

Boggart: Towards General-Purpose Acceleration of Retrospective Video Analytics

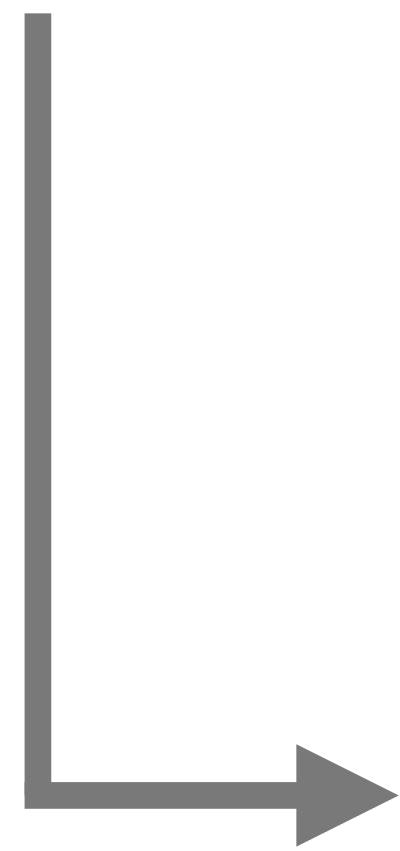
Neil Agarwal, Ravi Netravali

 PRINCETON UNIVERSITY

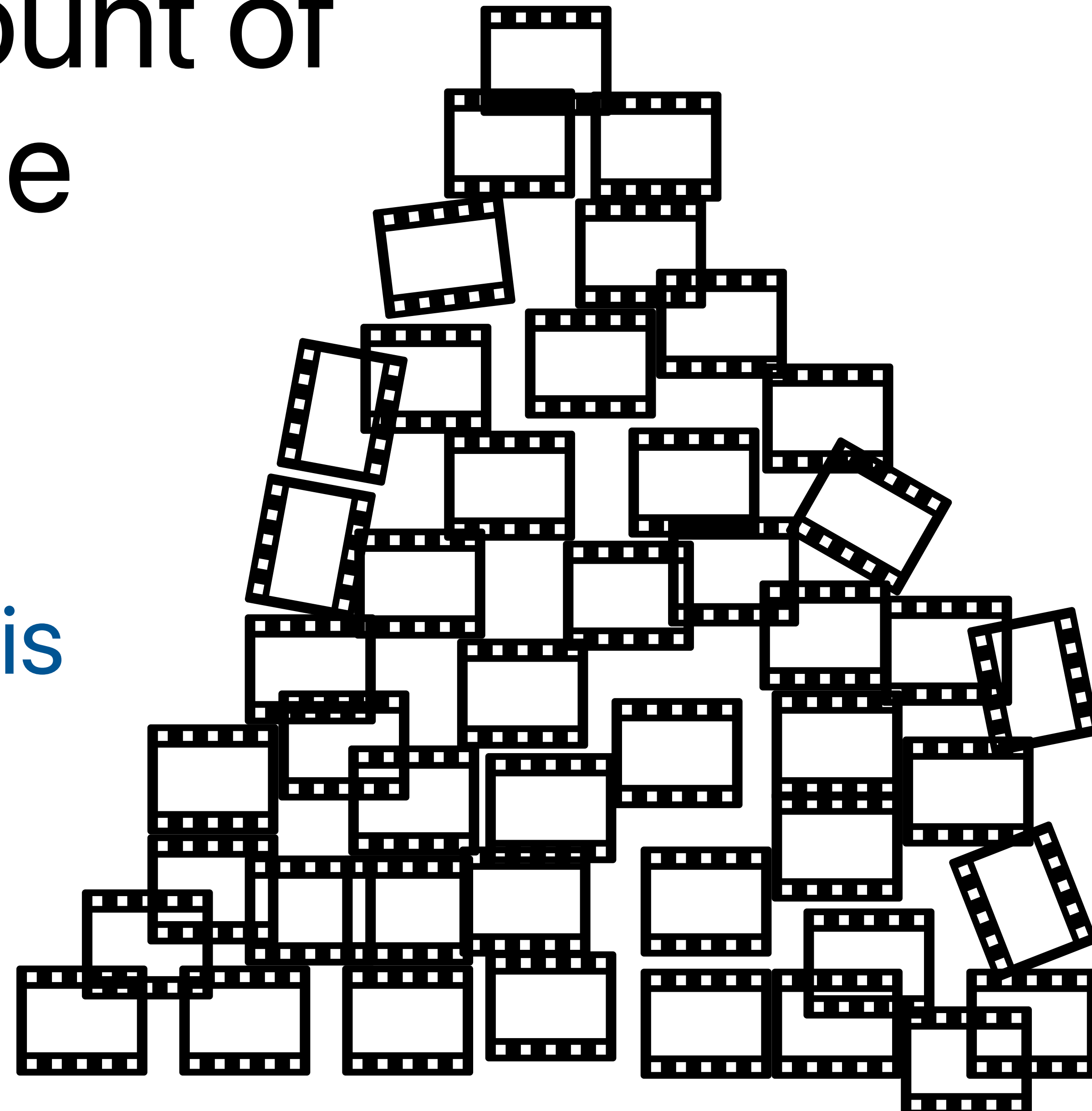
April 18, 2023

NSDI 2023

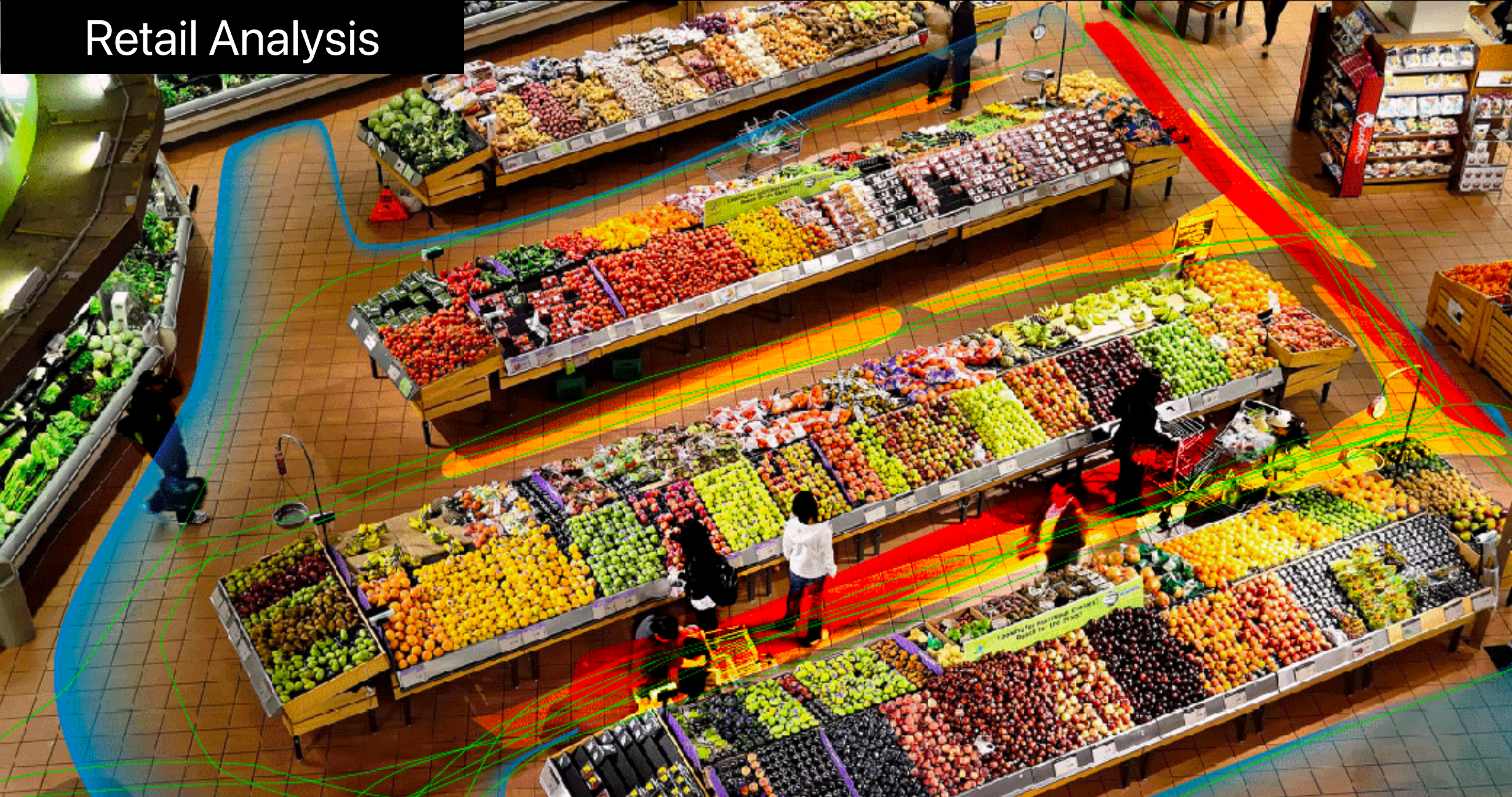
Unprecedented amount of video camera footage



After-the-fact analysis



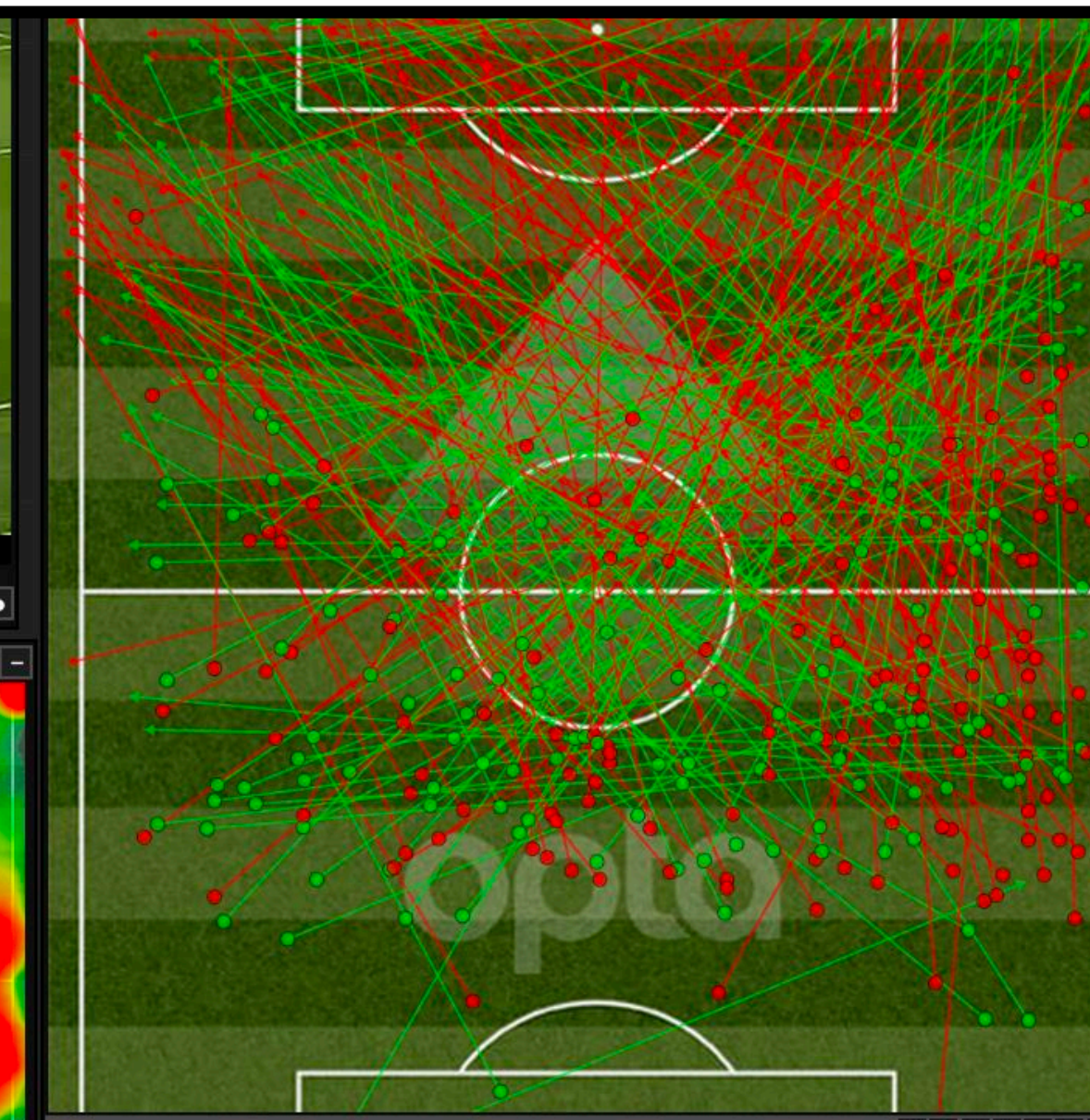
Retail Analysis



Traffic Analysis



Retrospective Video Analytics



ANGLE SELECTION

LENGTH SELECTION

HEATMAP

0 360

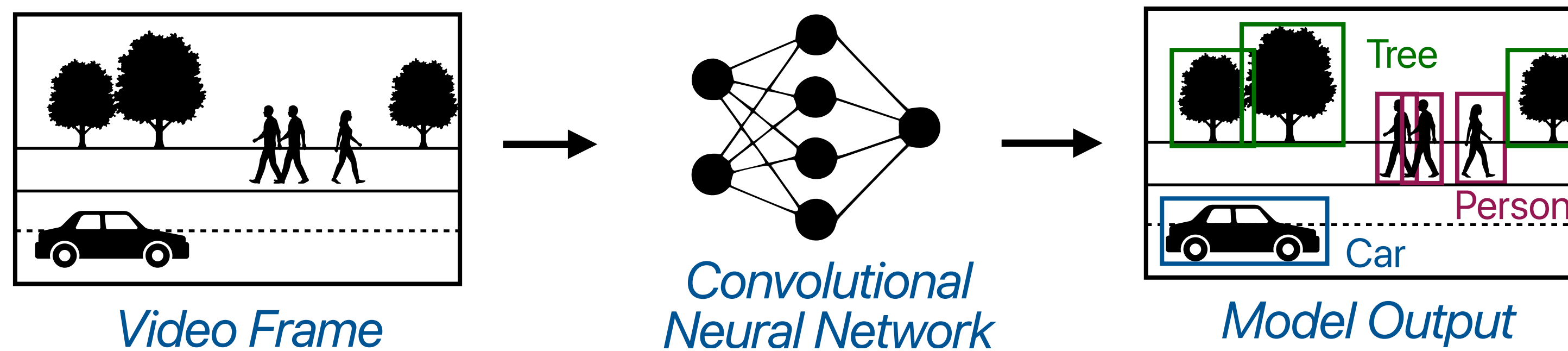
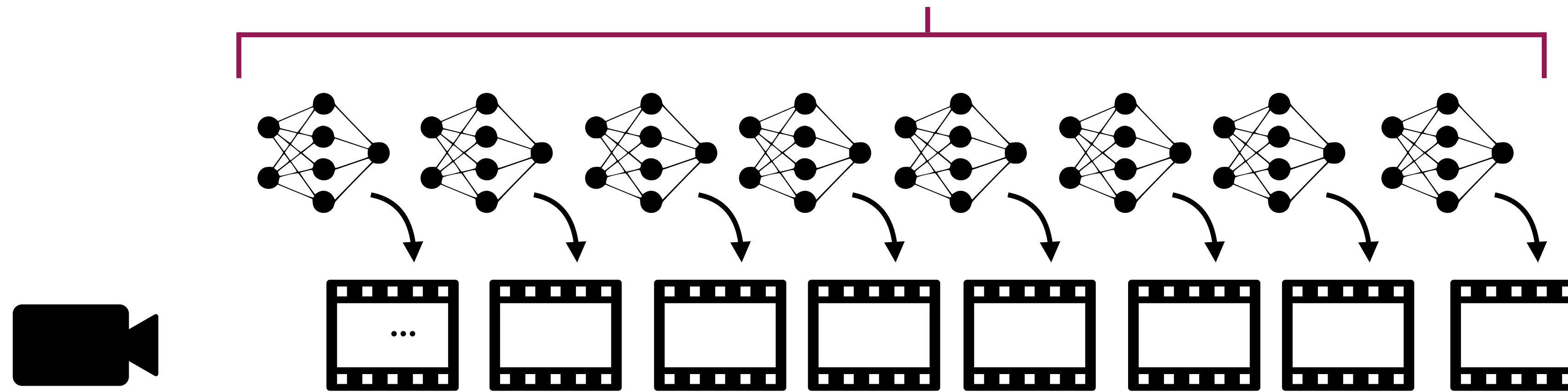
50 100

Sports Analysis

Audits/Investigations

Retrospective Video Analytics Pipeline

Challenge: High Compute Overheads → Querying is Expensive & Slow



Acceleration Strategy: Model-Specific Preprocessing

Preprocessing

Query Execution

Acceleration Strategy: Model-Specific Preprocessing

Preprocessing

Extract model-specific content similarities

Query Execution

Acceleration Strategy: Model-Specific Preprocessing

Preprocessing

Extract model-specific content similarities

Query Execution

Run model sparingly to label similar content

Acceleration Strategy: Model-Specific Preprocessing

Preprocessing

Extract model-specific content similarities

...



...

Query Execution

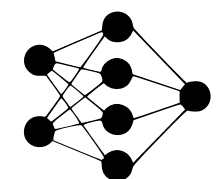
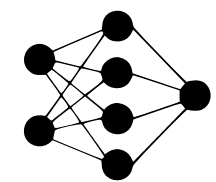
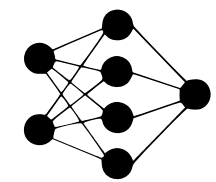
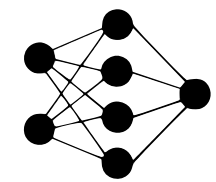
Run model sparingly to label similar content

Acceleration Strategy: Model-Specific Preprocessing

Preprocessing

Extract model-specific content similarities

...



...

Query Execution

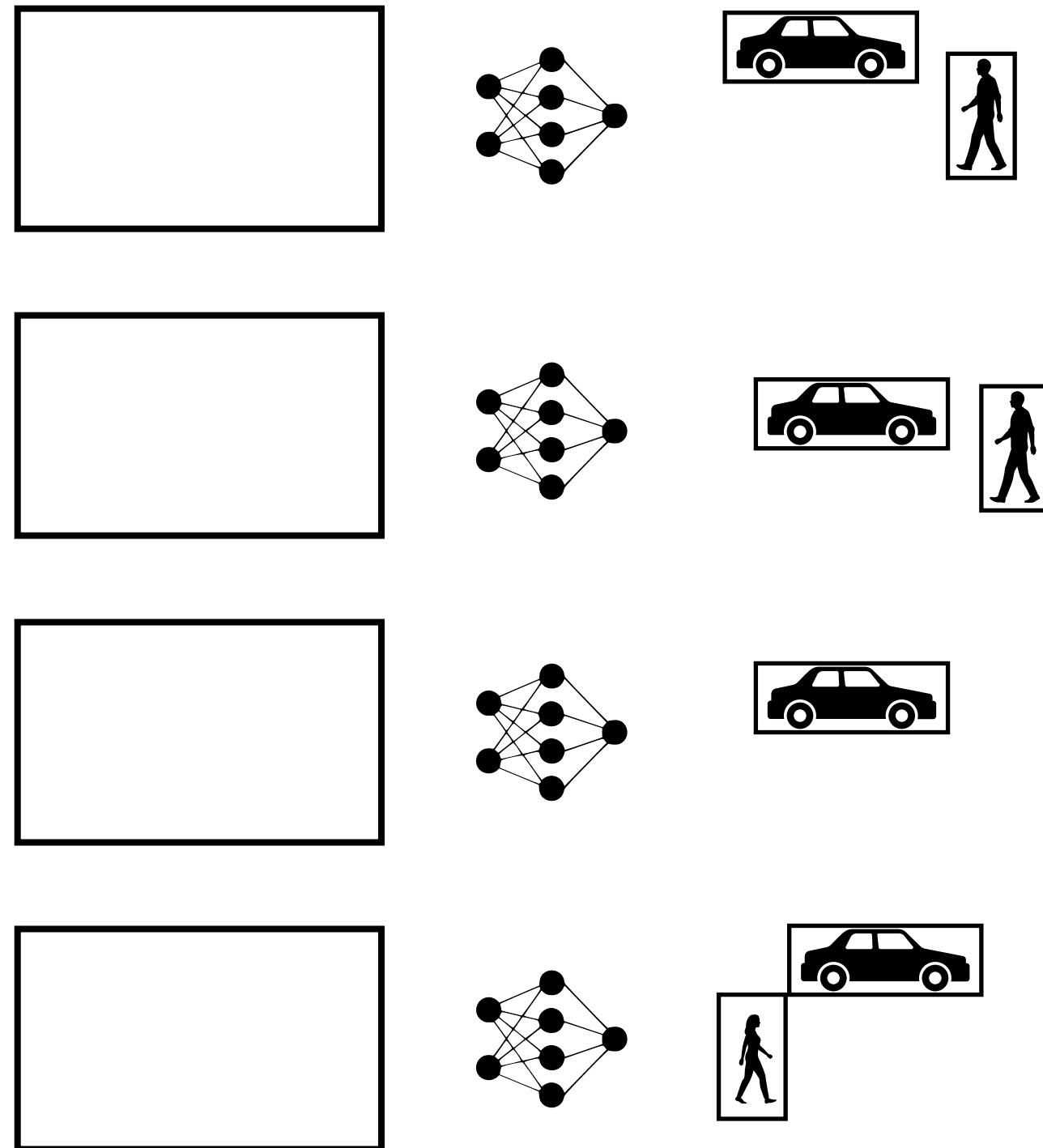
Run model sparingly to label similar content

Acceleration Strategy: Model-Specific Preprocessing

Preprocessing

Extract model-specific content similarities

...



...

