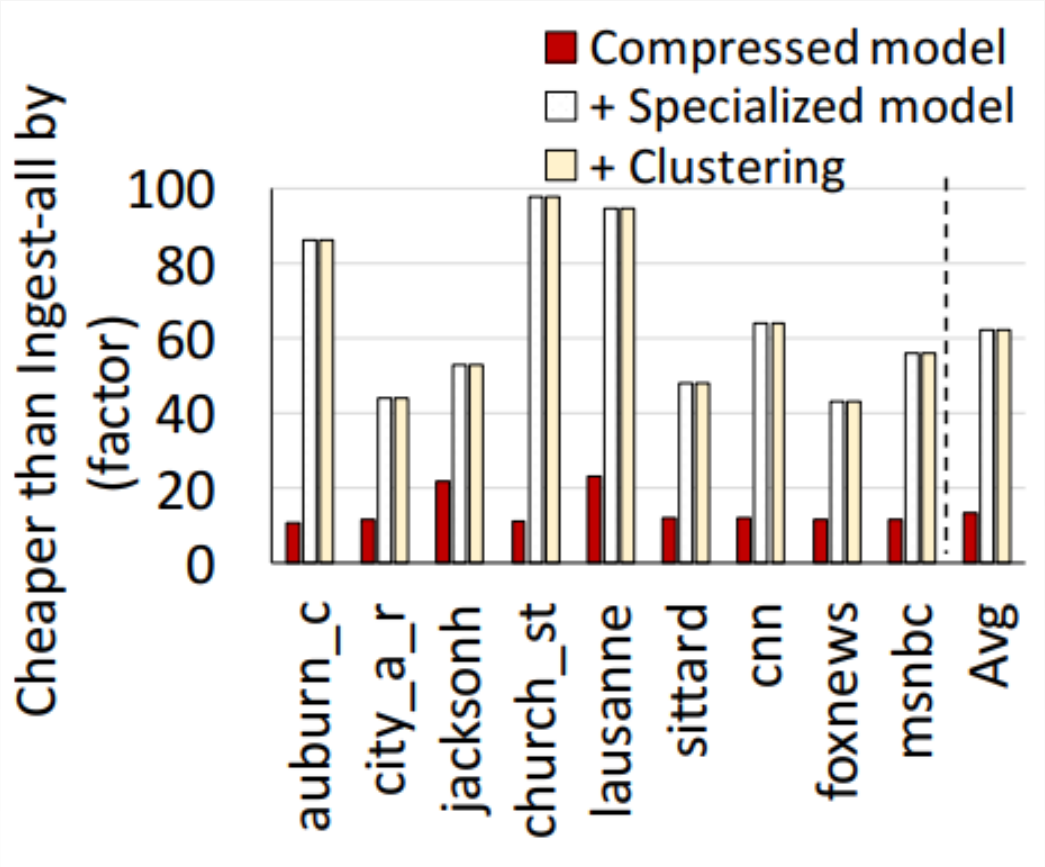
- Largely decreases costs
- Specializing increases accuracy
- Speeds up query latency by 5-25x
- Decreases ingest cost by 7-71x

EXPERIMENTS: +CLUSTERING

- Cluster feature vectors of objects at ingest time
- Reduces work at query time
- Lowered query latency by up to 56x
- Ran clustering on CPUs, and specialized model on GPUs