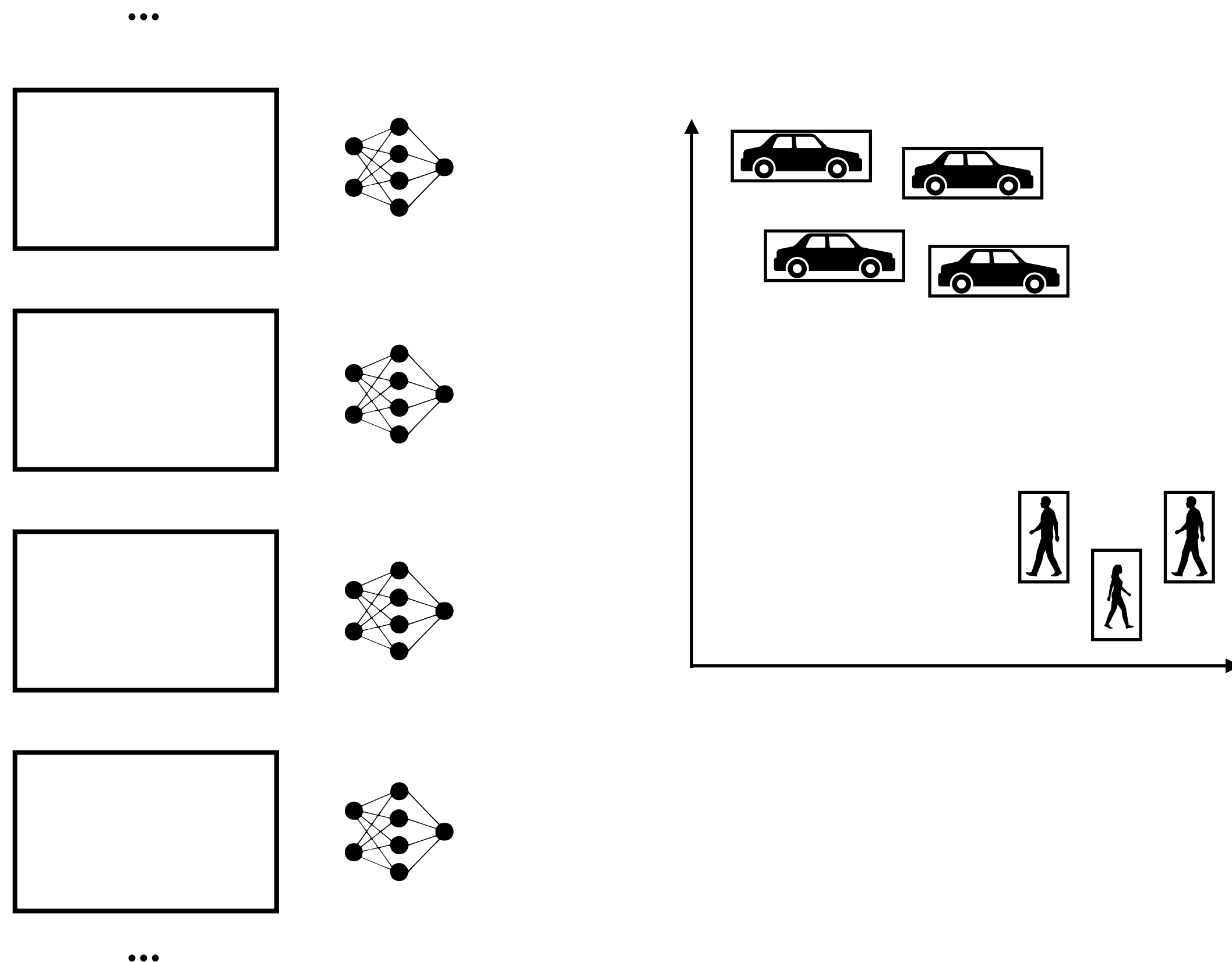
Query Execution

Run model sparingly to label similar content

Acceleration Strategy: Model-Specific Preprocessing

Preprocessing

Extract model-specific content similarities



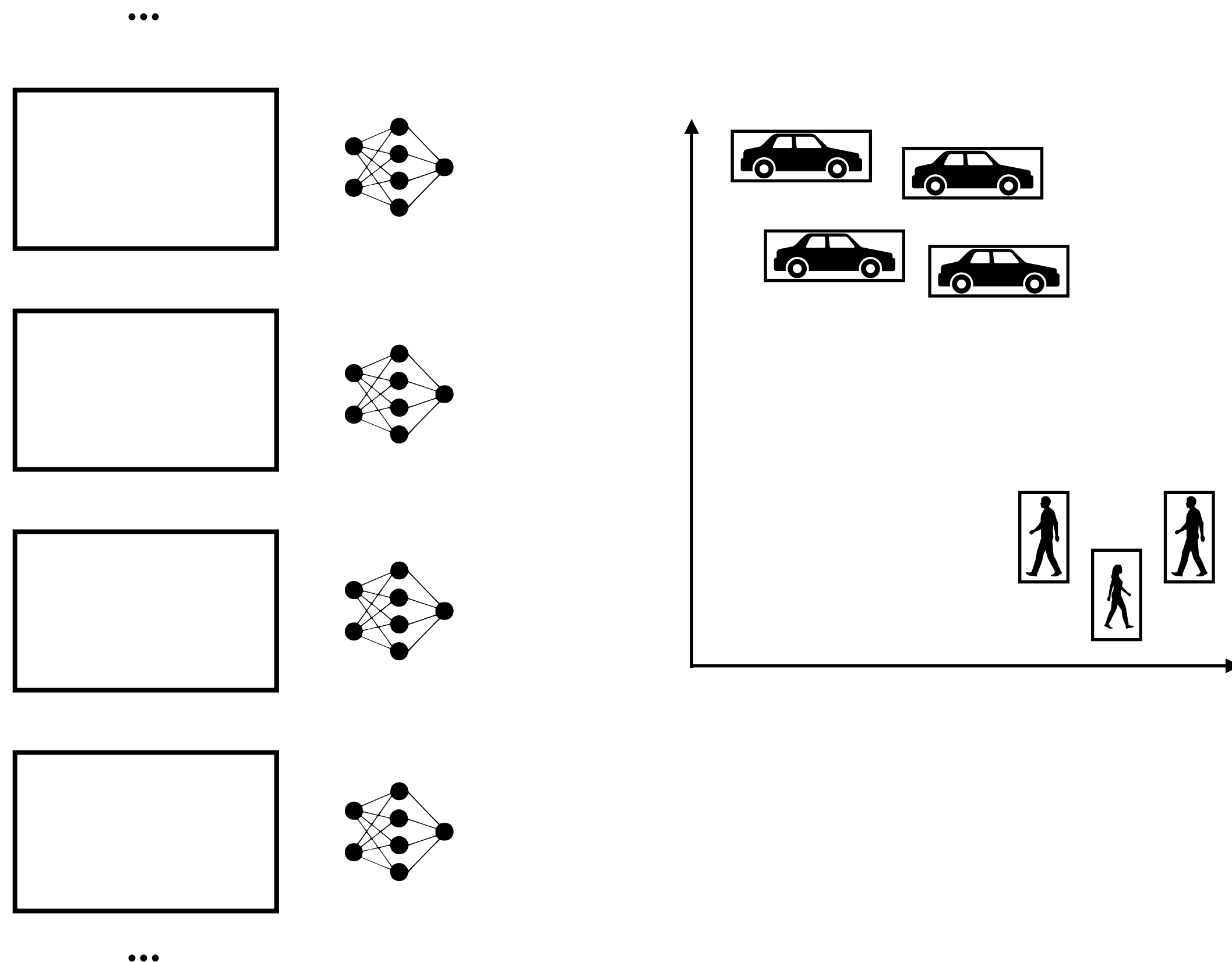
Query Execution

Run model sparingly to label similar content

Acceleration Strategy: Model-Specific Preprocessing

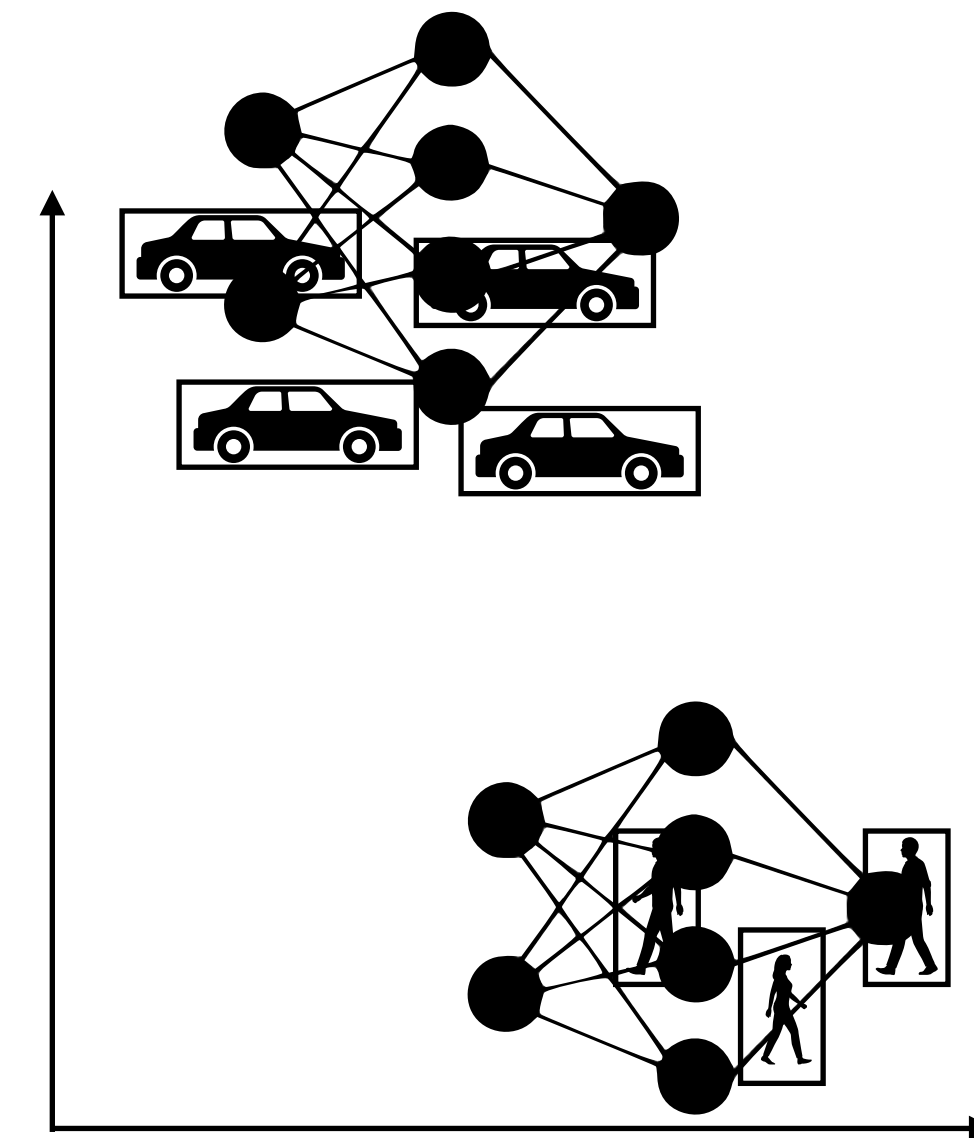
Preprocessing

Extract model-specific content similarities



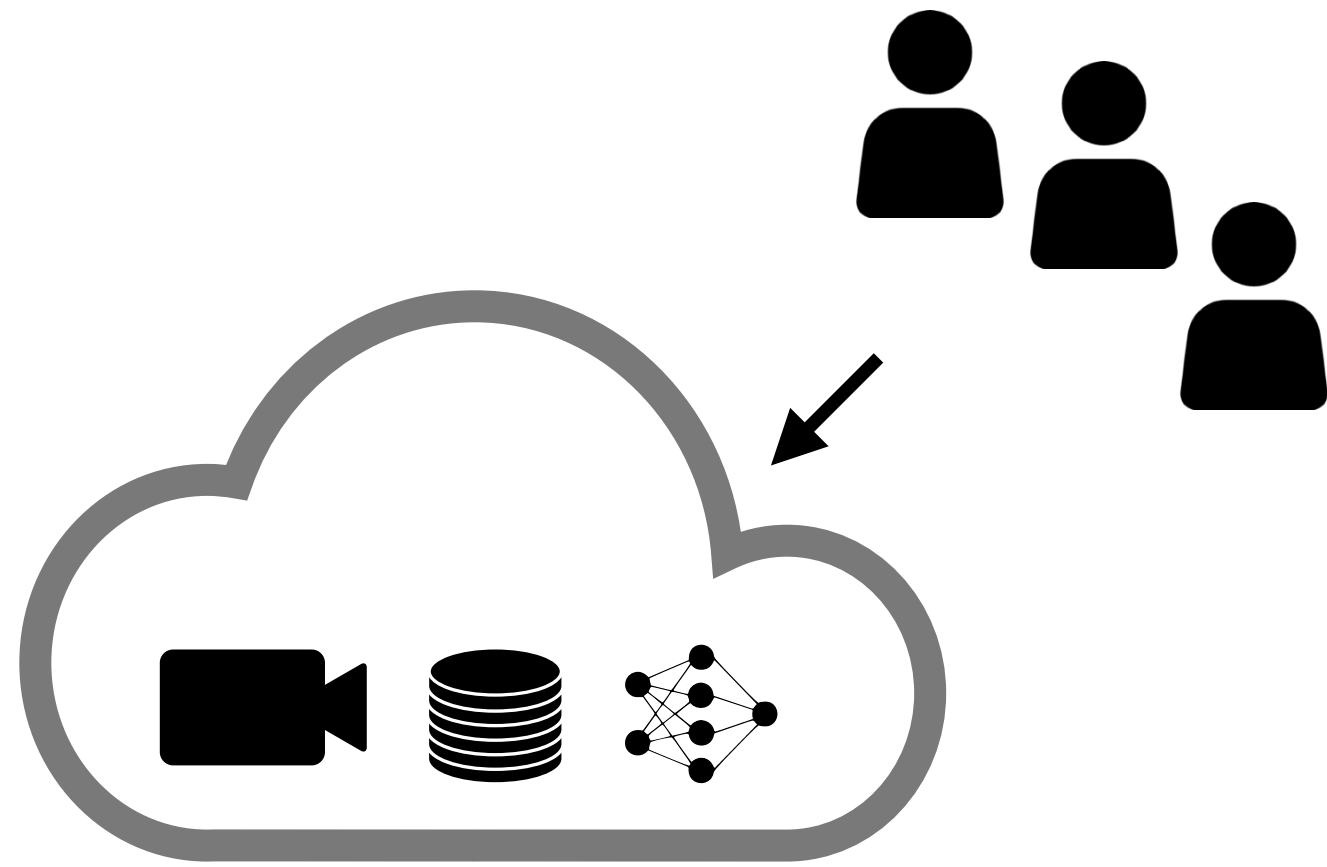
Query Execution

Run model sparingly to label similar content



Querying Behavior

Previously

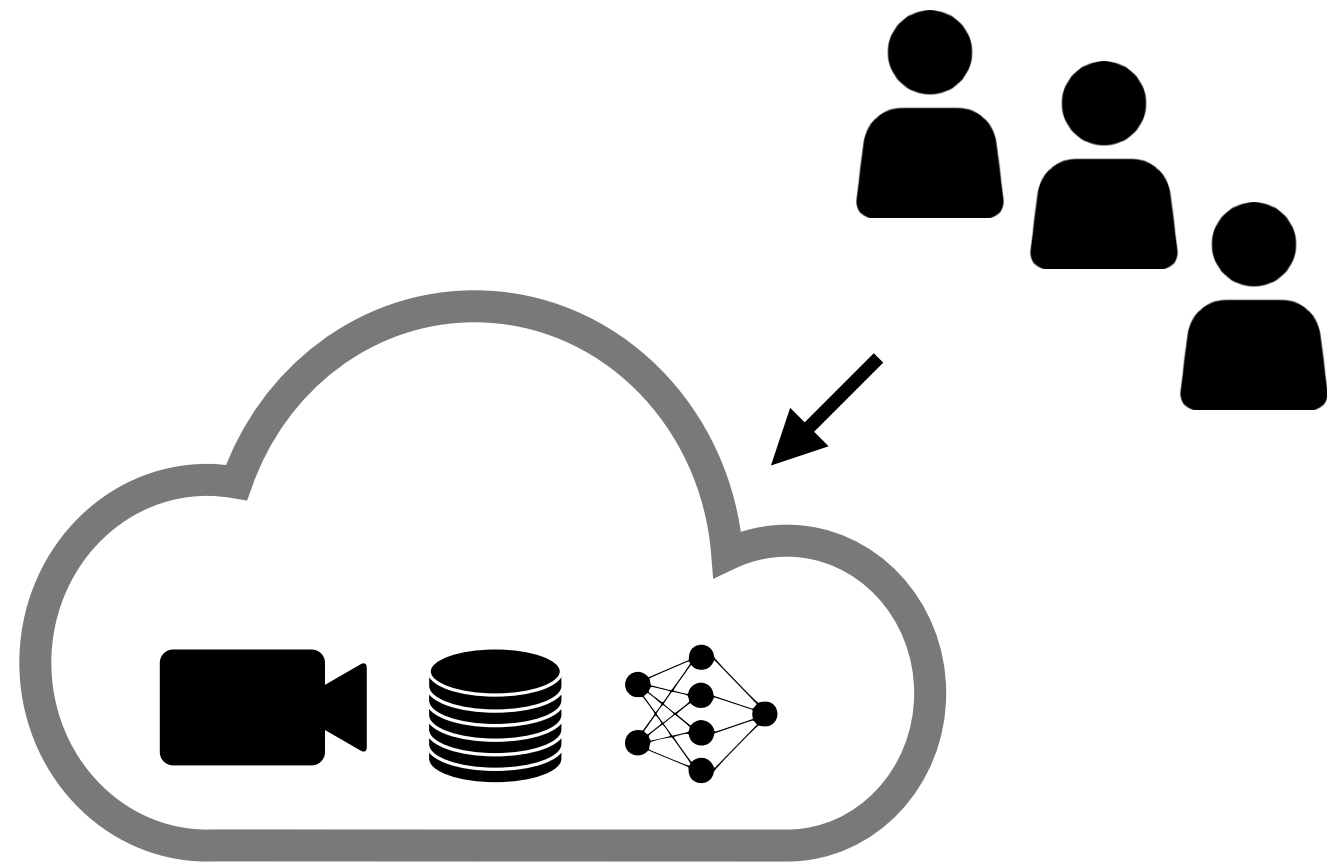


Today

Implication:
preprocessing model = query model

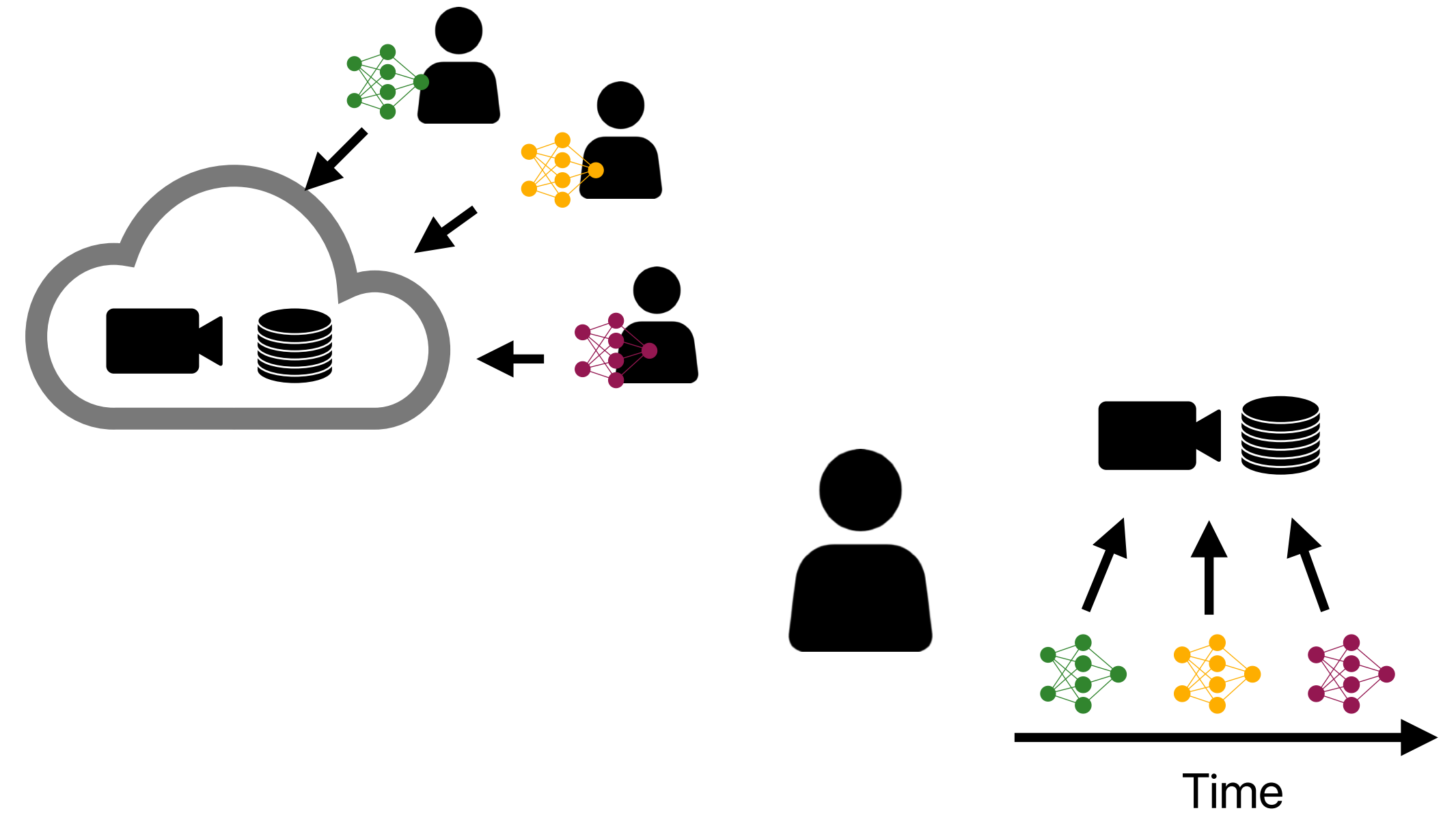
Querying Behavior

Previously



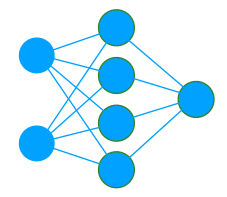
Implication:
preprocessing model = query model

Today

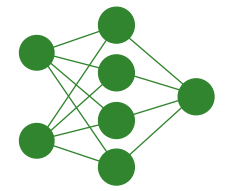


Implication:
preprocessing model \neq query model

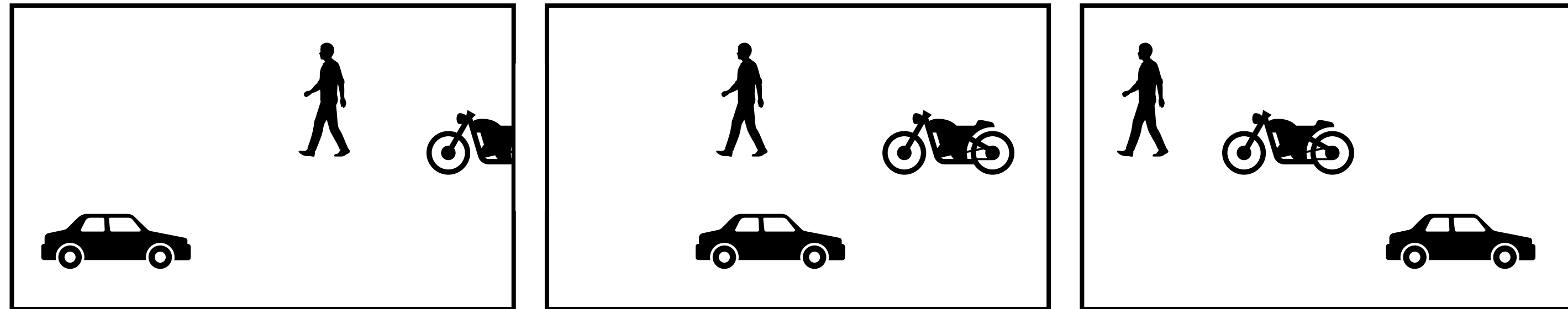
Models Behave Differently



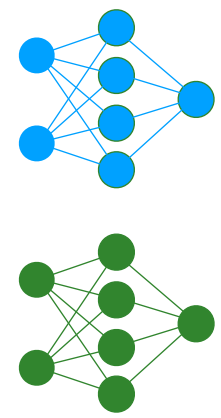
Model 1



Model 2

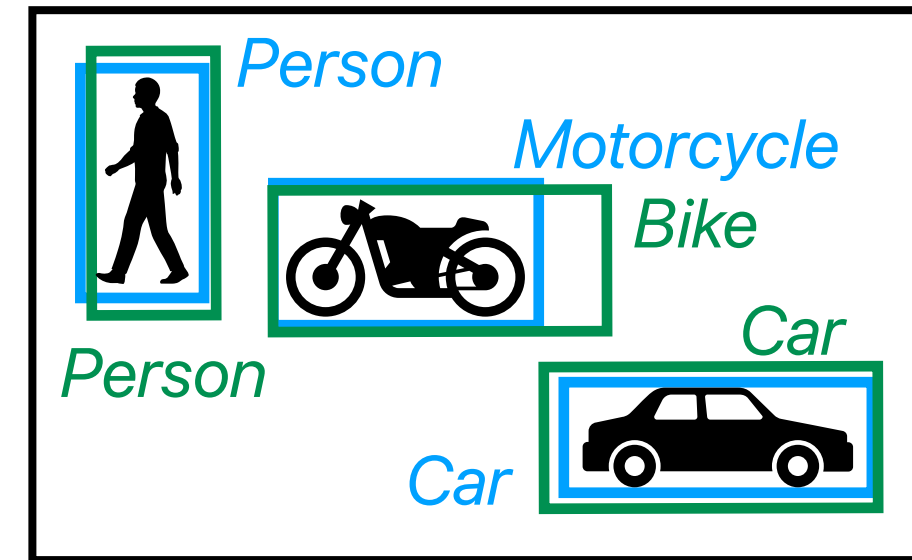
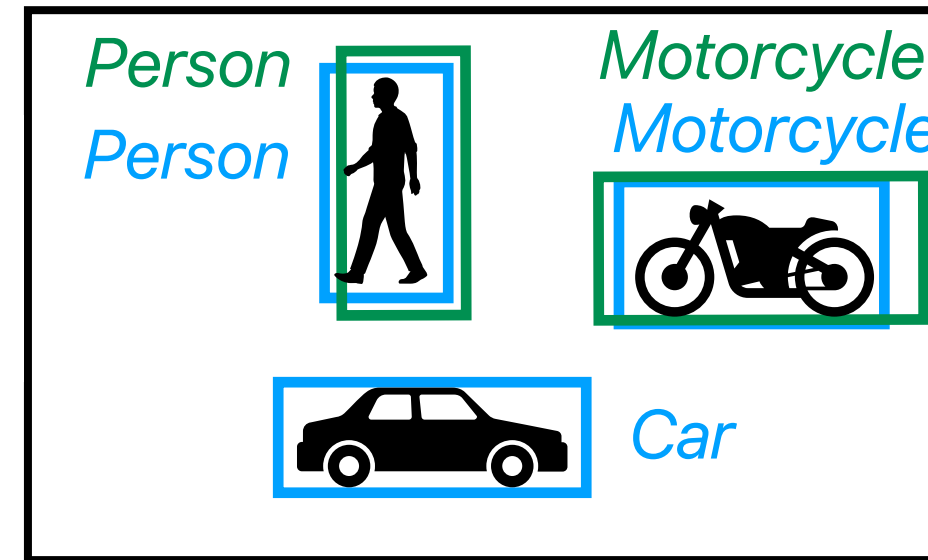
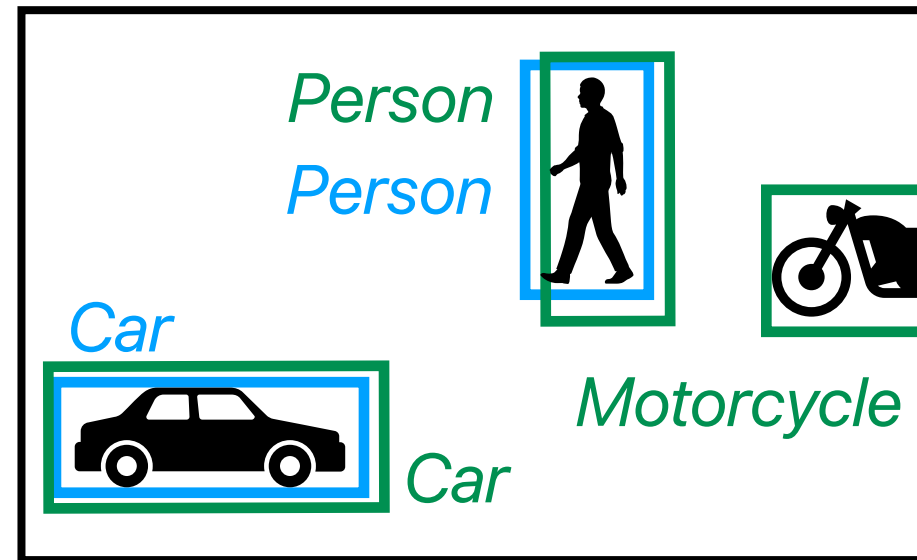


Models Behave Differently

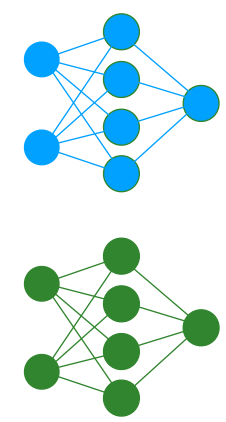


Model 1

Model 2

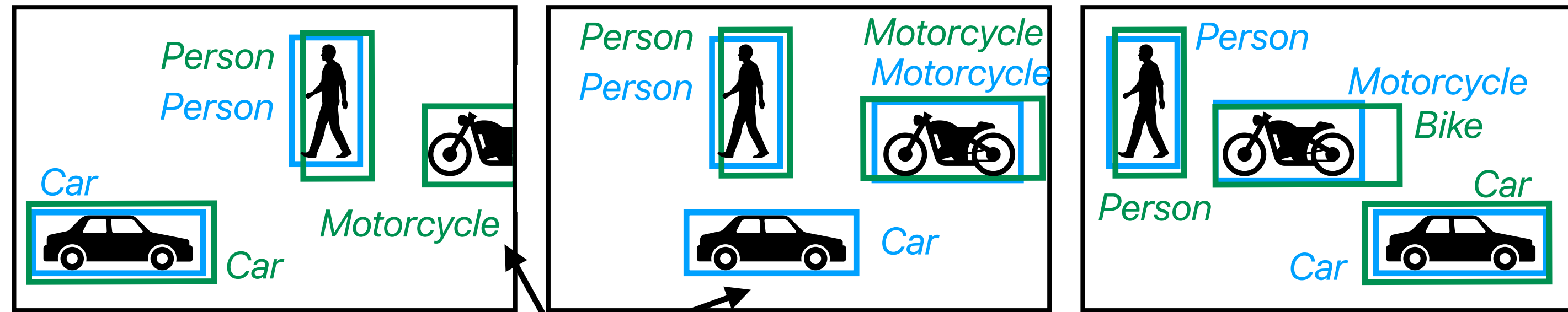


Models Behave Differently



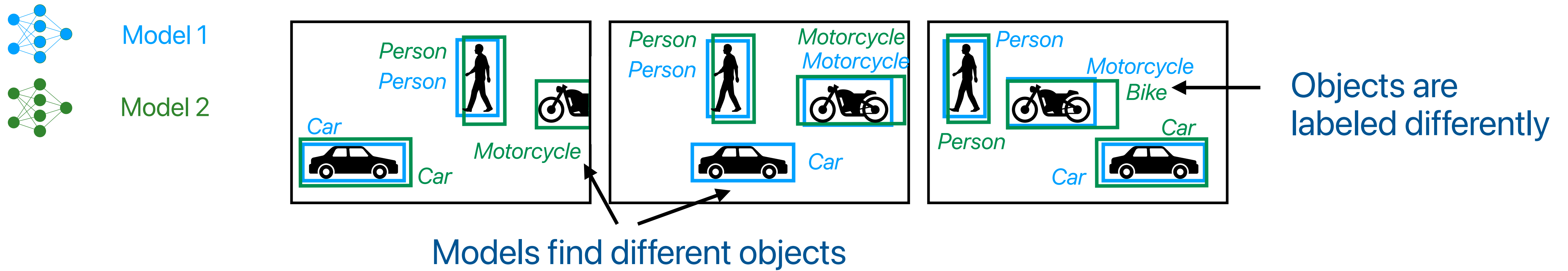
Model 1

Model 2

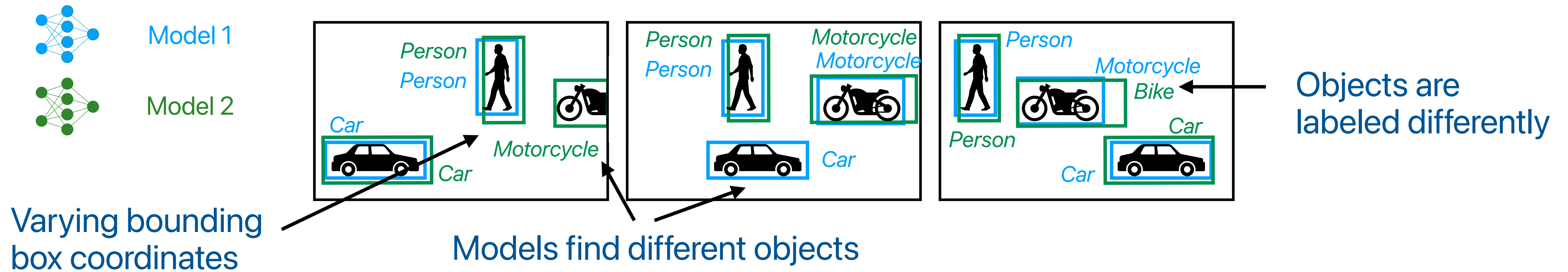


Models find different objects

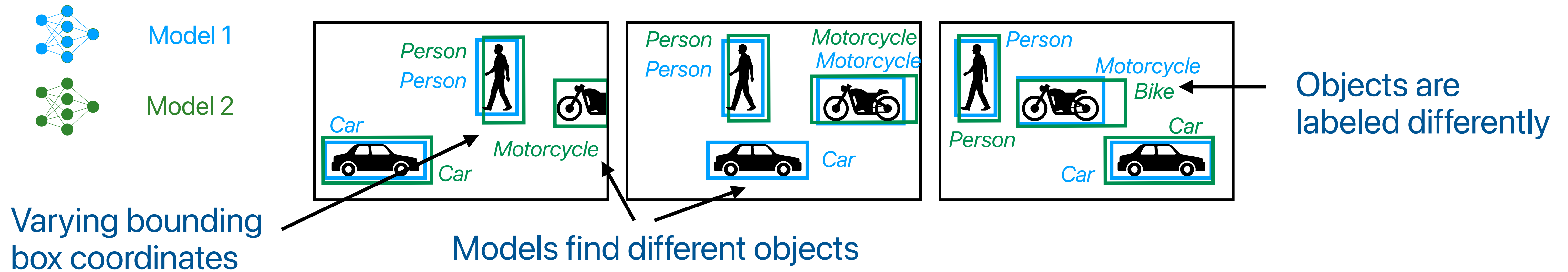
Models Behave Differently



Models Behave Differently

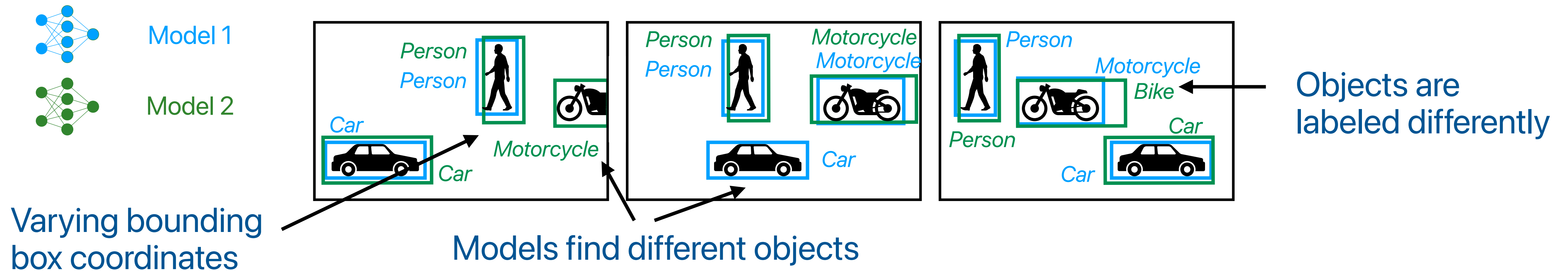


Models Behave Differently



Preprocessing Model: Model 2
Query Model: Model 1

Models Behave Differently



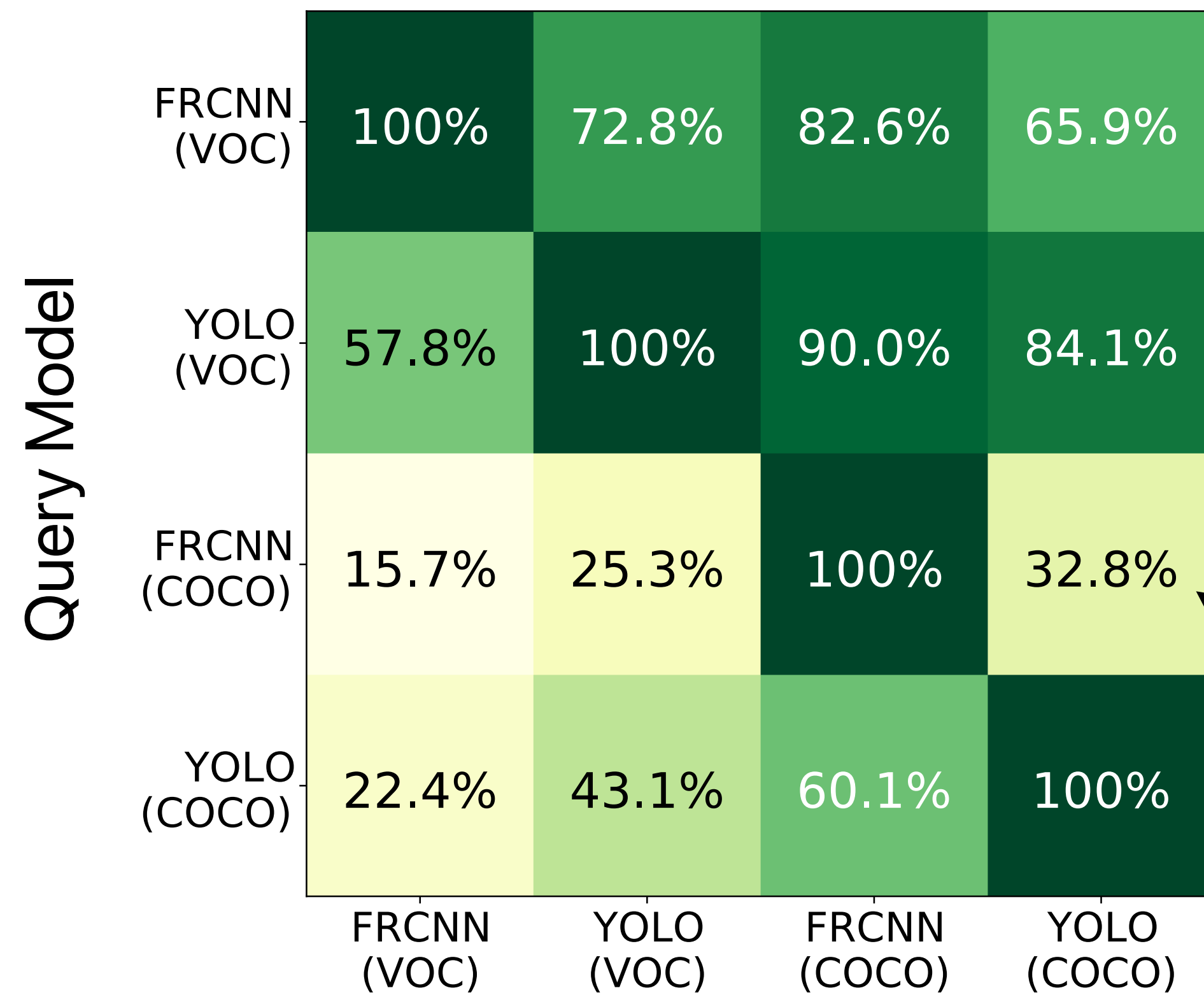
Preprocessing Model: Model 2
Query Model: Model 1

Query: Counting # of cars per frame
Accuracy: $\text{avg}(100\%, 0\%, 100\%) = 66\%$

Discrepancies Across Real Models

Discrepancies Across Real Models

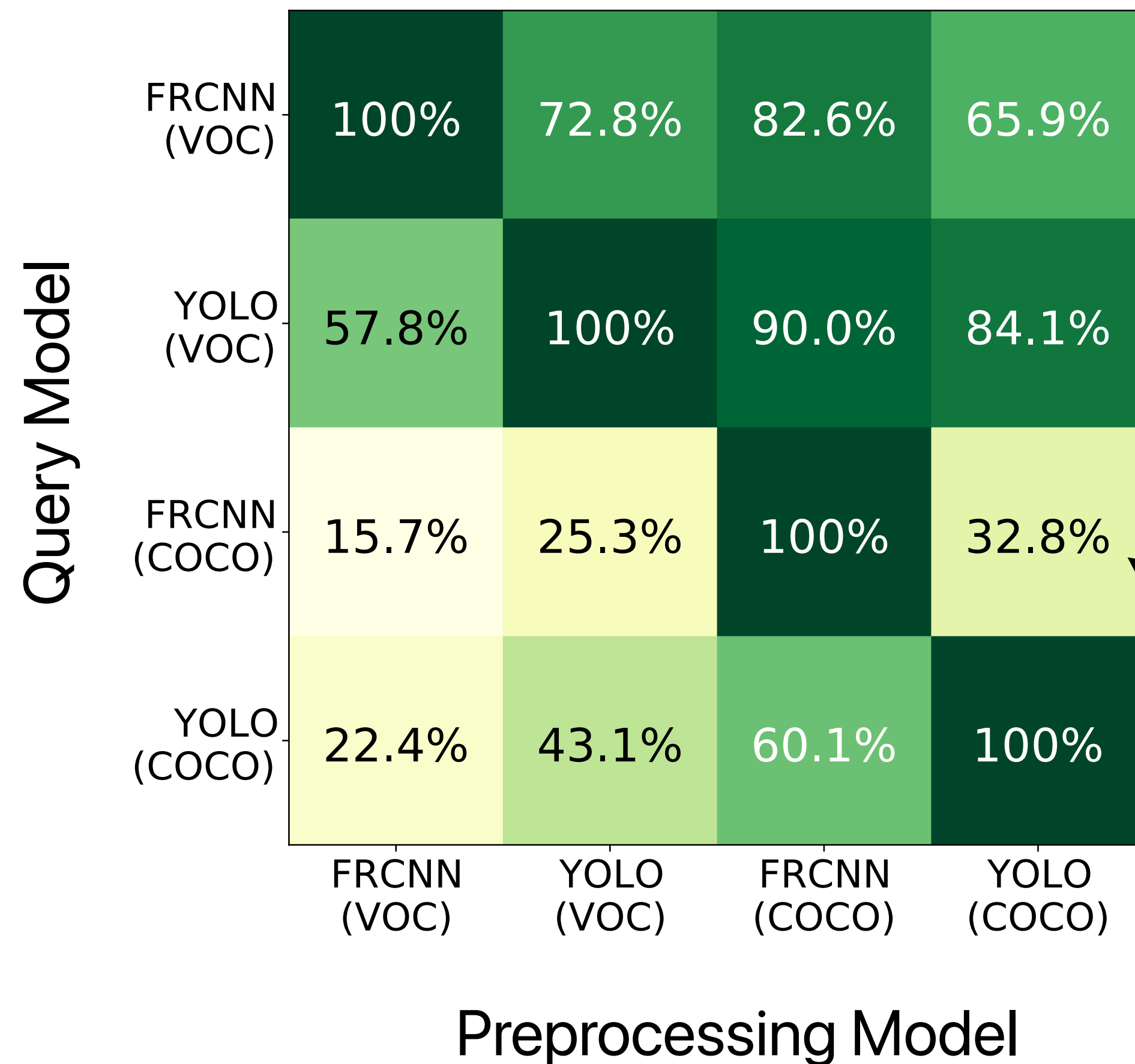
Query: Counting # Cars per Frame



Query accuracy of preprocessing with YOLO model trained on the COCO dataset but querying with FRCNN model trained on the COCO dataset is 32.8%

Discrepancies Across Real Models

Query: Counting # Cars per Frame



Accuracy of Full Dataset Analysis

Counting Queries: 16-92%

Bounding Box Queries: 6-54%

Query accuracy of preprocessing with YOLO model trained on the COCO dataset but querying with FRCNN model trained on the COCO dataset is 32.8%

Boggart

baa · grt

How do you preprocess video data to accelerate retrospective querying with diverse models?

Preprocessing Requirements

Preprocessing Requirements

- 1 Relatively cheap to perform

Preprocessing Requirements

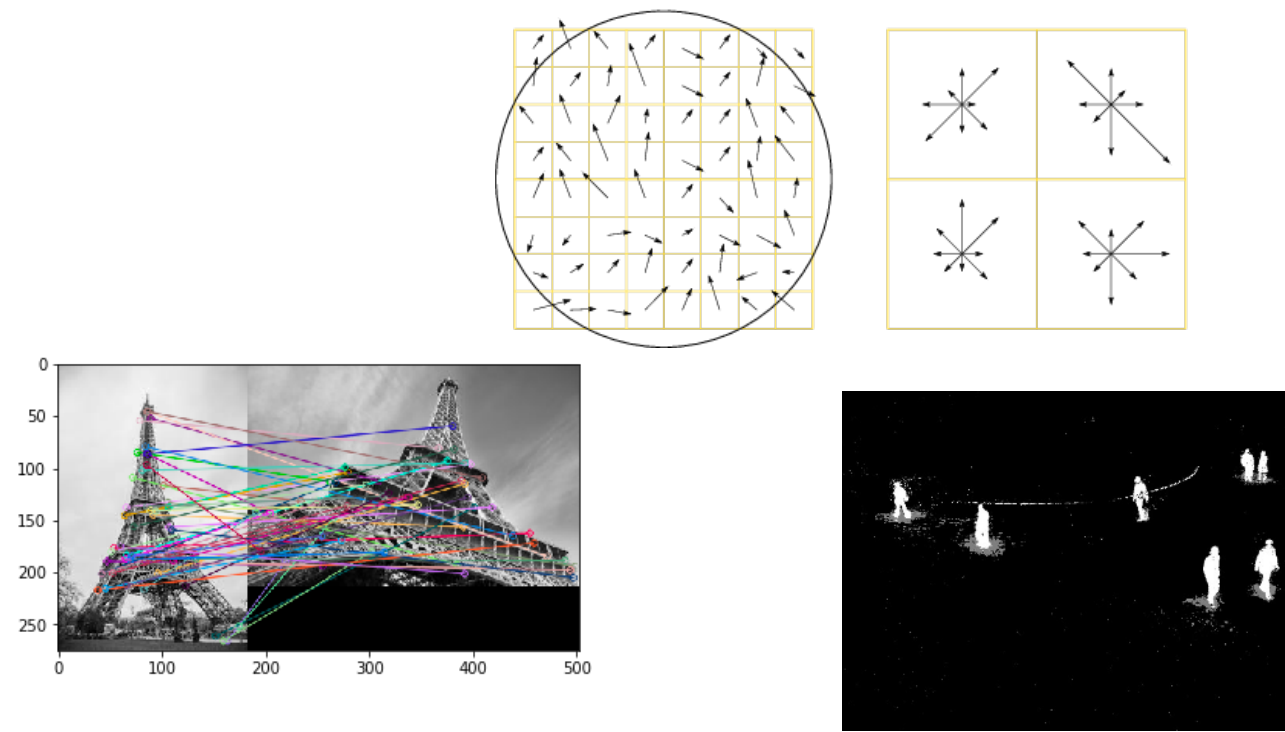
- 1 Relatively cheap to perform
- 2 General-purpose and comprehensive

Preprocessing Requirements

- 1 Relatively cheap to perform
- 2 General-purpose and comprehensive
- 3 Provide a way to link information across frames