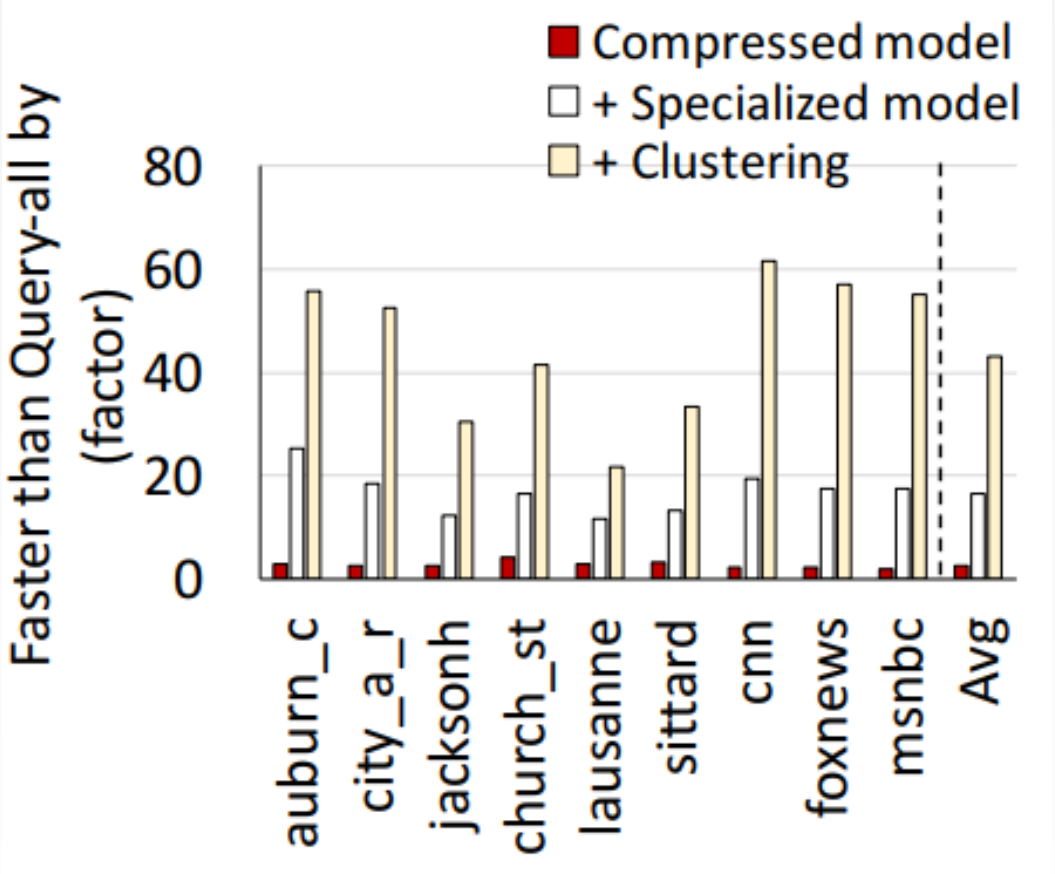
EXPERIMENTS: INGEST COST

- Adding specialized led to dramatic improvement
- Clustering did not increase ingest cost too much