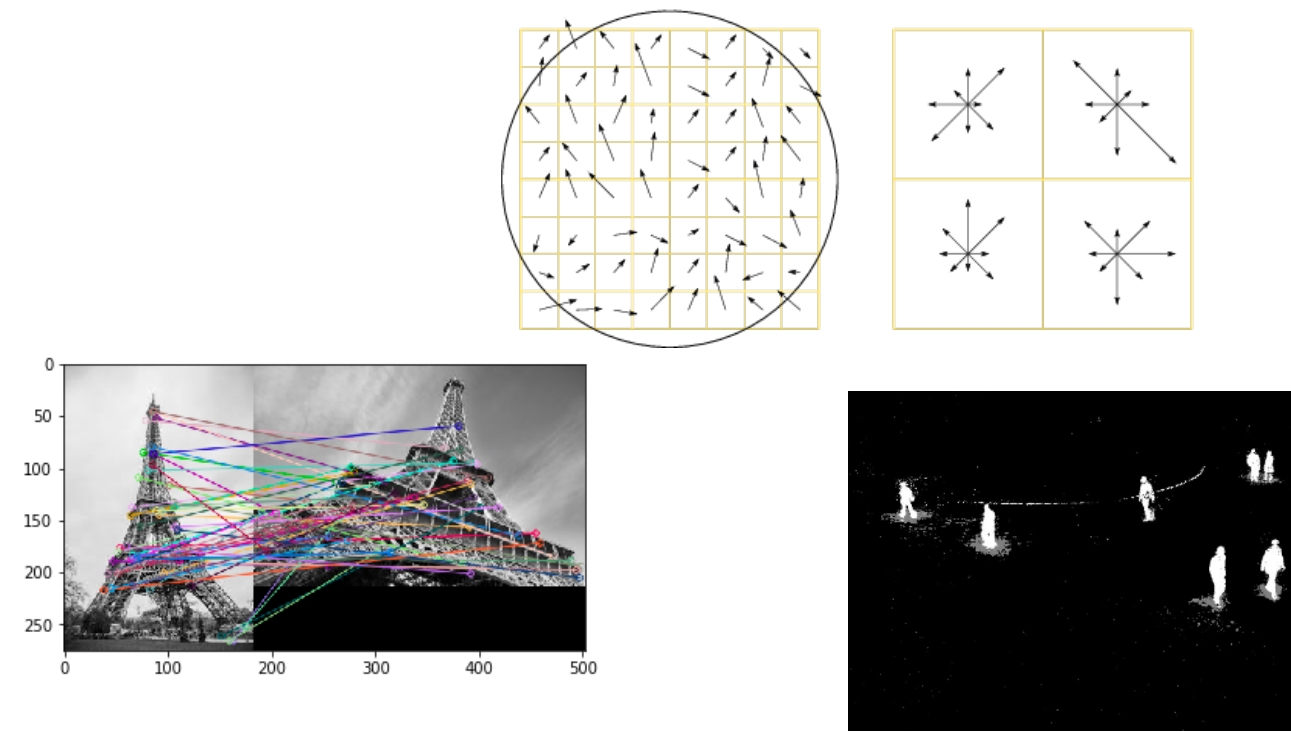
Boggart's Insight

Boggart's Insight



Classical Computer Vision Techniques

Boggart's Insight

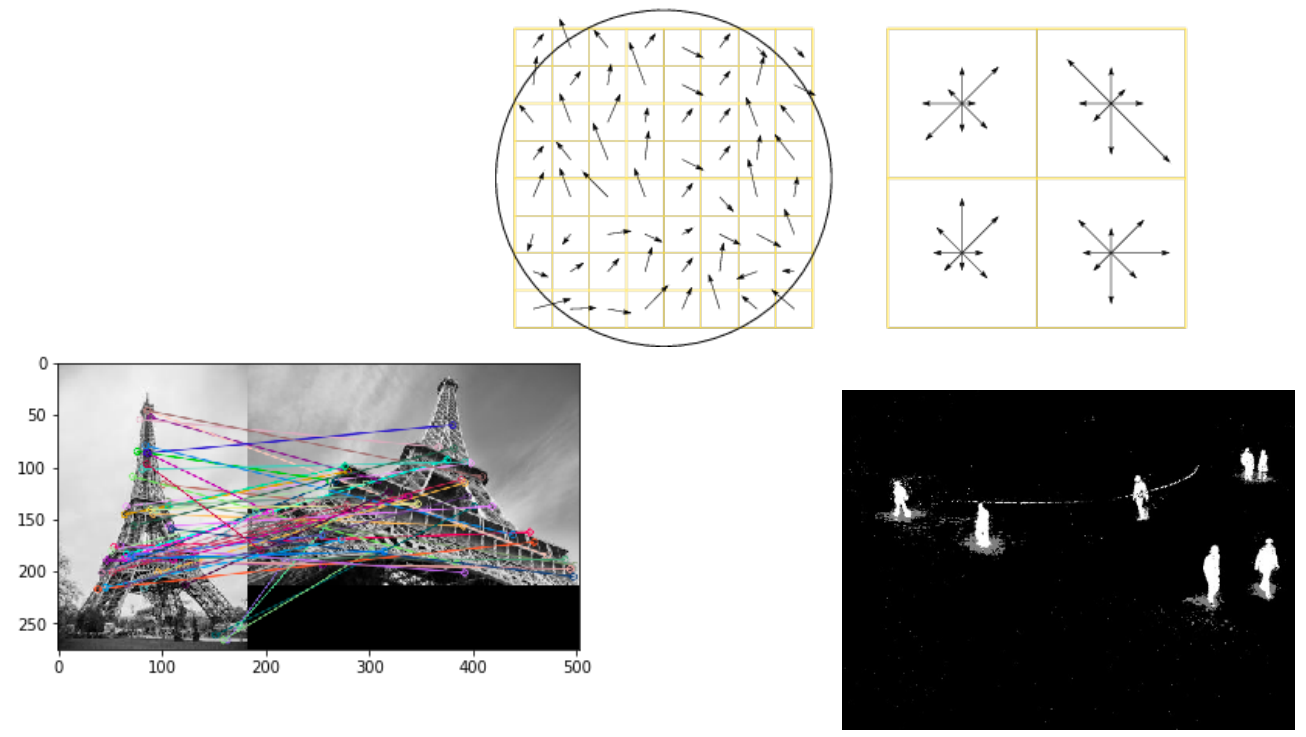


- ▶ Can be leveraged to comprehensively & generically extract info from a video




Classical Computer Vision Techniques

Boggart's Insight

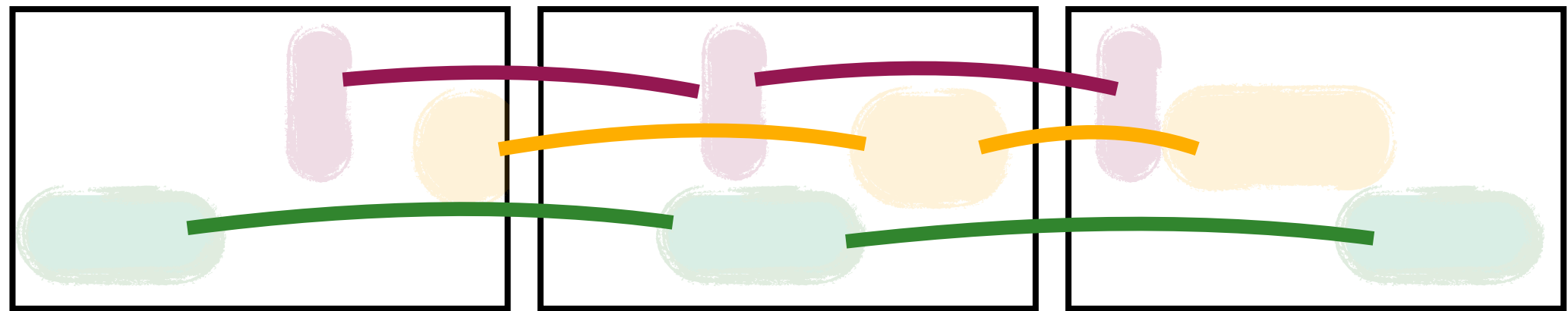


Classical Computer Vision Techniques

- ▶ Can be leveraged to comprehensively & generically extract info from a video 

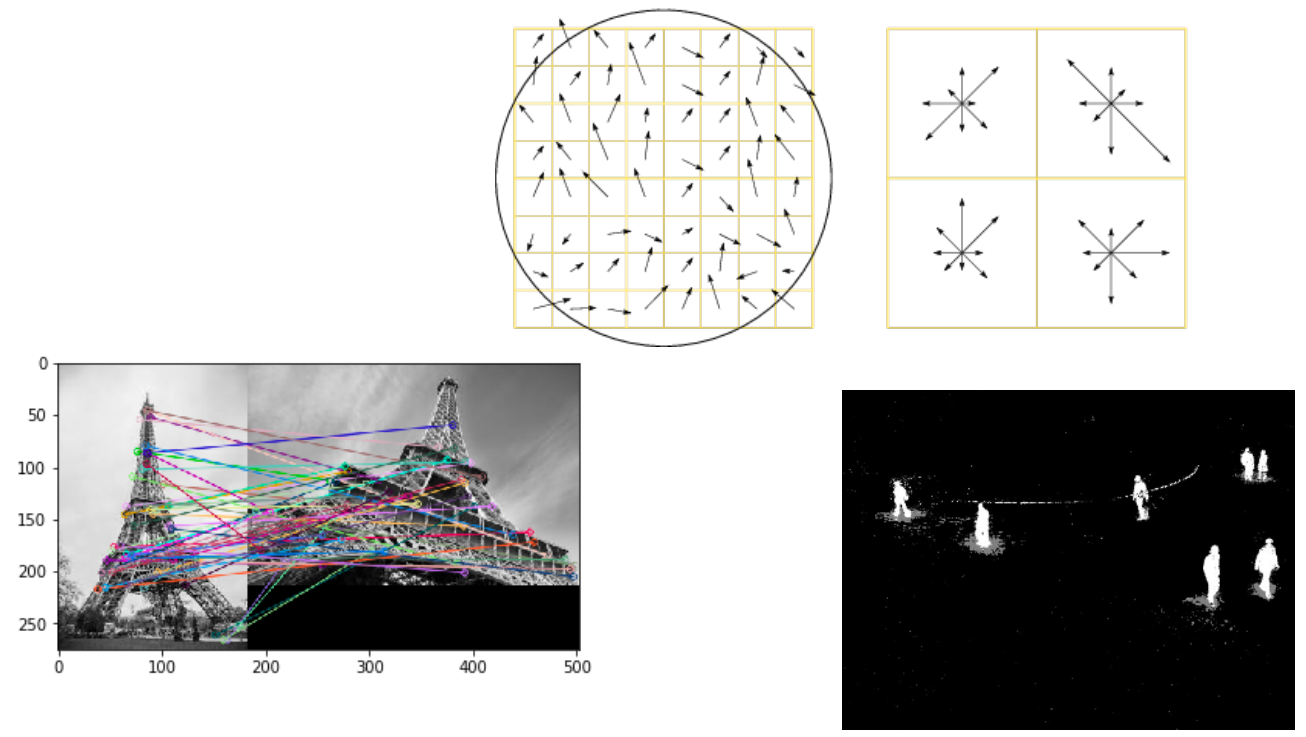
The image shows four examples of classical computer vision techniques: a circular optical flow field, a 2x2 grid of starburst patterns, a grayscale image of the Eiffel Tower with colored motion trajectories and a coordinate axis, and a dark image of people with white motion trajectories.

Preprocessing





Extracting trajectories of areas of motion

Boggart's Insight

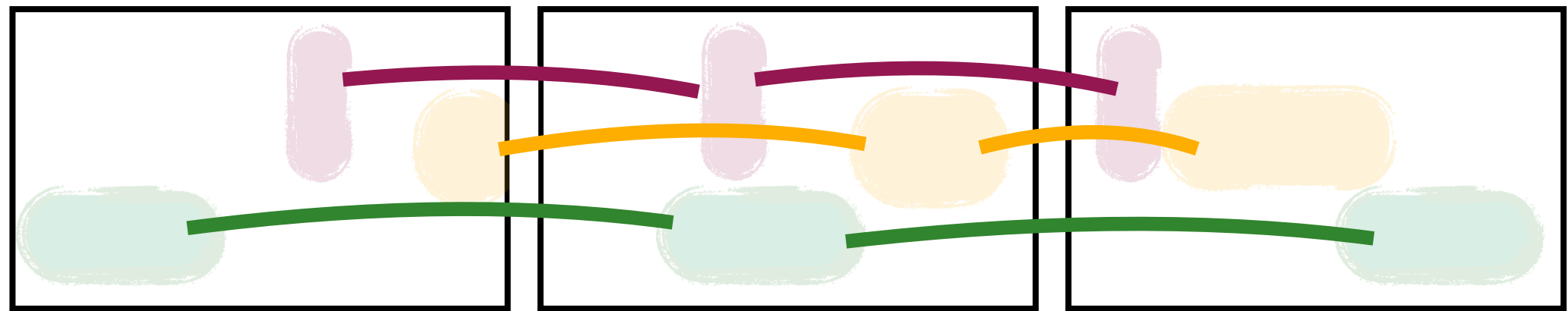


Classical Computer Vision Techniques

- ▶ Can be leveraged to comprehensively & generically extract info from a video 
- ▶ Less accurate than ML 

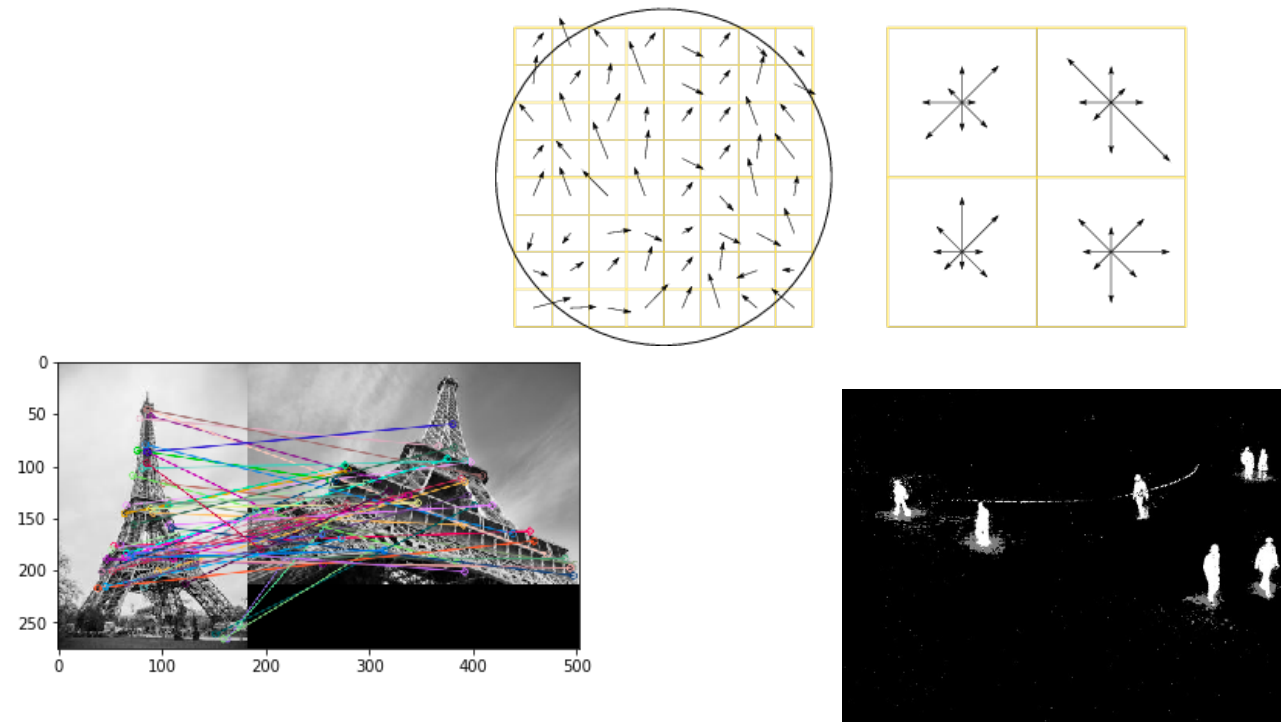
The image shows four examples of classical computer vision techniques: 1) A grid of arrows representing optical flow. 2) A 2x2 grid of star-like patterns. 3) A grayscale image of the Eiffel Tower with a coordinate system and colored lines connecting corresponding points in two frames. 4) A dark image of a scene with white spots and lines indicating motion paths.

Preprocessing





Extracting trajectories of areas of motion

Boggart's Insight

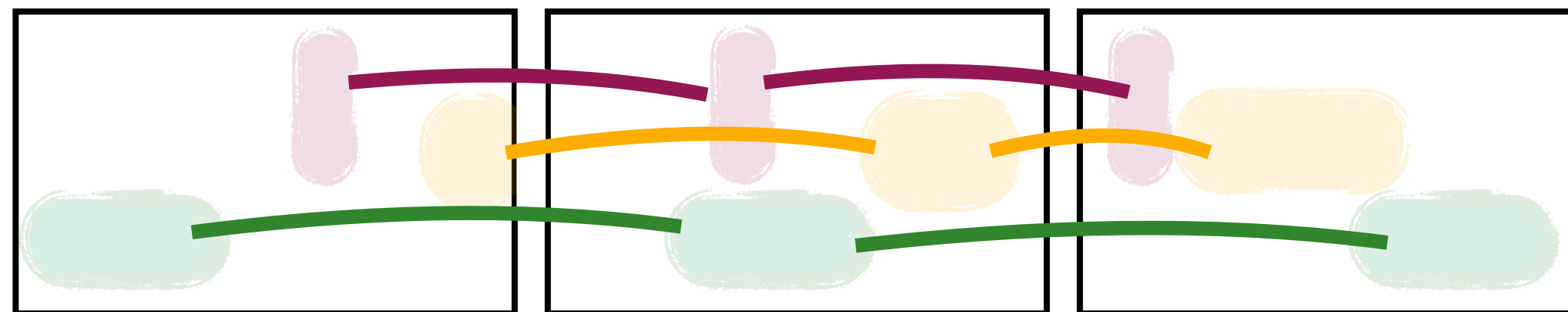


Classical Computer Vision Techniques

- ▶ Can be leveraged to comprehensively & generically extract info from a video 
- ▶ Less accurate than ML 

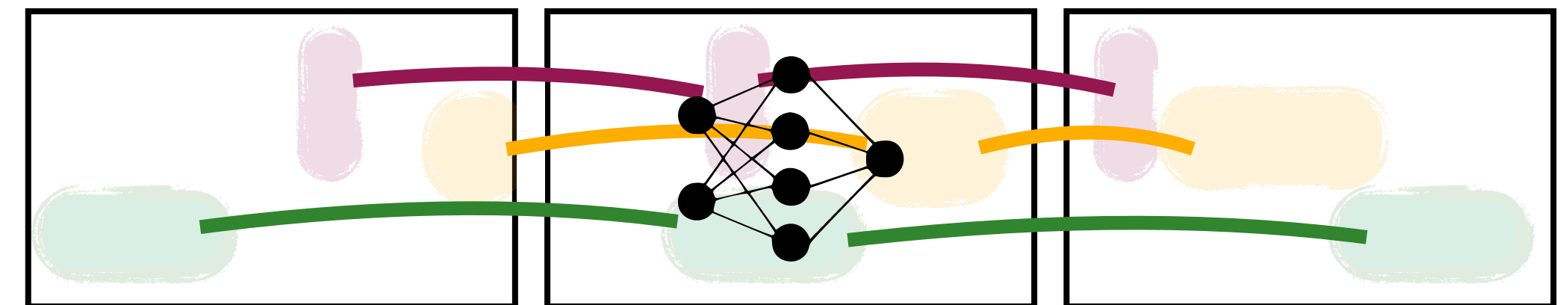
The image shows four examples of classical computer vision techniques: 1) A grid of arrows representing optical flow. 2) A 2x2 grid of star-like patterns. 3) A grayscale image of the Eiffel Tower with colored lines connecting corresponding points across two frames. 4) A dark image of a scene with white spots and lines, possibly representing object tracking or feature matching.

Preprocessing



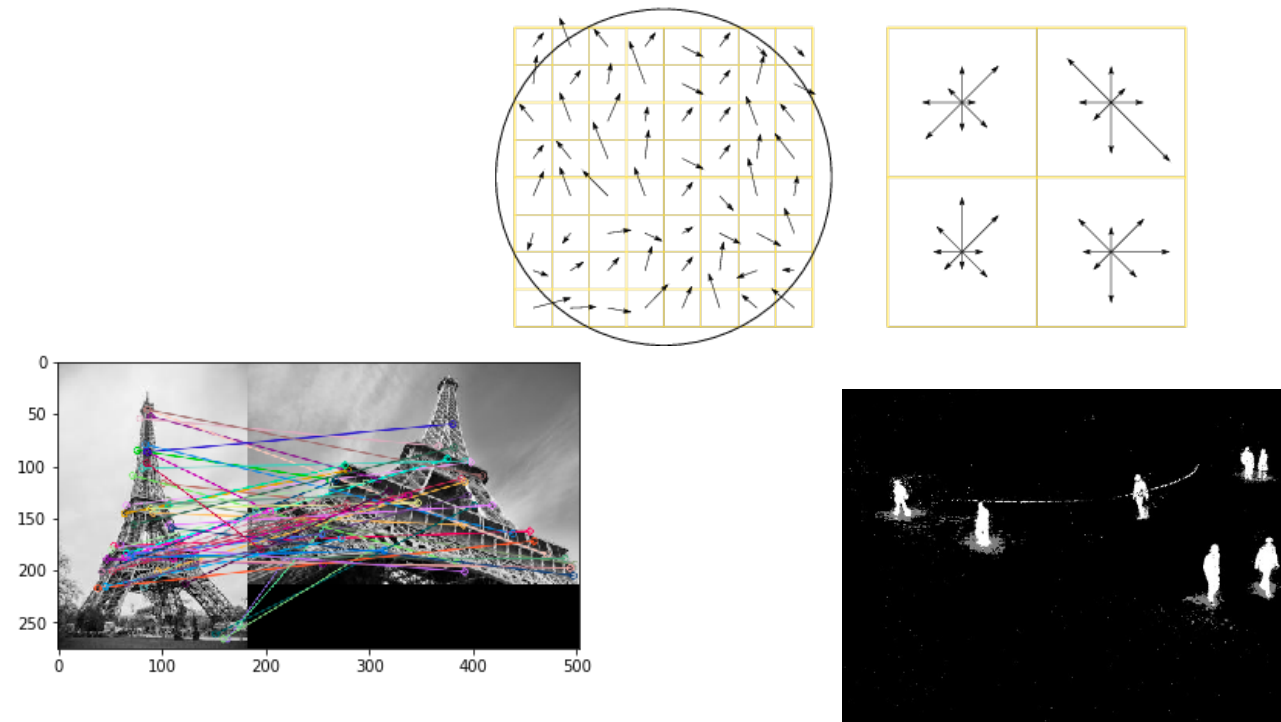
Extracting trajectories of areas of motion

Query Execution





Model-specific labeling & propagation

Boggart's Insight

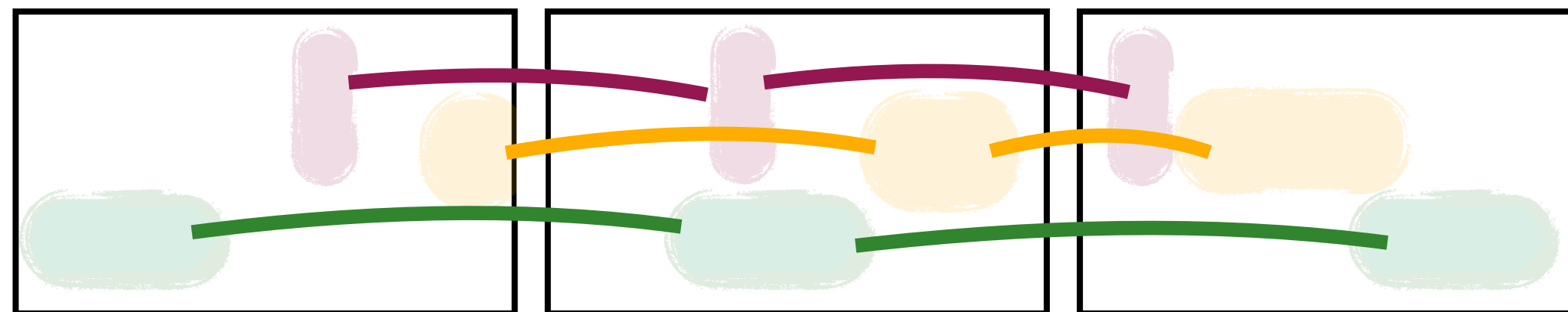


Classical Computer Vision Techniques

- ▶ Can be leveraged to comprehensively & generically extract info from a video 
- ▶ Less accurate than ML 

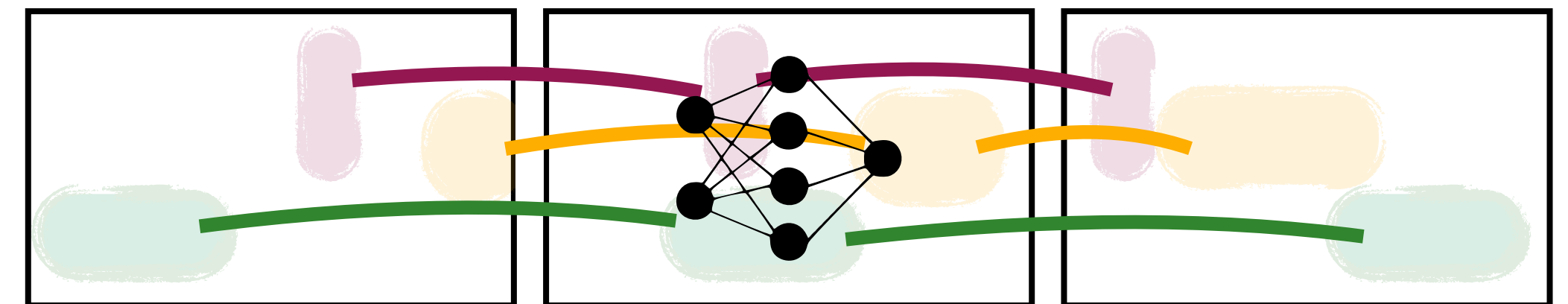
The image shows four examples of classical computer vision techniques: a vector field on a grid, a 2x2 grid of star-like patterns, a video frame of the Eiffel Tower with colored motion trajectories, and a dark frame with white motion trajectories.

Preprocessing



Extracting trajectories of areas of motion

Query Execution



Model-specific labeling & propagation

Preprocessing

Trajectories of Blobs

Frame ID	Trajectory ID	x1	y1	x2	y2
1	1	100	200	100	300
1	2	200	600	300	500
1	3	80	120	90	230
2	1	105	205	105	305
...



Preprocessing

Trajectories of Blobs

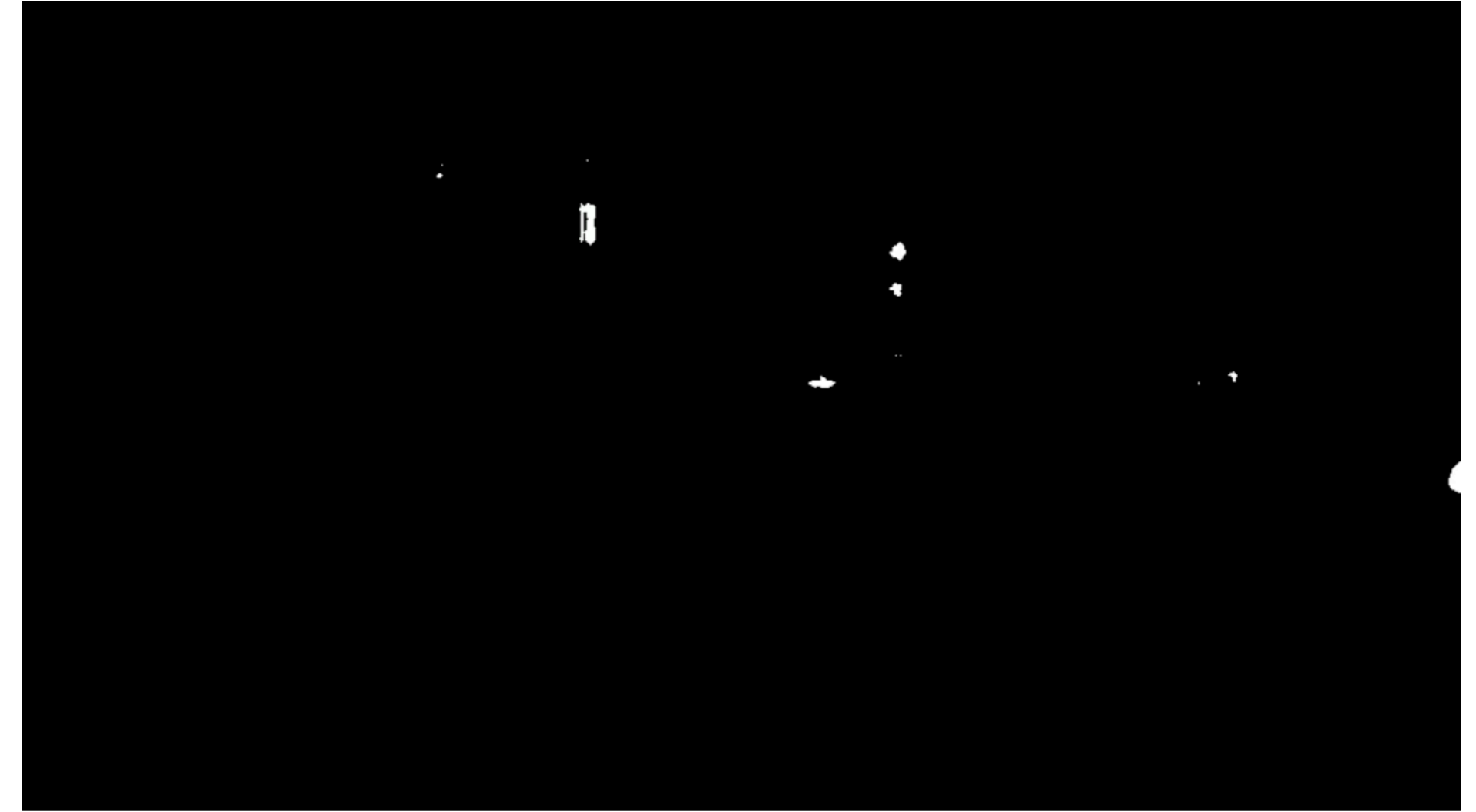
Frame ID	Trajectory ID	x1	y1	x2	y2
1	1	100	200	100	300
1	2	200	600	300	500
1	3	80	120	90	230
2	1	105	205	105	305
...