EXPERIMENTS: QUERY LATENCY

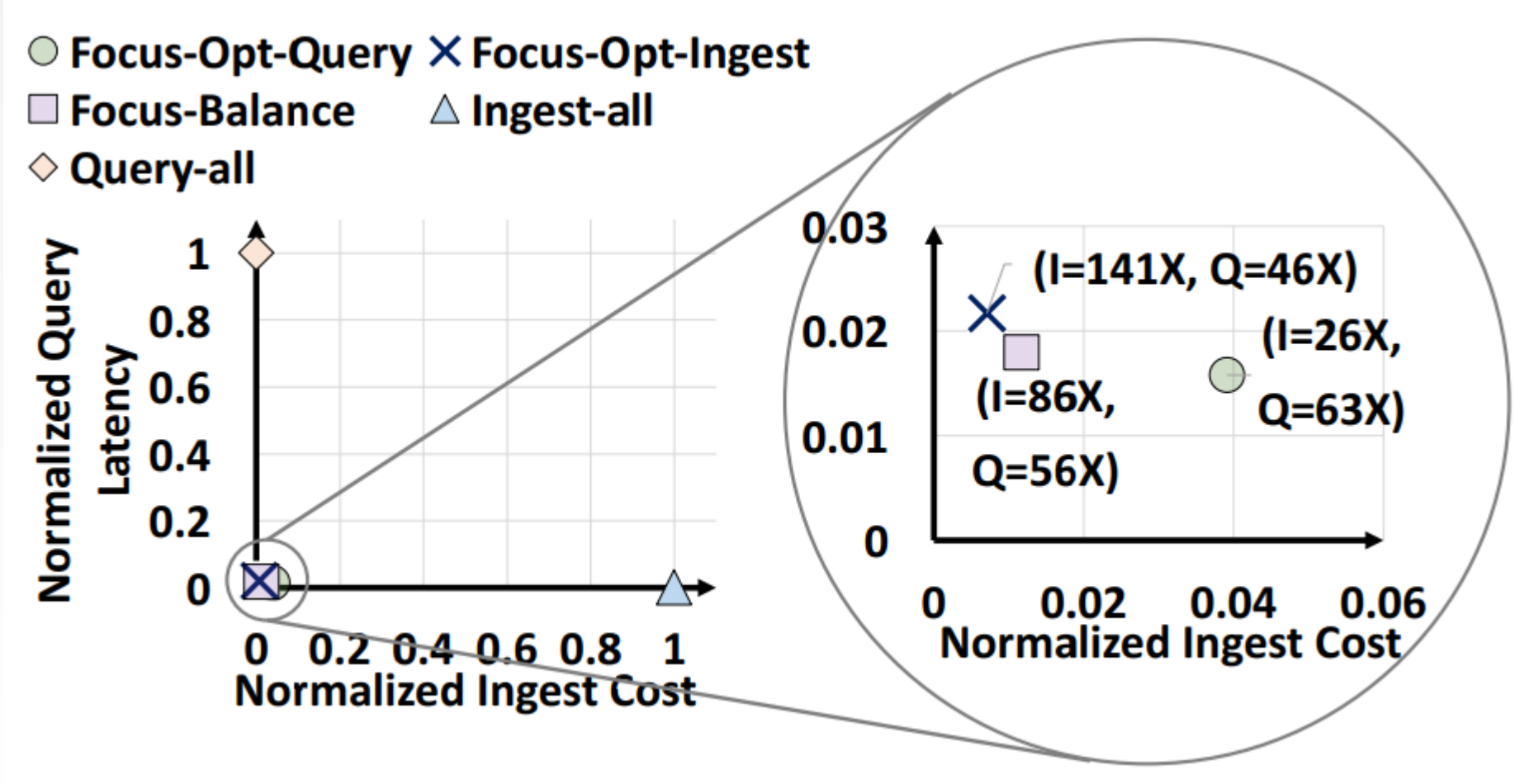
- Compressed model has minimal improvement compared to specialized
- Clustering greatly speeds up query processing



EXPERIMENTS: REVIEW OF OPTIONS

- Opt-Ingest
- Opt-Query
- Balanced

EXPERIMENTS: OPTIONS



EXPERIMENTS: OPTIONS

- Opt-ingest
  - 141x faster ingest
  - 46x faster query
- Opt-query
  - 63x faster query
  - 26x faster ingest