Raw Video

➔ Background Estimate ➔



Foreground (Moving Pixels)

Foreground Extraction



Blob Extraction



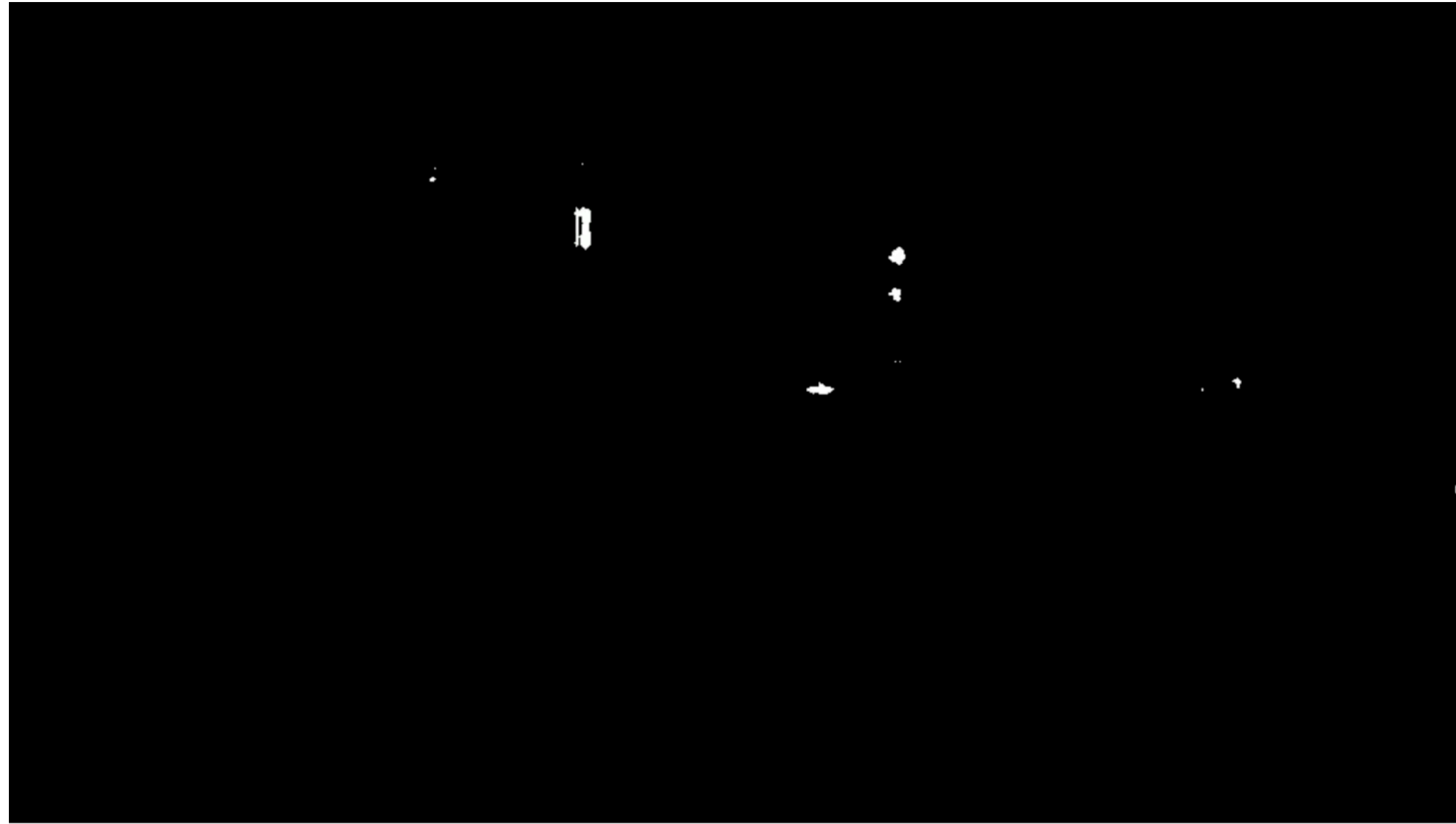
Keypoint Detection



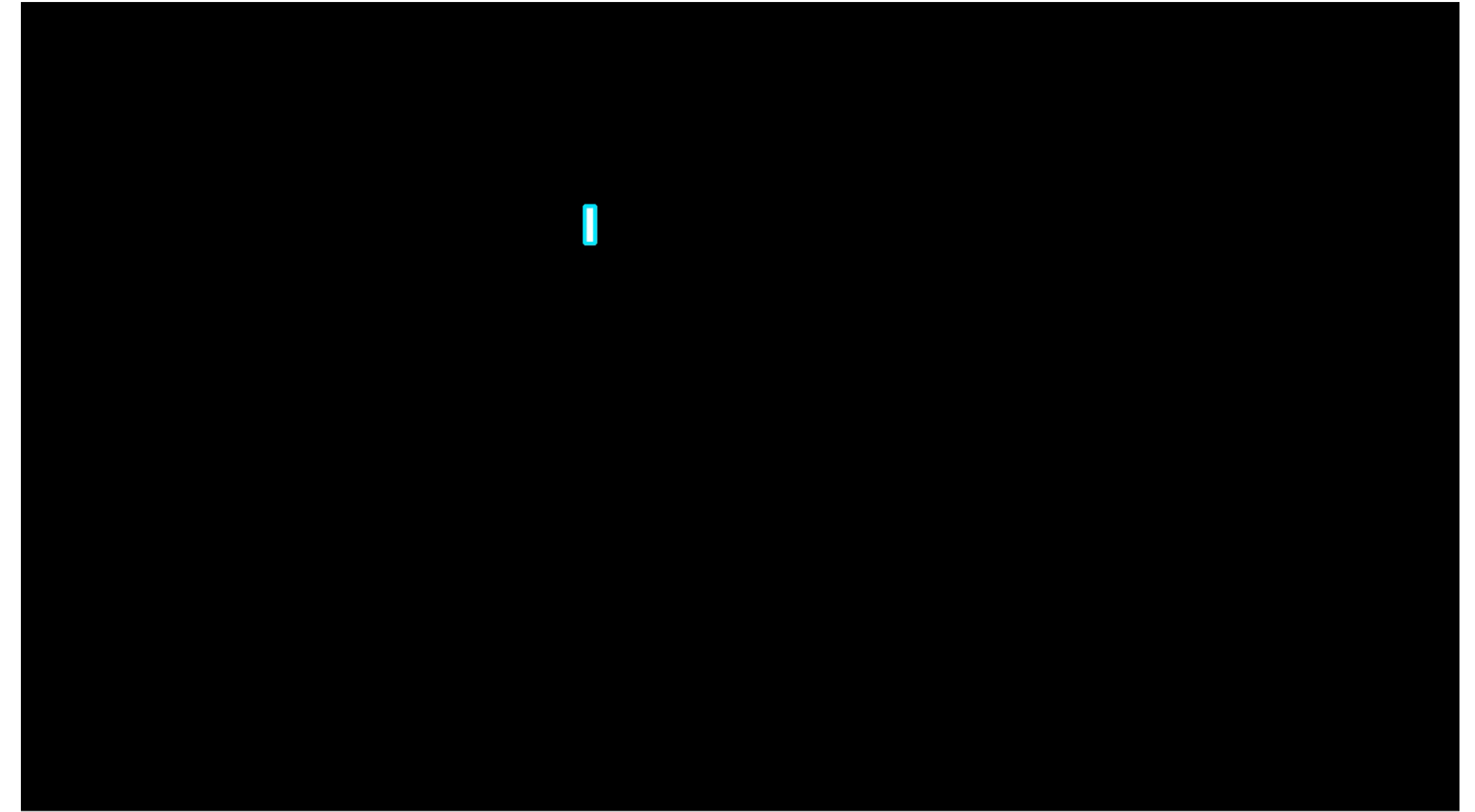
Keypoint Matching



Trajectory Stitching



Foreground (Moving Pixels)



Blobs

*Foreground
Extraction*



*Blob
Extraction*



*Keypoint
Detection*



*Keypoint
Matching*



*Trajectory
Stitching*



*Foreground
Extraction*



*Blob
Extraction*



*Keypoint
Detection*



*Keypoint
Matching*



*Trajectory
Stitching*



*Foreground
Extraction*



*Blob
Extraction*



*Keypoint
Detection*



*Keypoint
Matching*



*Trajectory
Stitching*



*Foreground
Extraction*



*Blob
Extraction*



*Keypoint
Detection*



*Keypoint
Matching*



*Trajectory
Stitching*

Preprocessing

Trajectories of Blobs

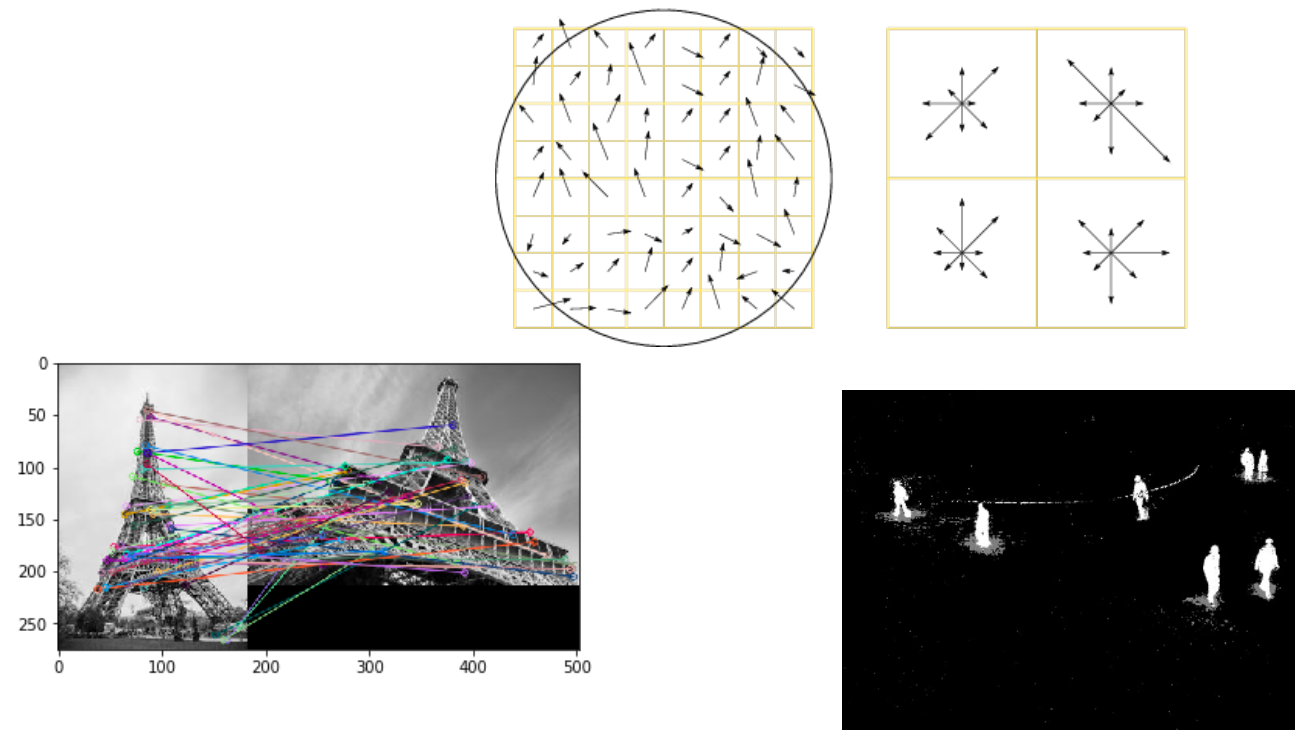
Frame ID	Trajectory ID	x1	y1	x2	y2
1	1	100	200	100	300
1	2	200	600	300	500
1	3	80	120	90	230
2	1	105	205	105	305
...





Need to tune CV techniques conservatively to *comprehensively* extract information!



Boggart's Insight

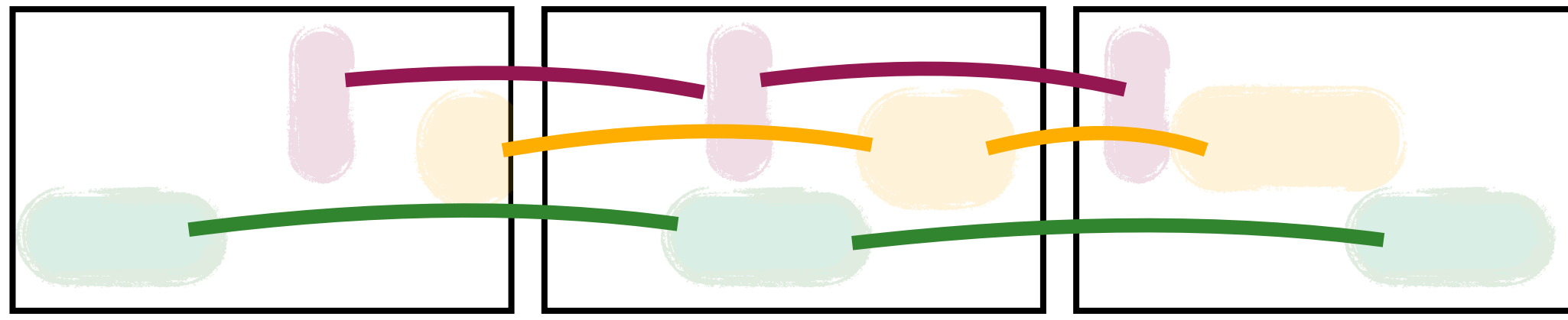


Classical Computer Vision Techniques

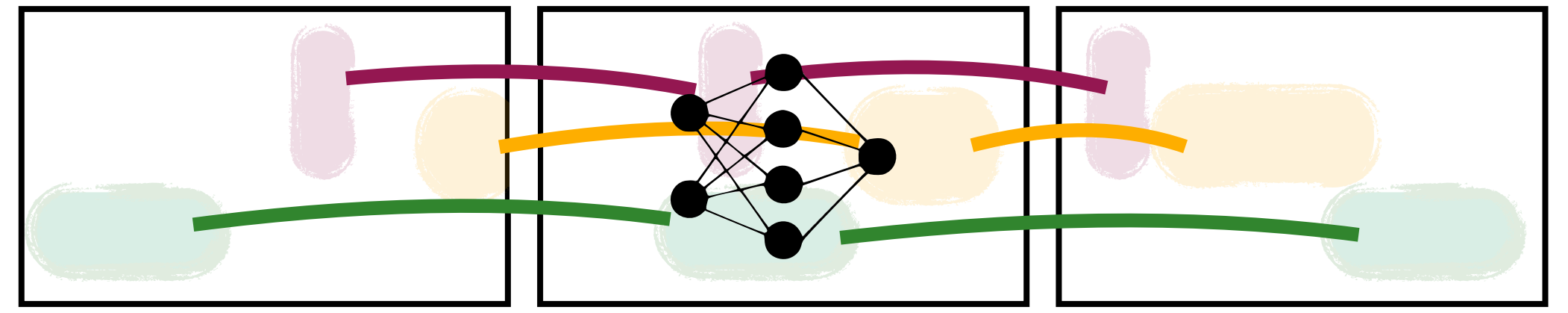
- ▶ Can be leverage to comprehensively & generically extract info from a video 
- ▶ Less accurate than ML 

The image shows four examples of classical computer vision techniques: 1) A circular optical flow field with a grid of arrows. 2) A 2x2 grid of star-shaped feature detectors. 3) A grayscale image of the Eiffel Tower with colored lines connecting corresponding points between two views. 4) A dark image of a scene with white motion vectors overlaid.

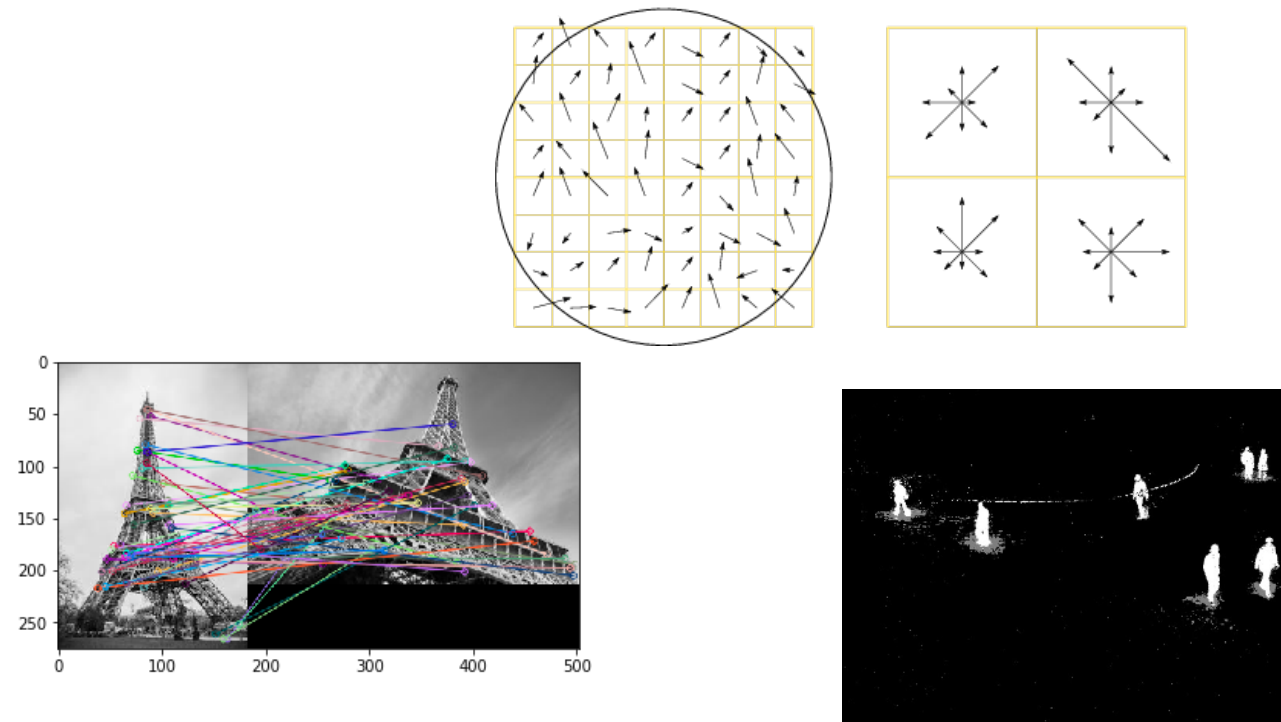
Preprocessing





Query Execution



Boggart's Insight

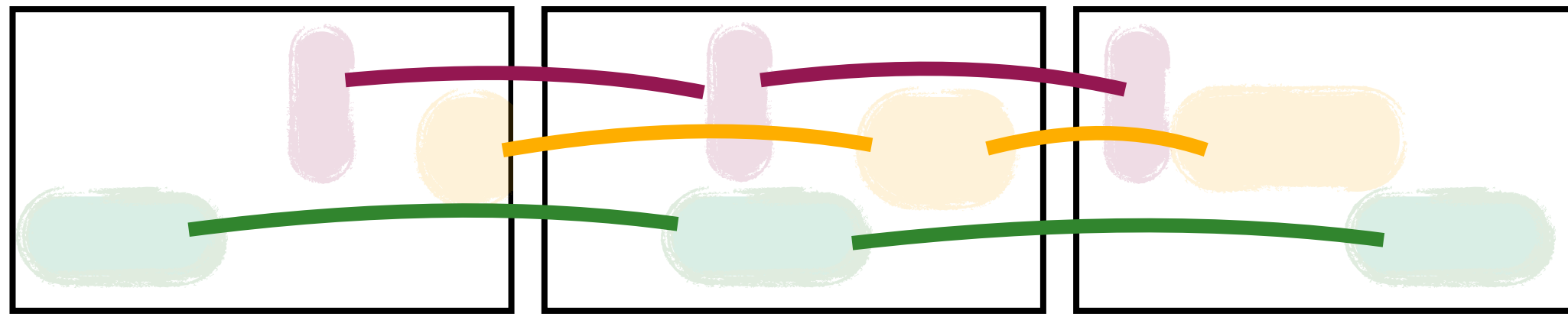


Classical Computer Vision Techniques

- ▶ Can be leverage to comprehensively & generically extract info from a video 
- ▶ Less accurate than ML 

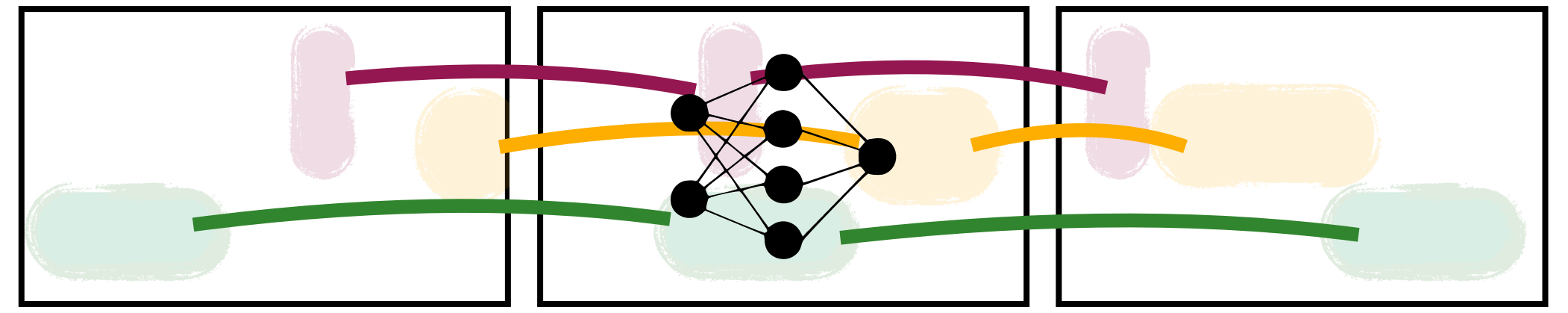
The image shows four examples of classical computer vision techniques: 1) A circular optical flow field with a grid of arrows. 2) A 2x2 grid of starburst patterns. 3) A grayscale image of the Eiffel Tower with a coordinate system (0-500 on x-axis, 0-250 on y-axis) and multi-colored lines representing object trajectories. 4) A dark image of a scene with white motion vectors overlaid on it.

Preprocessing



Trajectories of areas of motion

Query Execution



Model-specific labeling & propagation

Query Execution

Idea: run model on as few frames as possible and use trajectories to propagate model results to the remaining frames

Query Execution

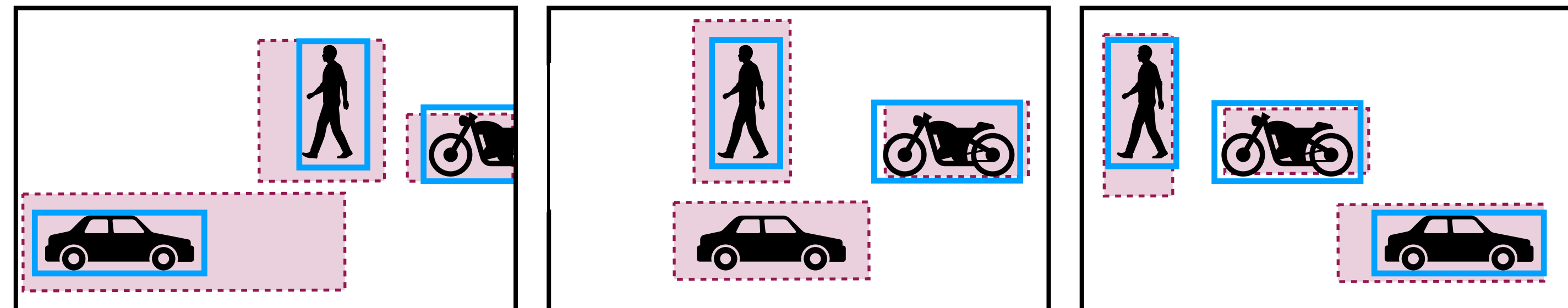
Idea: run model on as few frames as possible and use trajectories to propagate model results to the remaining frames

Challenge: misalignment of blobs with ML model output

Query Execution

Idea: run model on as few frames as possible and use trajectories to propagate model results to the remaining frames

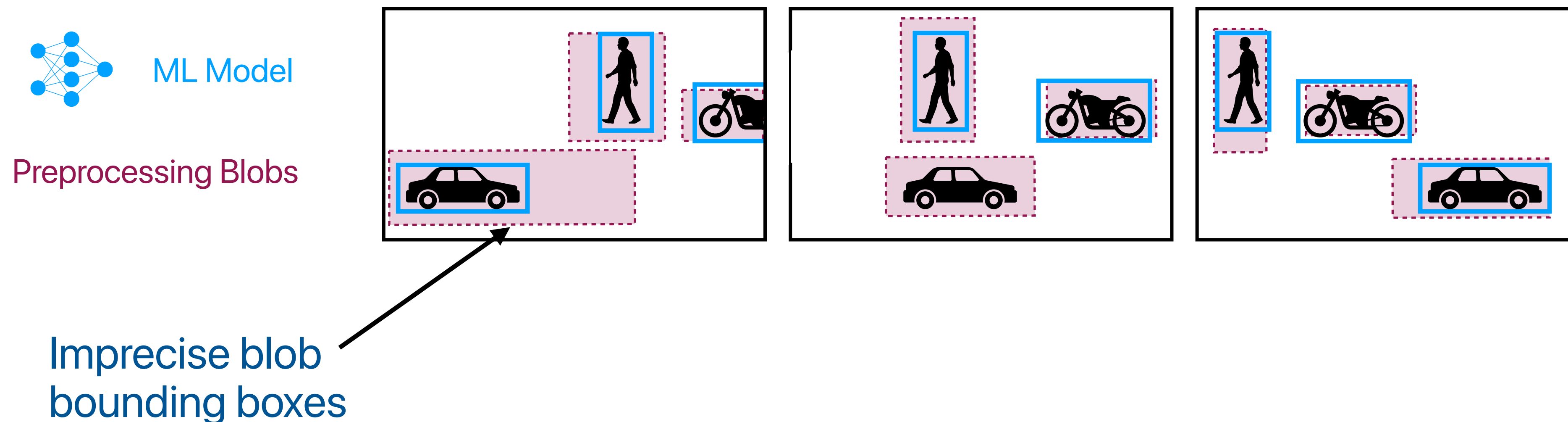
Challenge: misalignment of blobs with ML model output



Query Execution

Idea: run model on as few frames as possible and use trajectories to propagate model results to the remaining frames

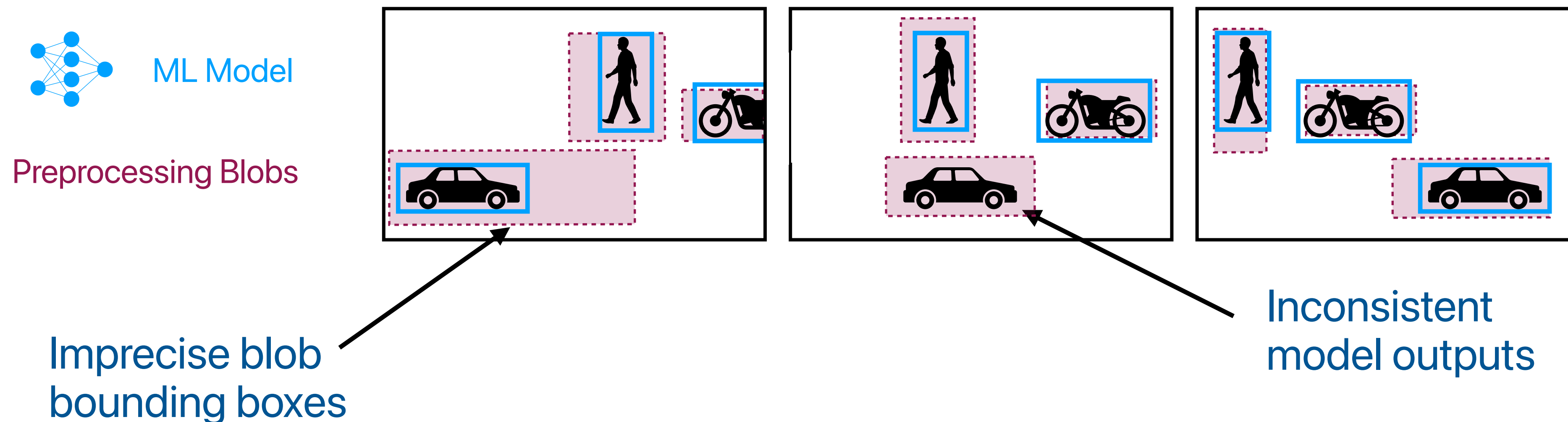
Challenge: misalignment of blobs with ML model output



Query Execution

Idea: run model on as few frames as possible and use trajectories to propagate model results to the remaining frames

Challenge: misalignment of blobs with ML model output



Query Execution: New Techniques

Query Execution: New Techniques

1

Identify the smallest set of frames on which to run the model

Query Execution: New Techniques

1

Identify the smallest set of frames on which to run the model

2

Correct imprecisions during model result propagation across the remaining frames

Query Execution: New Techniques

1

Identify the smallest set of frames on which to run the model

2

Correct imprecisions during model result propagation across the remaining frames

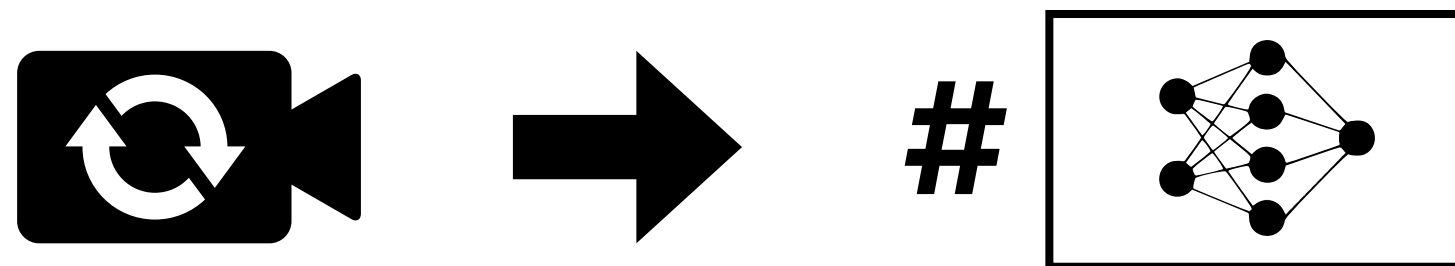
Query Execution: New Techniques

1

Identify the smallest set of frames on which to run the model

2

Correct imprecisions during model result propagation across the remaining frames



of frames on which to run the model
is influenced by video dynamism

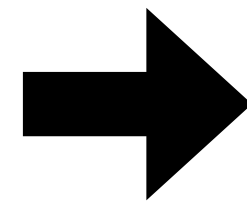
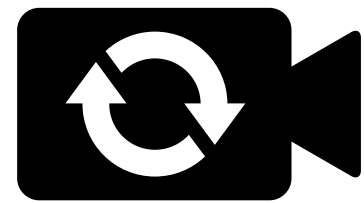
Query Execution: New Techniques

1

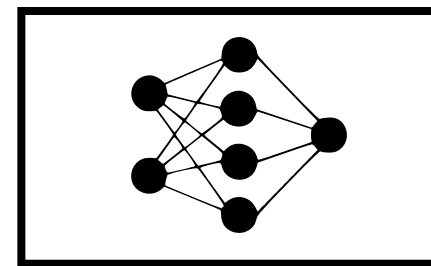
Identify the smallest set of frames on which to run the model

2

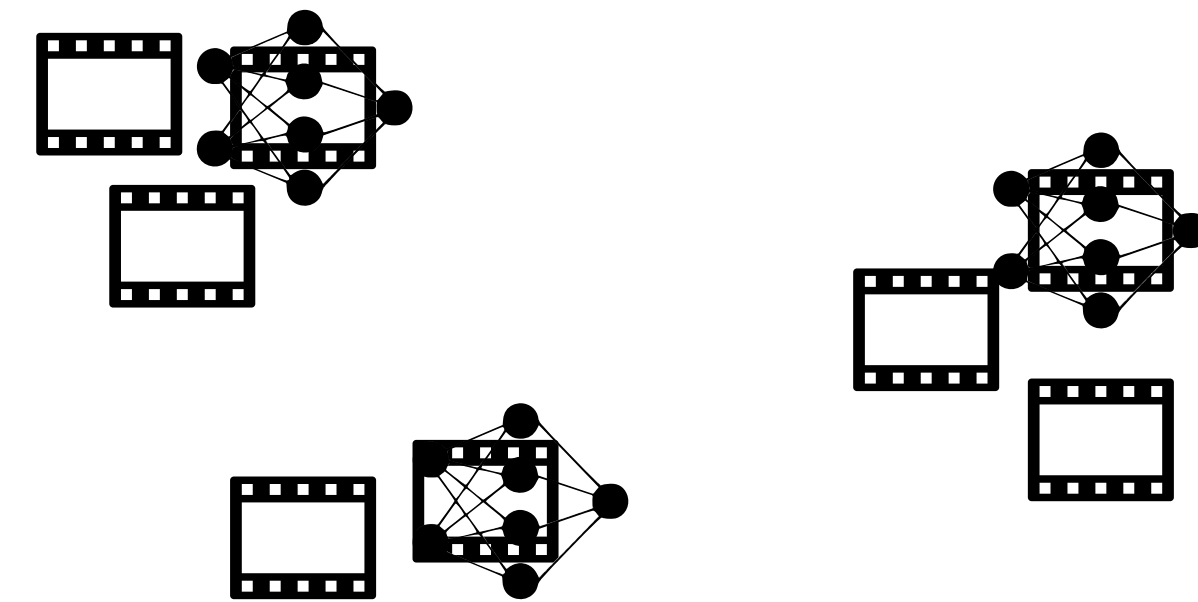
Correct imprecisions during model result propagation across the remaining frames



#



of frames on which to run the model is influenced by video dynamism



Cluster similar video segments and profile a small portion of each cluster

Query Execution: New Techniques

1

Identify the smallest set of frames on which to run the model

2

Correct imprecisions during model result propagation across the remaining frames

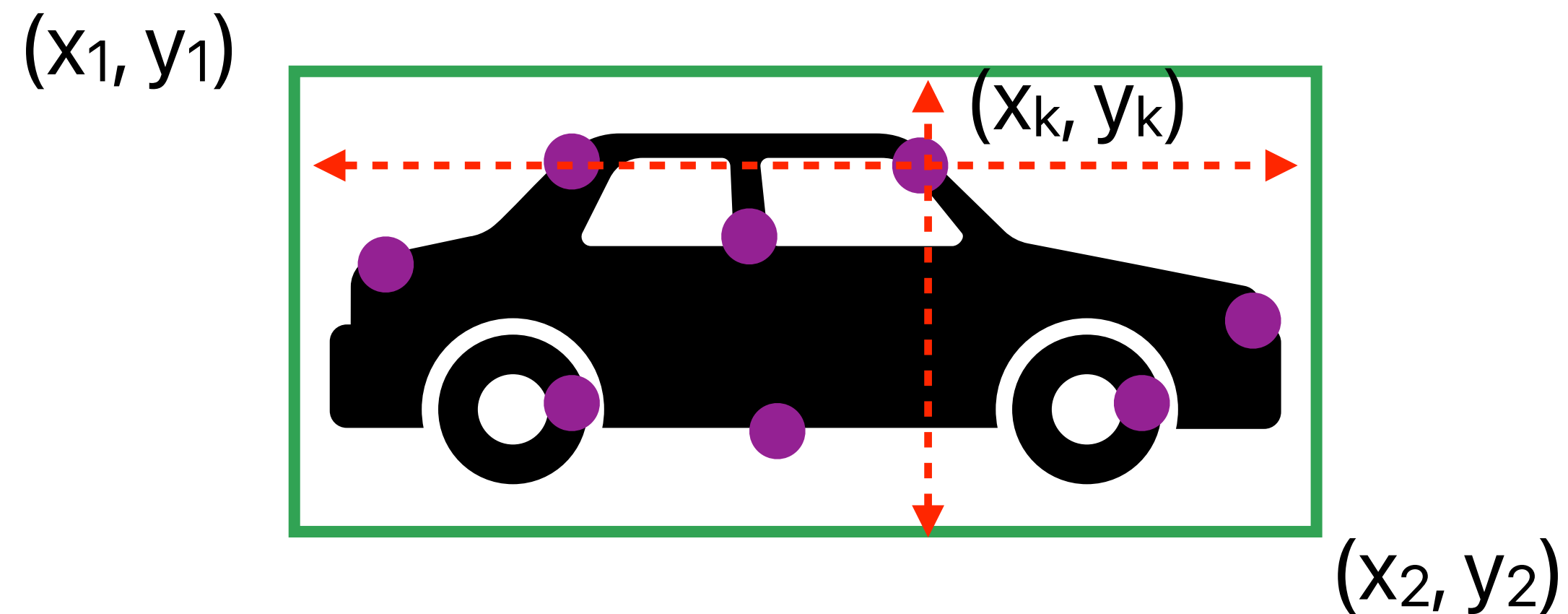
Query Execution: New Techniques

1

Identify the smallest set of frames on which to run the model

2

Correct imprecisions during model result propagation across the remaining frames



Relative position between an object's keypoints and its bounding boxes remain stable over time

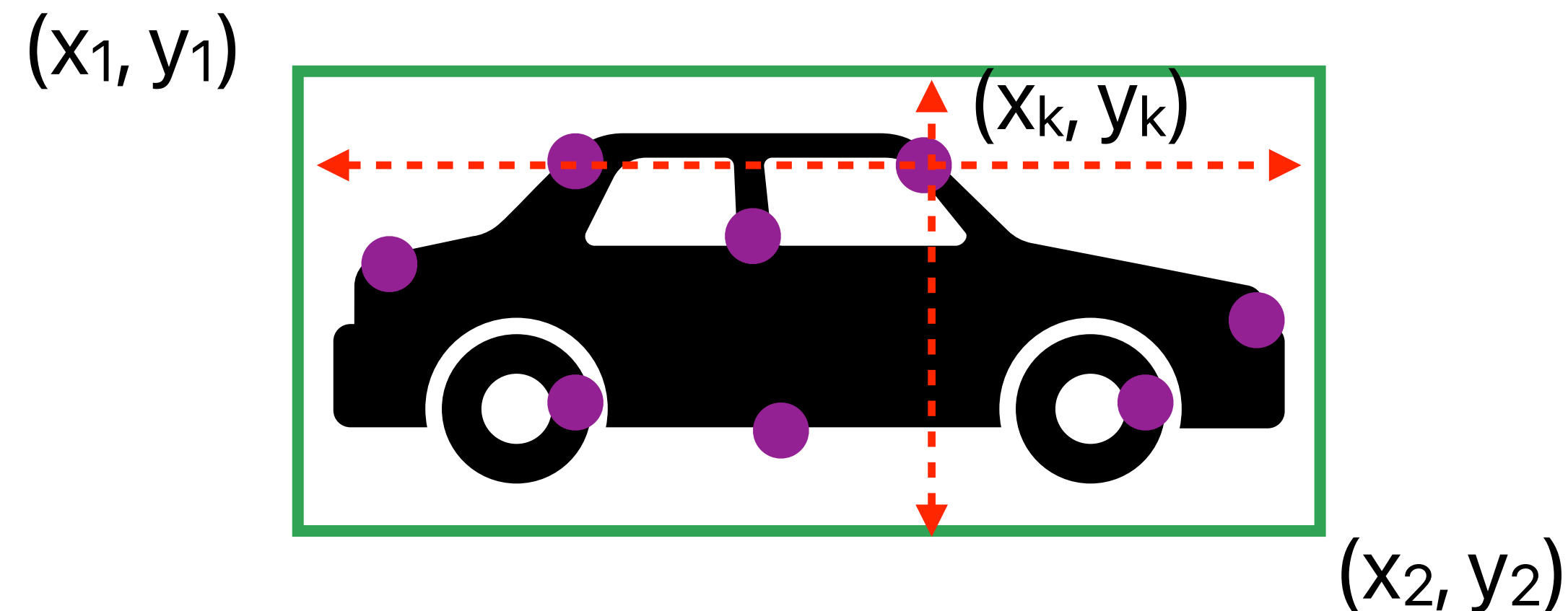
Query Execution: New Techniques

1

Identify the smallest set of frames on which to run the model

2

Correct imprecisions during model result propagation across the remaining frames



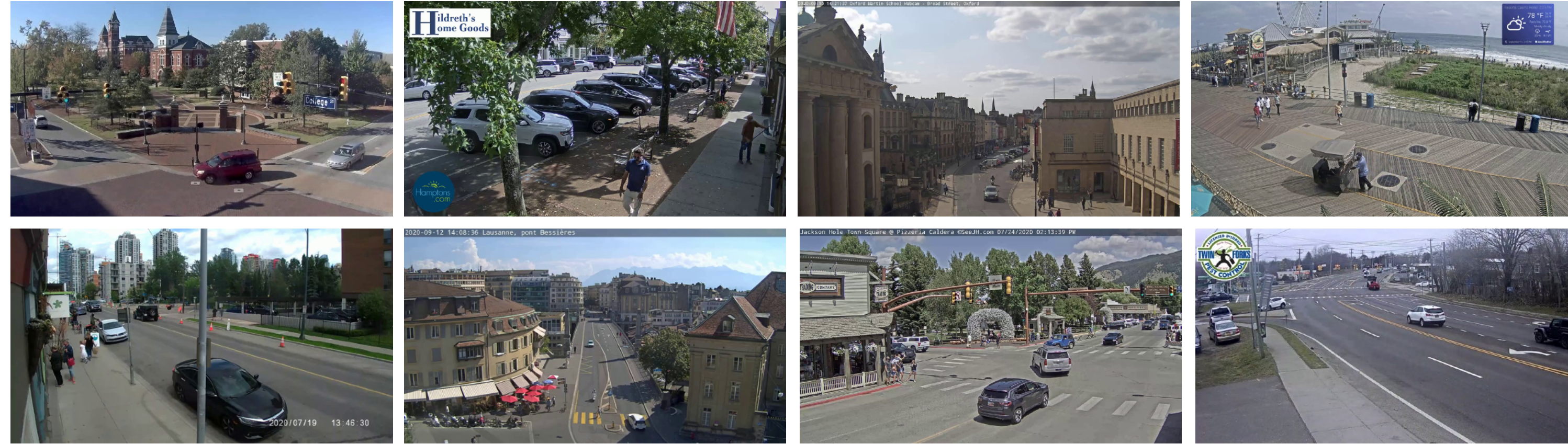
Relative position between an object's keypoints and its bounding boxes remain stable over time

$$(ax_k, ay_k) = \left(\frac{x_2 - x_k}{x_2 - x_1}, \frac{y_2 - y_k}{y_2 - y_1} \right)$$

$$\sum_{k'} \left[\left(\frac{x_2 - x_{k'}}{x_2 - x_1} - ax_k \right)^2 + \left(\frac{y_2 - y_{k'}}{y_2 - y_1} - ay_k \right)^2 \right]$$

Search for blob coordinates that maximally preserve these relationships

Evaluation Methodology



96 hours of publicly available camera footage

Query Types: binary classification, counting, bounding box detection

Accuracy Targets: 80%, 90%, 95%

Objects of interest: cars & people

Query Models: 3 architectures, each trained on 2 datasets

Evaluation Axes

- ▶ Query-execution speedups
- ▶ Comparison to existing systems
- ▶ Performance on downsampled video
- ▶ Resource scaling
- ▶ Storage costs
- ▶ Parameter sensitivity
- ▶ Generalizability

Evaluation Axes

- ▶ Query-execution speedups

- ▶ Comparison to existing systems

- ▶ Performance on downsampled video

- ▶ Resource scaling

- ▶ Storage costs

- ▶ Parameter sensitivity

- ▶ Generalizability

Query Execution Speedups

Baseline: run query model on every frame

Query Execution Speedups

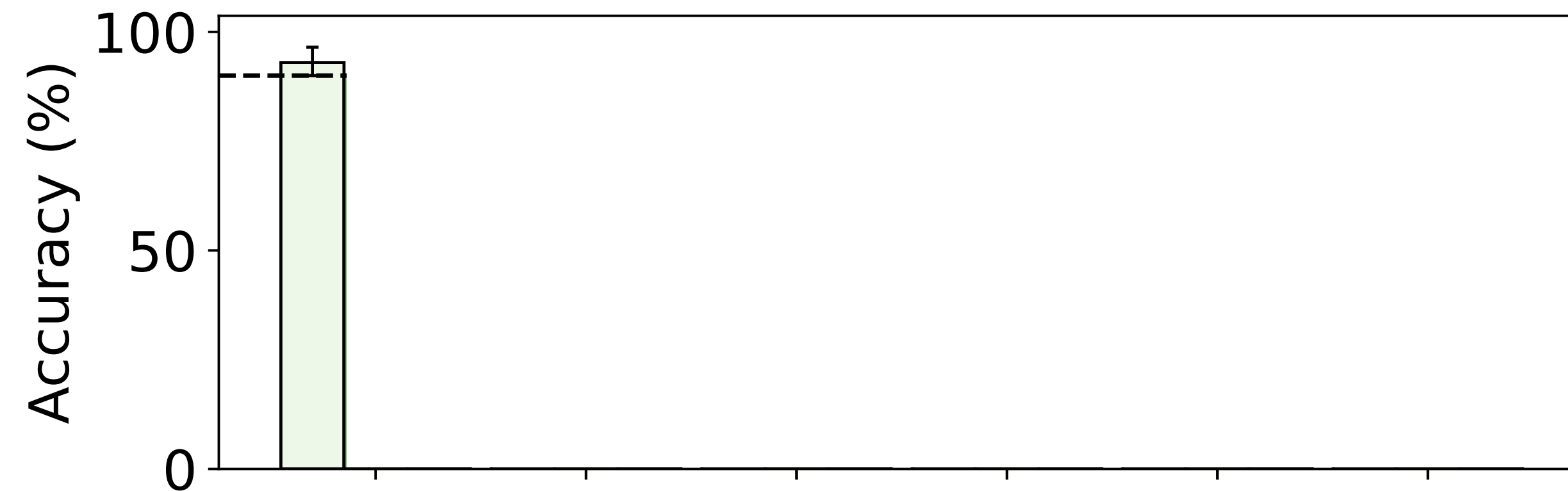
Baseline: run query model on every frame

Query:

- Model: YOLOv3+COCO
- Accuracy Target: 90%
- Query Type: Binary Classification

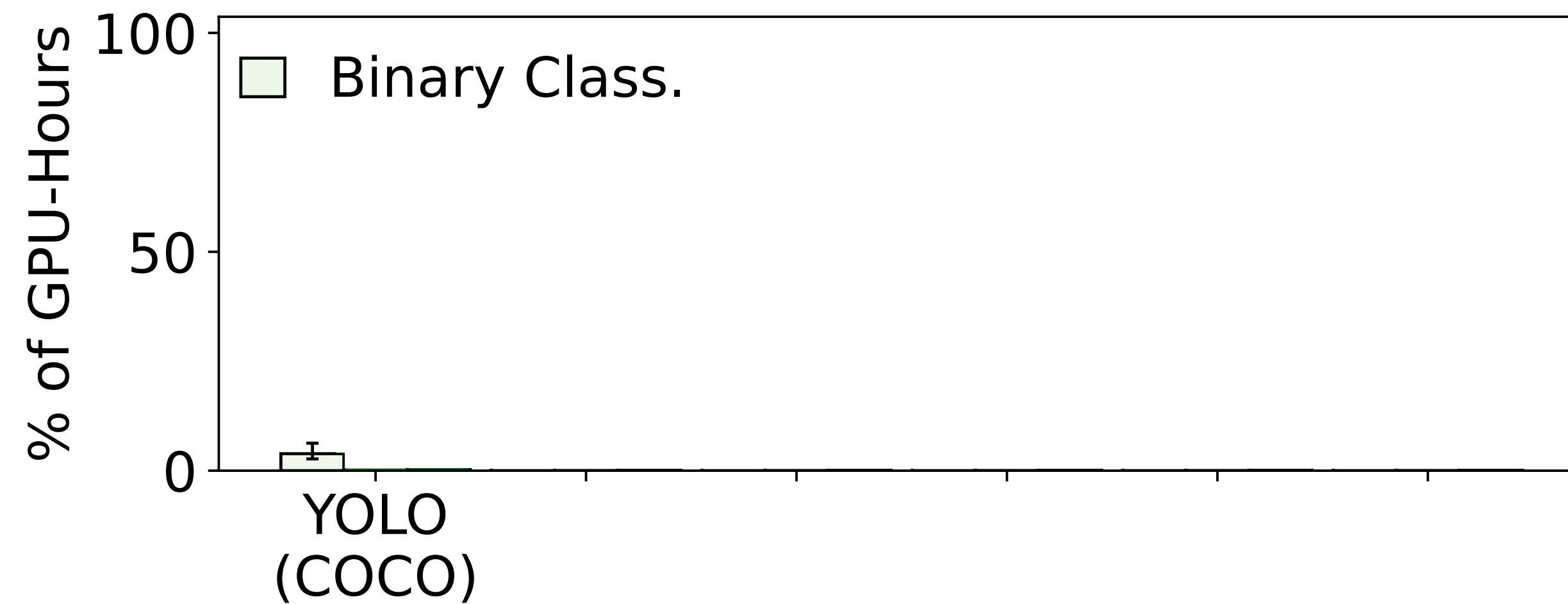
Query Execution Speedups

Baseline: run query model on every frame



Query:

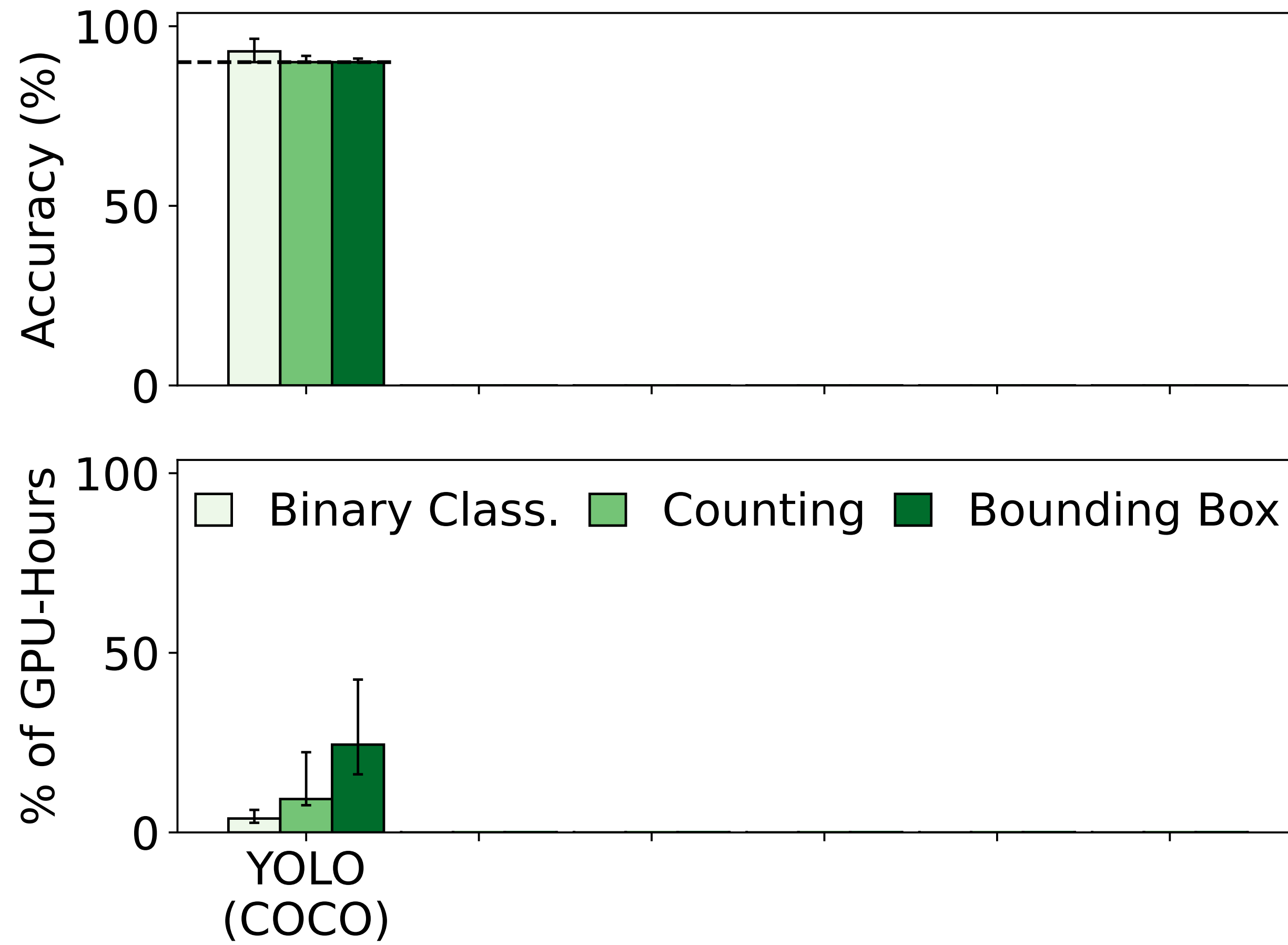
- Model: YOLOv3+COCO
- Accuracy Target: 90%
- Query Type: Binary Classification



Result: Boggart returned results that achieved an accuracy of 93% while requiring the query model to be run on only 5% of the total frames

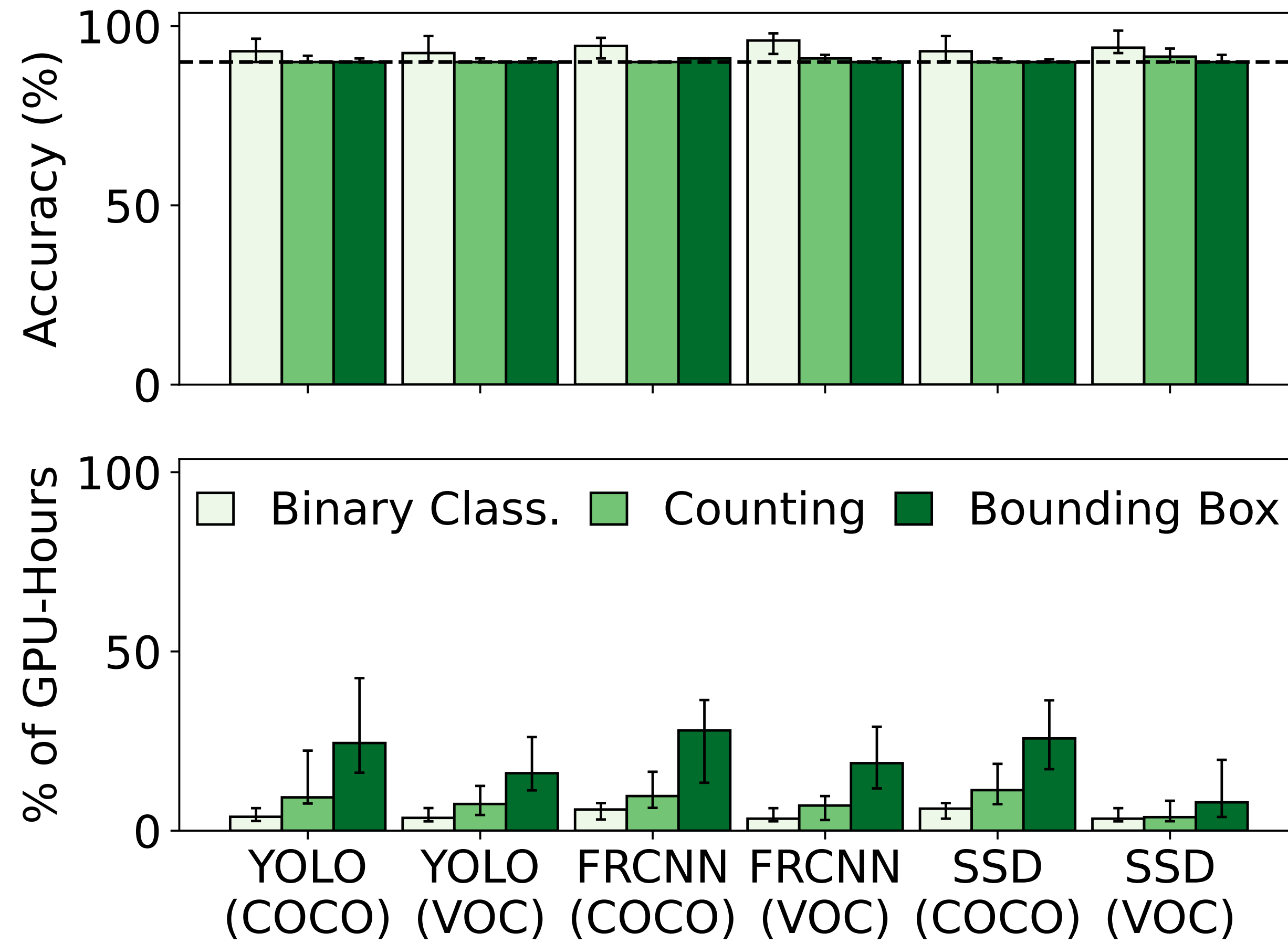
Query Execution Speedups

Baseline: run query model on every frame



Query Execution Speedups

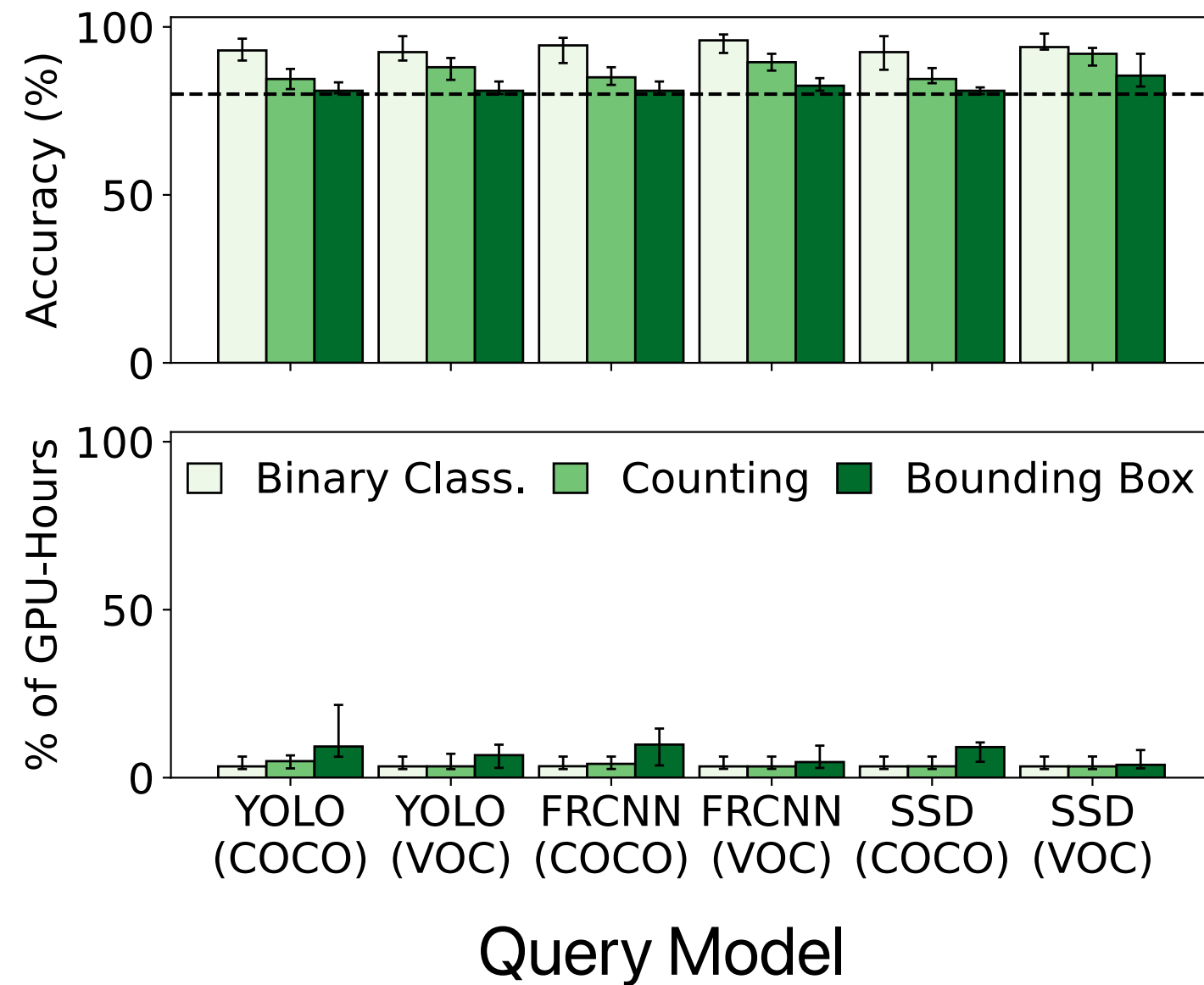
Baseline: run query model on every frame



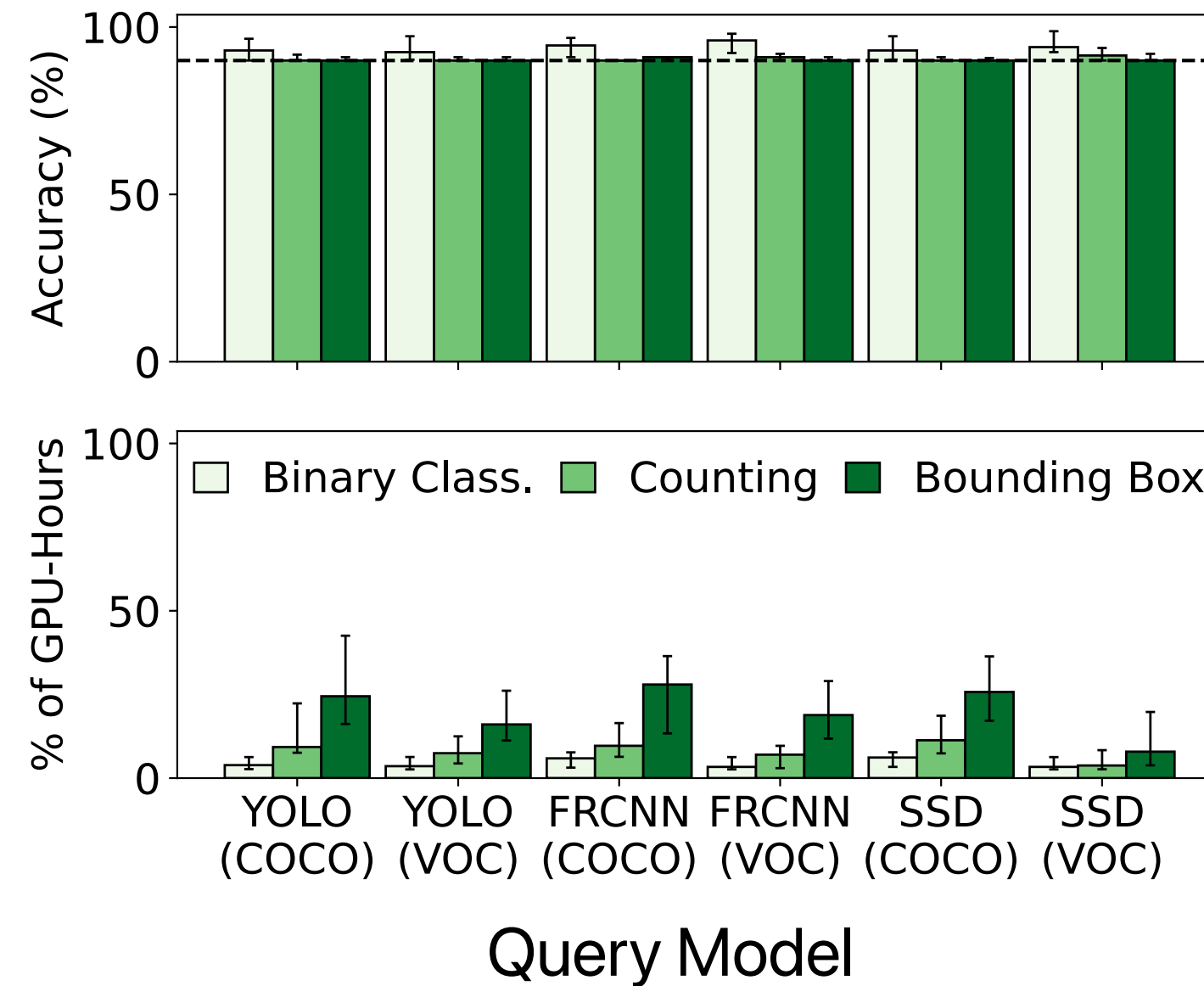
Query Execution Speedups

Baseline: run query model on every frame

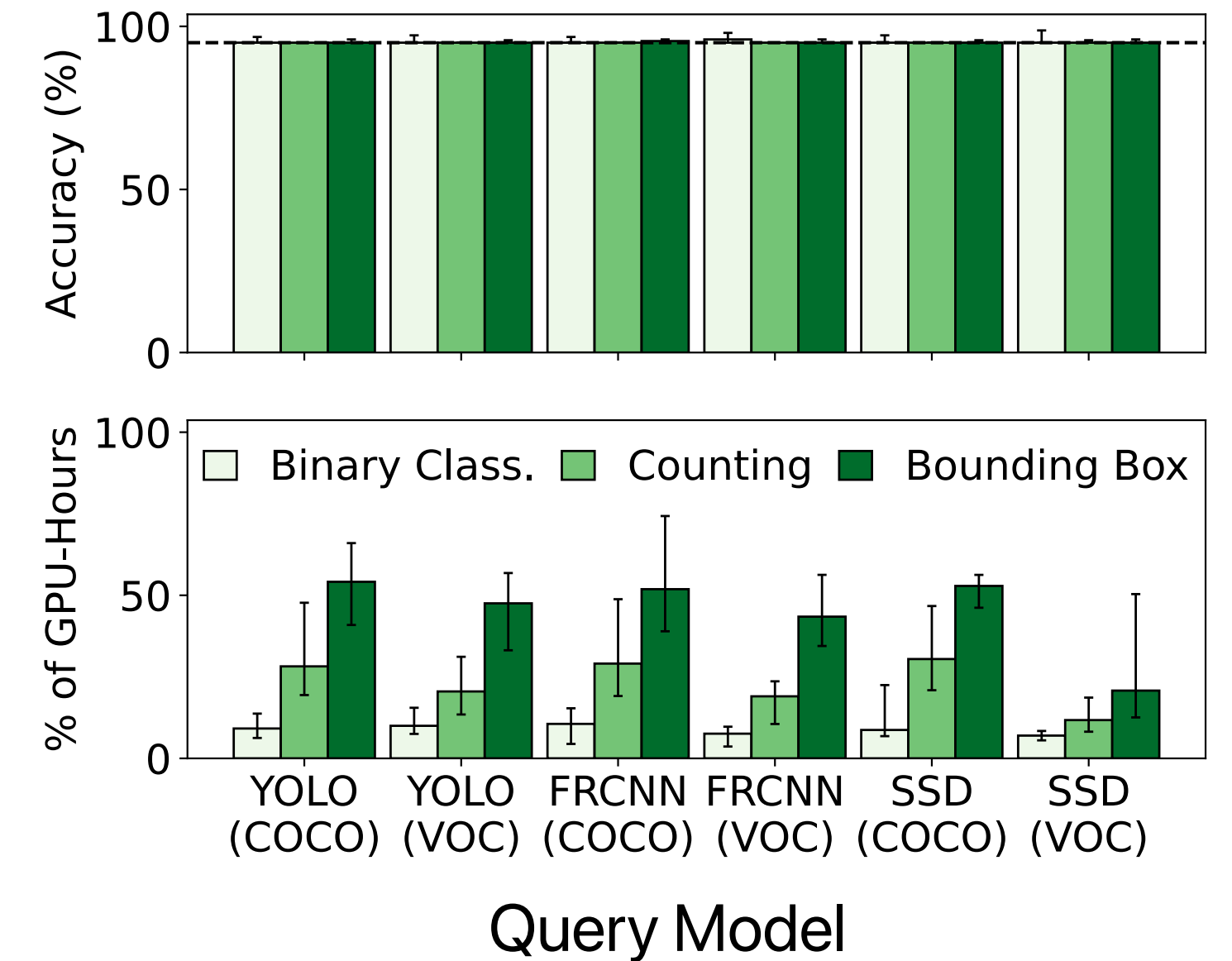
Accuracy Target: 80%



Accuracy Target: 90%



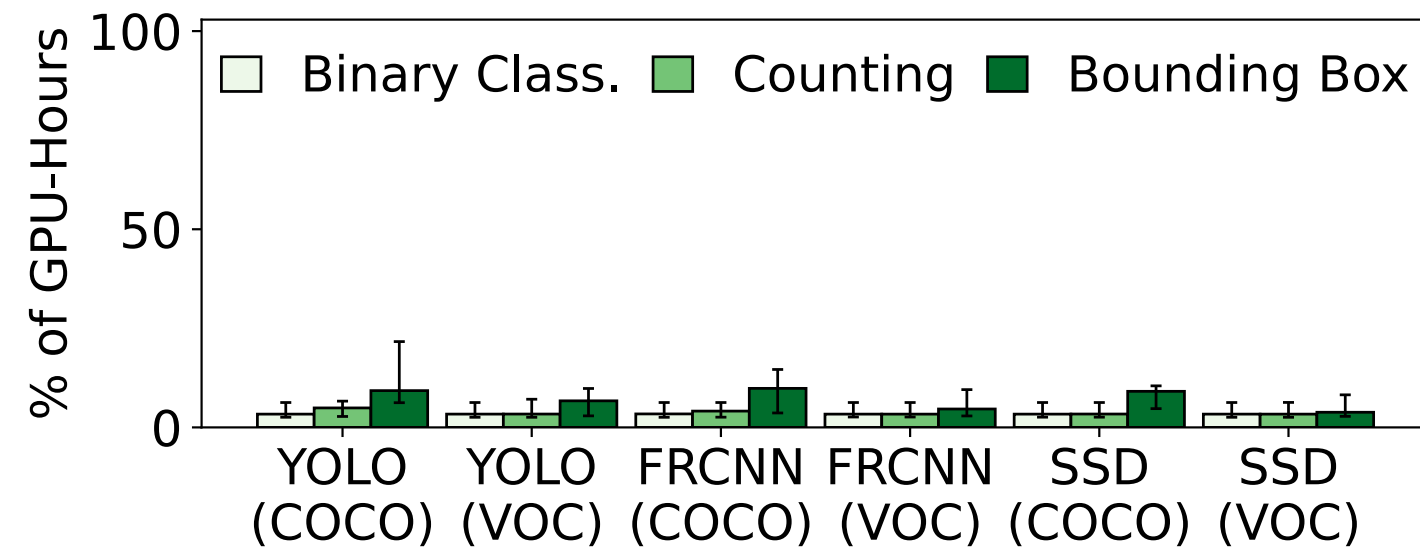
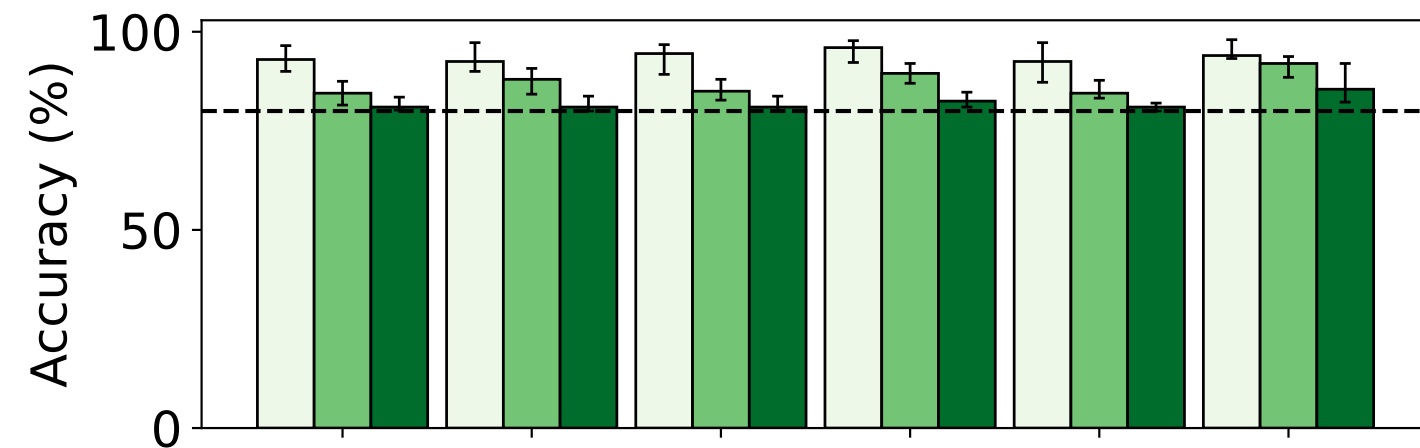
Accuracy Target: 95%



Query Execution Speedups

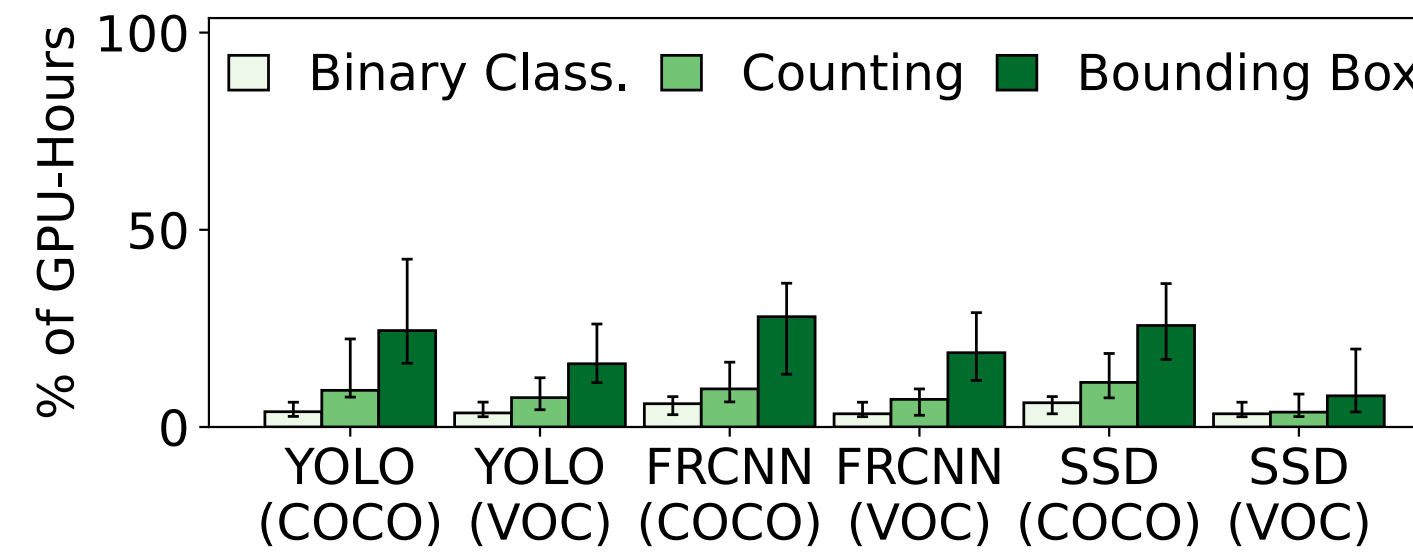
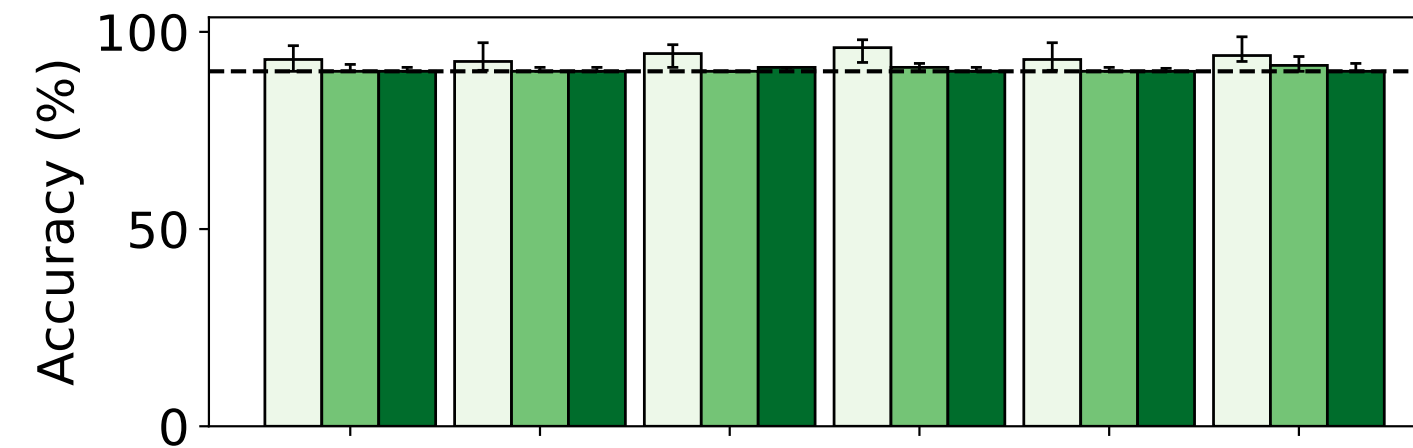
Baseline: run query model on every frame