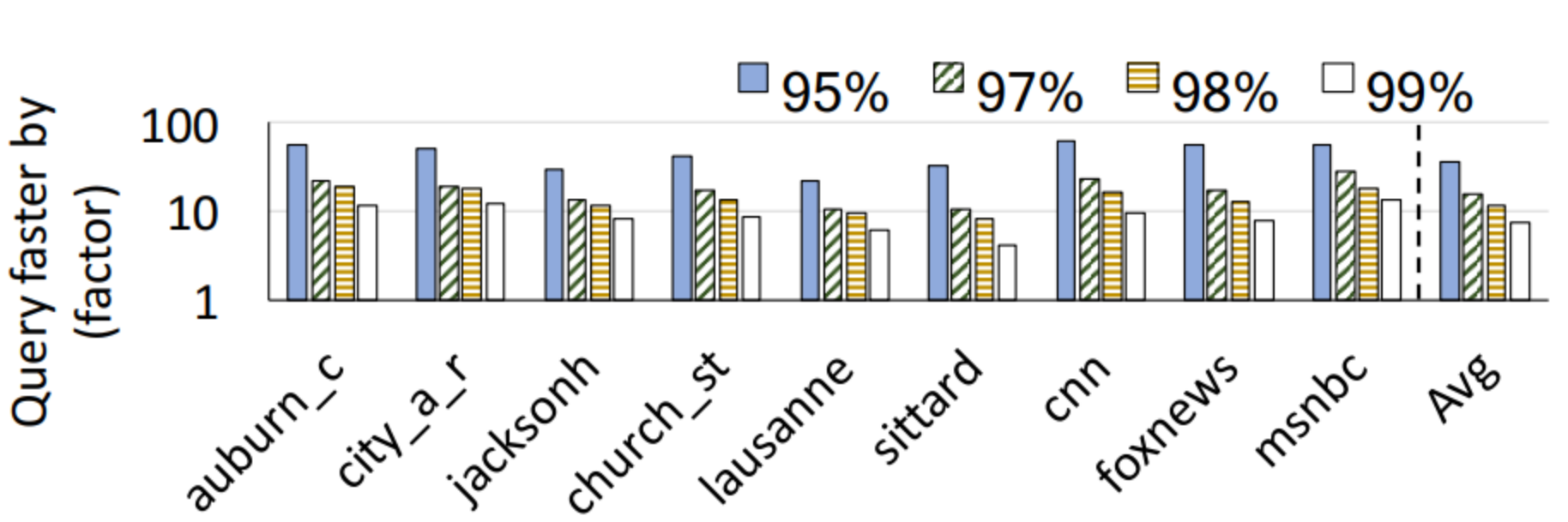
EXPERIMENTS: OPTIONS

Use cases:

- Opt-ingest
  - Traffic camera
- Opt-query
  - Surveillance camera

EXPERIMENTS: DIFFERENT ACCURACY TARGETS

- 97, 98, 99%
- Similar ingest costs
- Query latencies still fast: by 15, 12, and 8x



EXPERIMENTS: DIFFERENT FRAME RATES

- Different applications use different frame rates
- On average, at 30 fps, Focus has 62x cheaper ingest cost
- At lower frame rates, it is 64 to 58x cheaper
- Factors for lowering cost saving, using compressed and specialized models, are not affected by the frame sampling rate

EXPERIMENTS: DIFFERENT FRAME RATES

- Improvement for query latency lowers for lower frame rates
- Less redundancy
- Still faster at very lower frame rate – 1 fps, by 1 order of magnitude

EXPERIMENTS: EXTREME QUERIES

- Every class and every video is queried
  - Still 4x cheaper ingest cost
- Only a tiny percentage of video is queried
  - Still 22-34x faster query latency

STRENGTHS

- Achieves large speedups – 58x ingest cost, 37x query latency; $250/month to $4/month, and 1 hr to 2 min
- Is customizable – allows user to specify accuracy target, and whether to optimize ingest cost or query latency
- Allows user to input ground-truth CNN – possibly an improved one in the future

WEAKNESSES

- Did not talk much about storage space it needs, like for storing the cluster centroids – could be a lot
- Did not measure accuracies per class – some may be more important than others
- Did not talk about how it would handle more complex queries
- How does Focus update index as model is retrained on the fly?
- How does it perform when query asks for object in “Other” class?

DISCUSSION

- Experiment on longer videos
  - Affect class distribution?
- Specialize for a particular video domain
- BlazeIt and Probabilistic Predicates also used cheap neural networks to speed up
- BlazeIt is more of a blackbox; Focus provides options