Accuracy Target: 80%



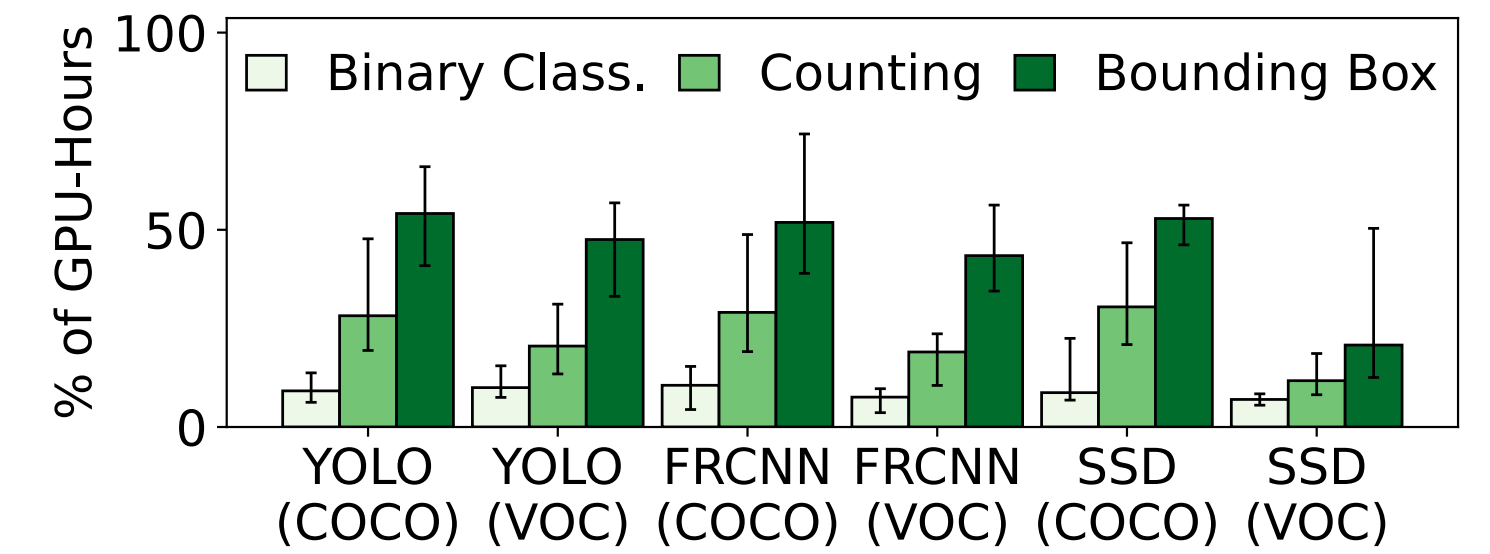
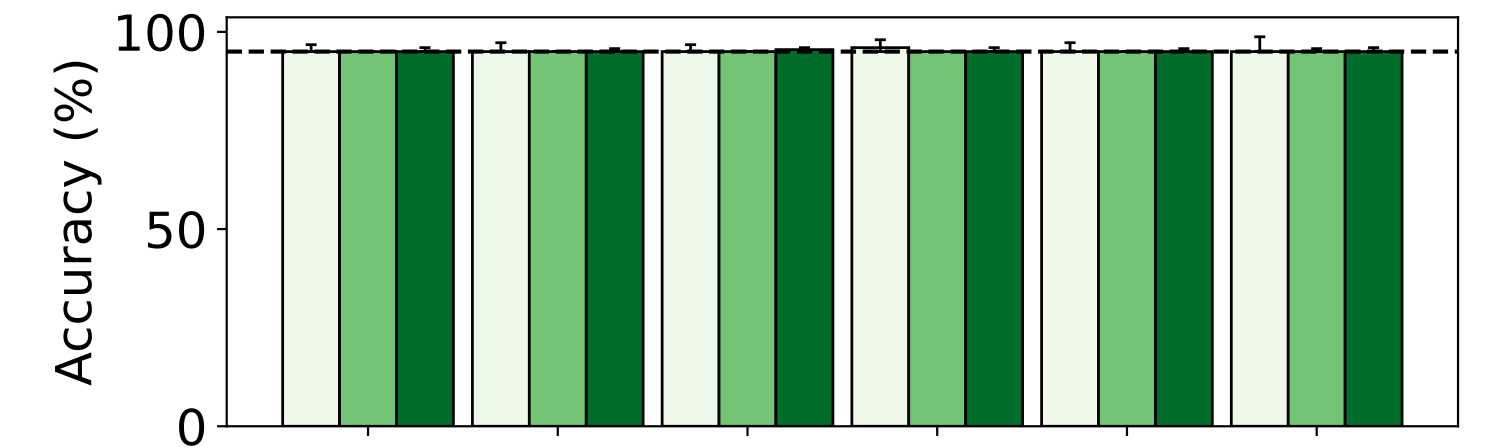
Query Model

Accuracy Target: 90%



Query Model

Accuracy Target: 95%



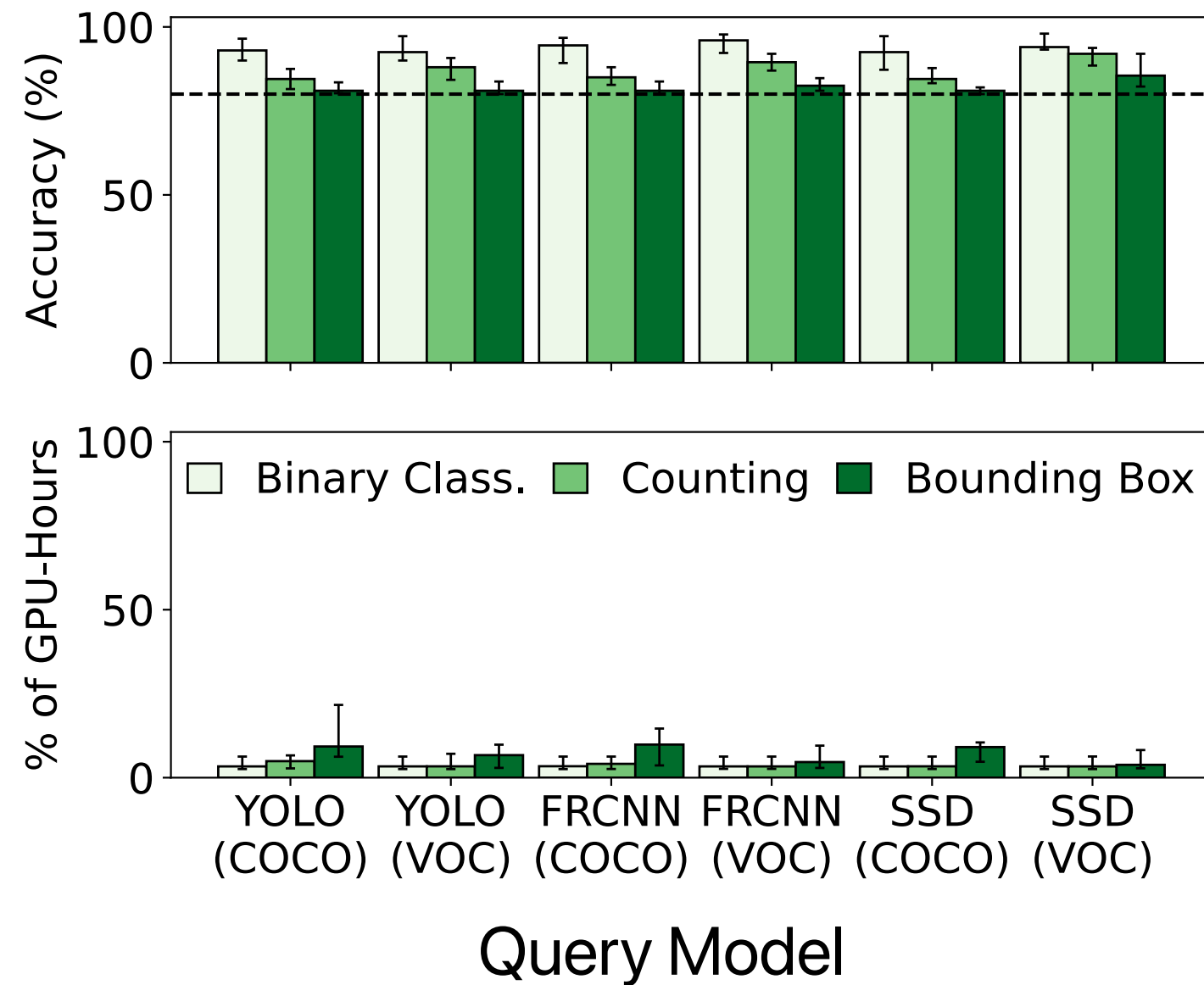
Query Model

Boggart consistently meets specified accuracy targets while requiring a fraction of the compute!

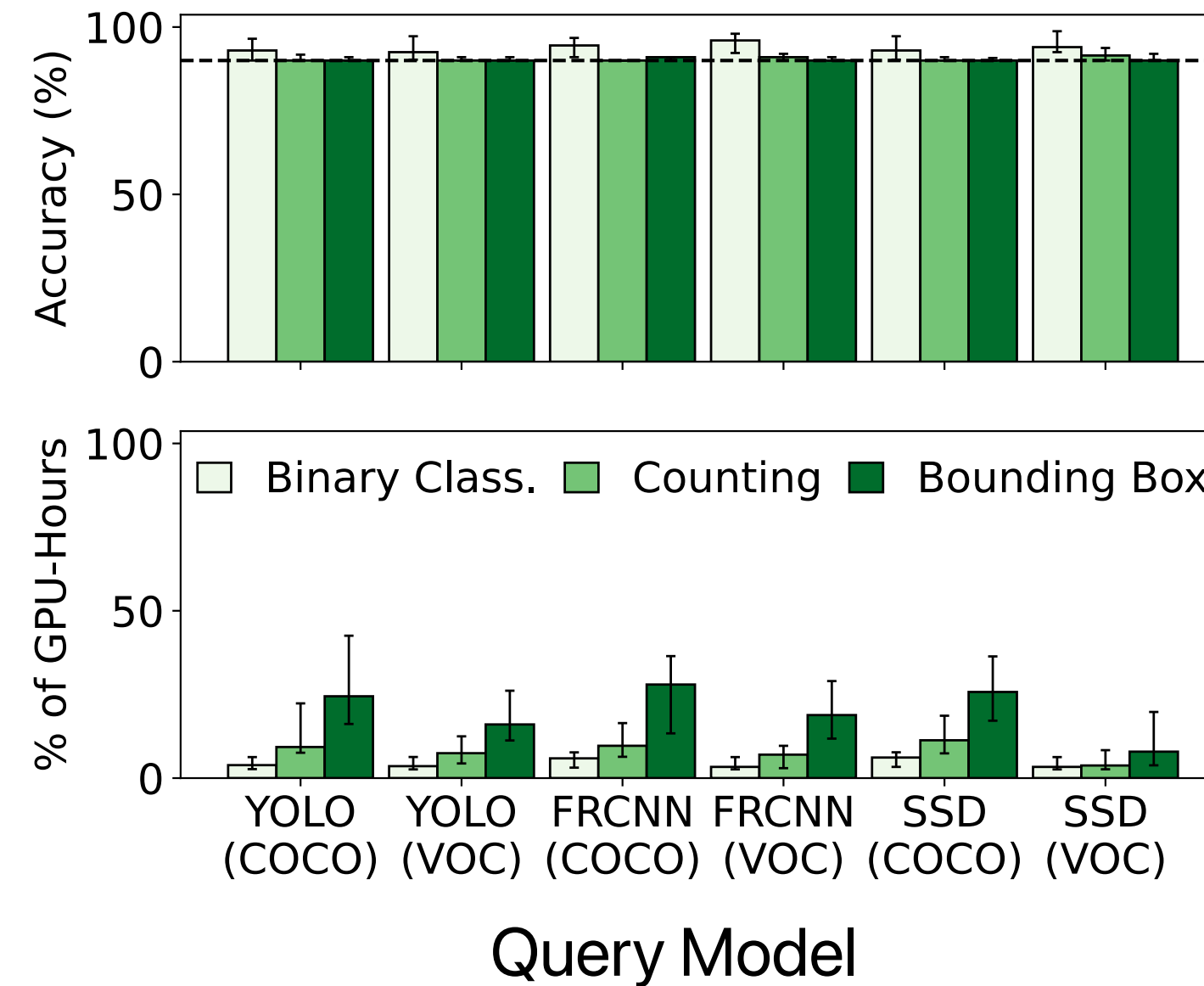
Query Execution Speedups

Baseline: run query model on every frame

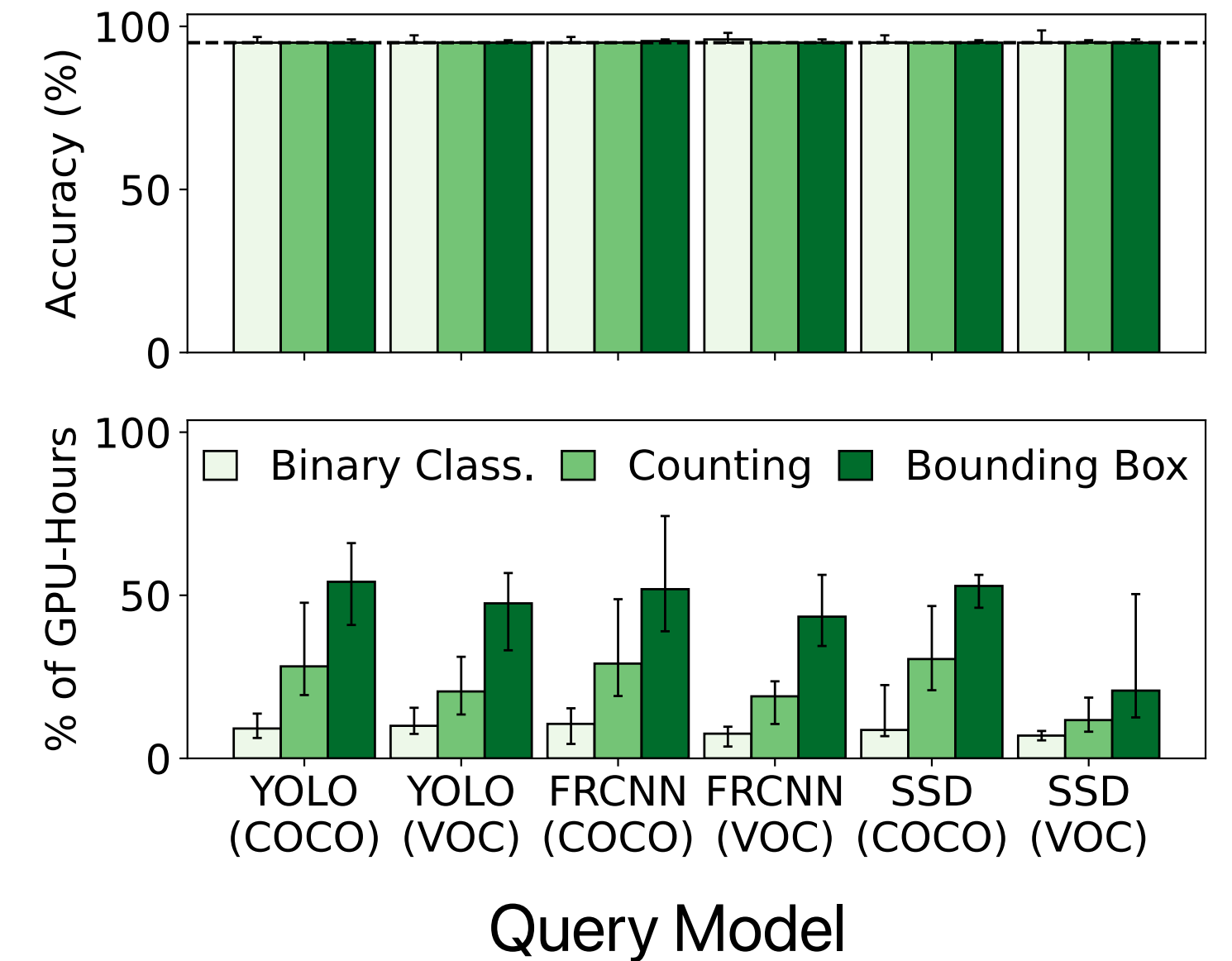
Accuracy Target: 80%



Accuracy Target: 90%



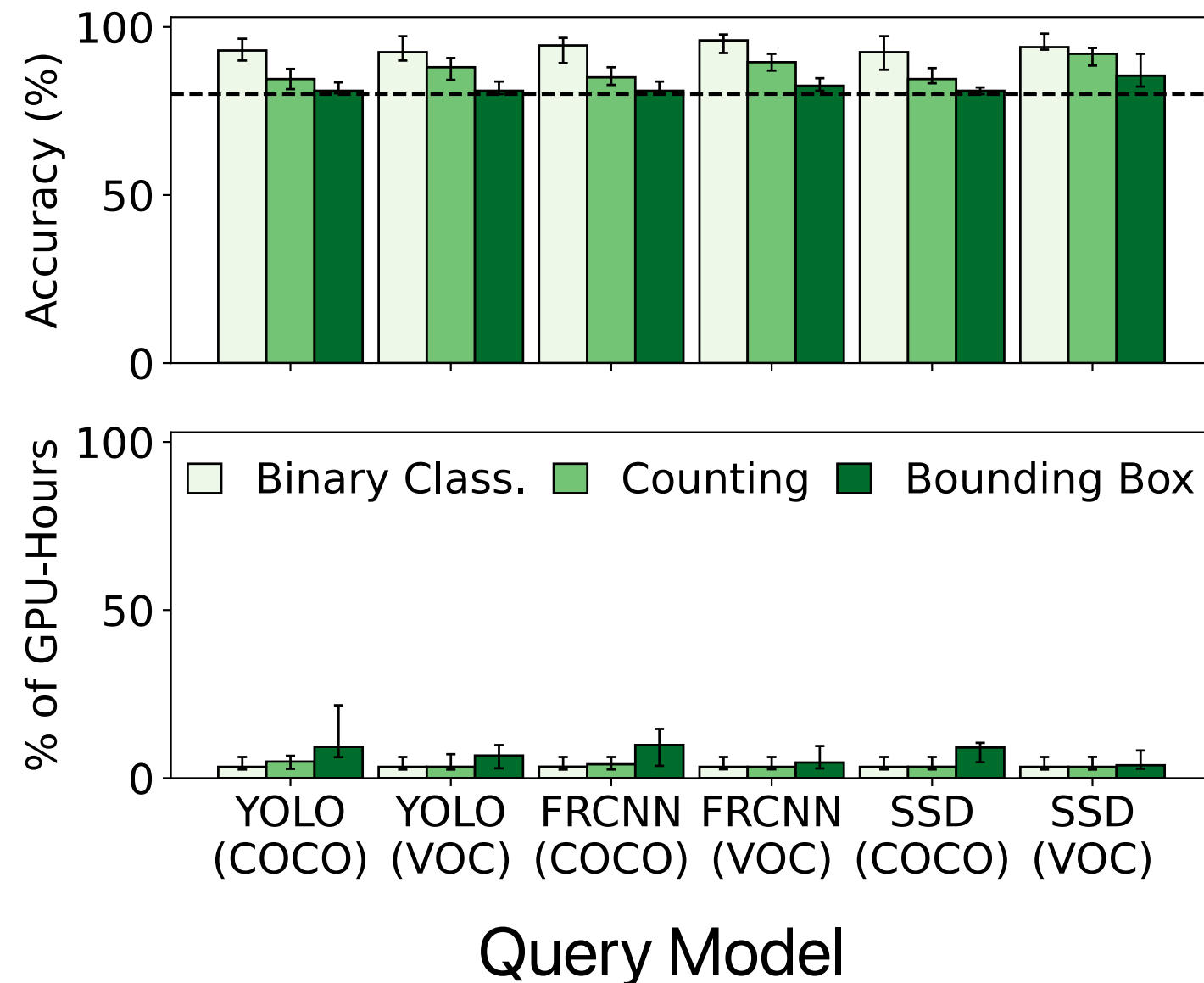
Accuracy Target: 95%



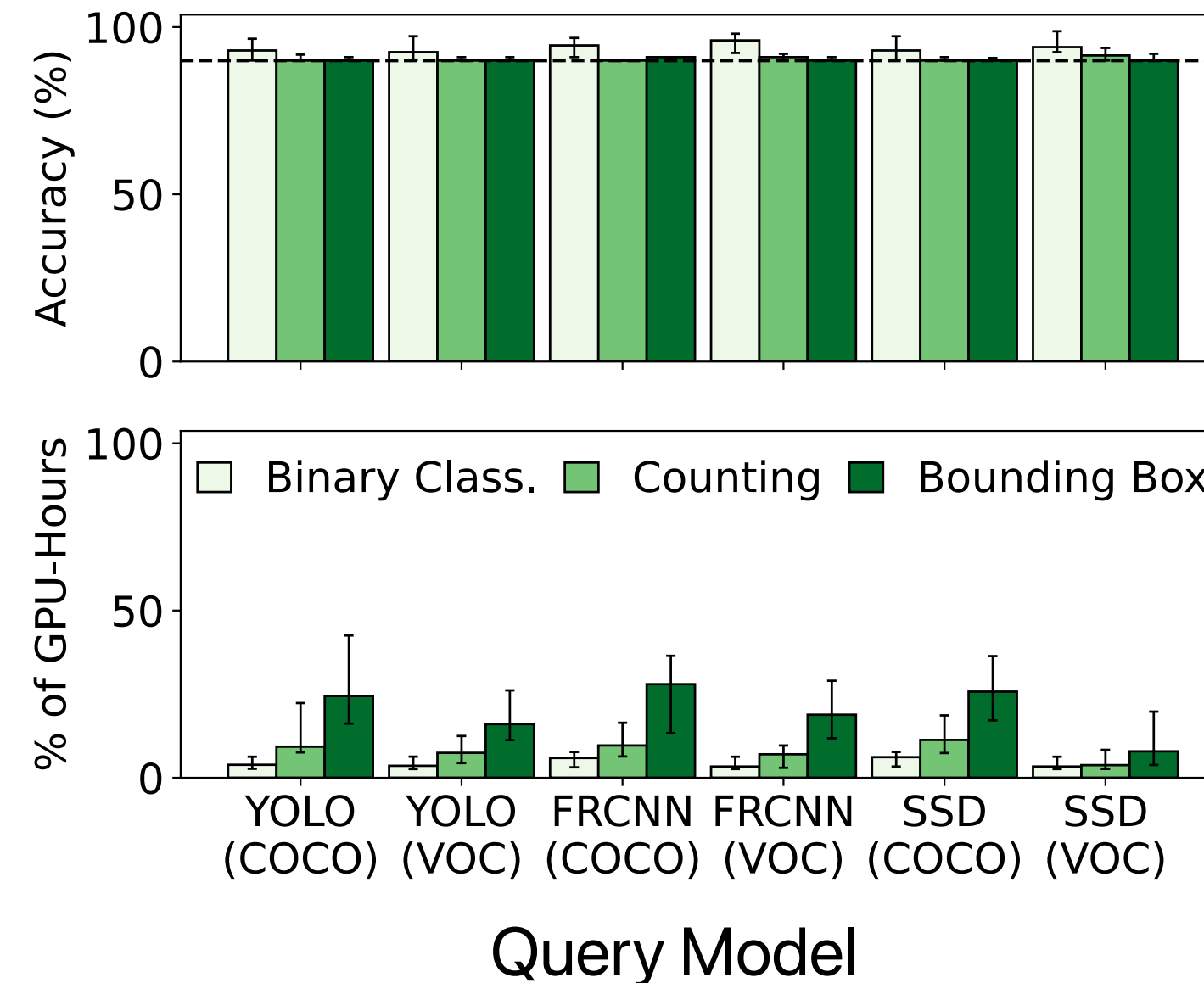
Query Execution Speedups

Baseline: run query model on every frame

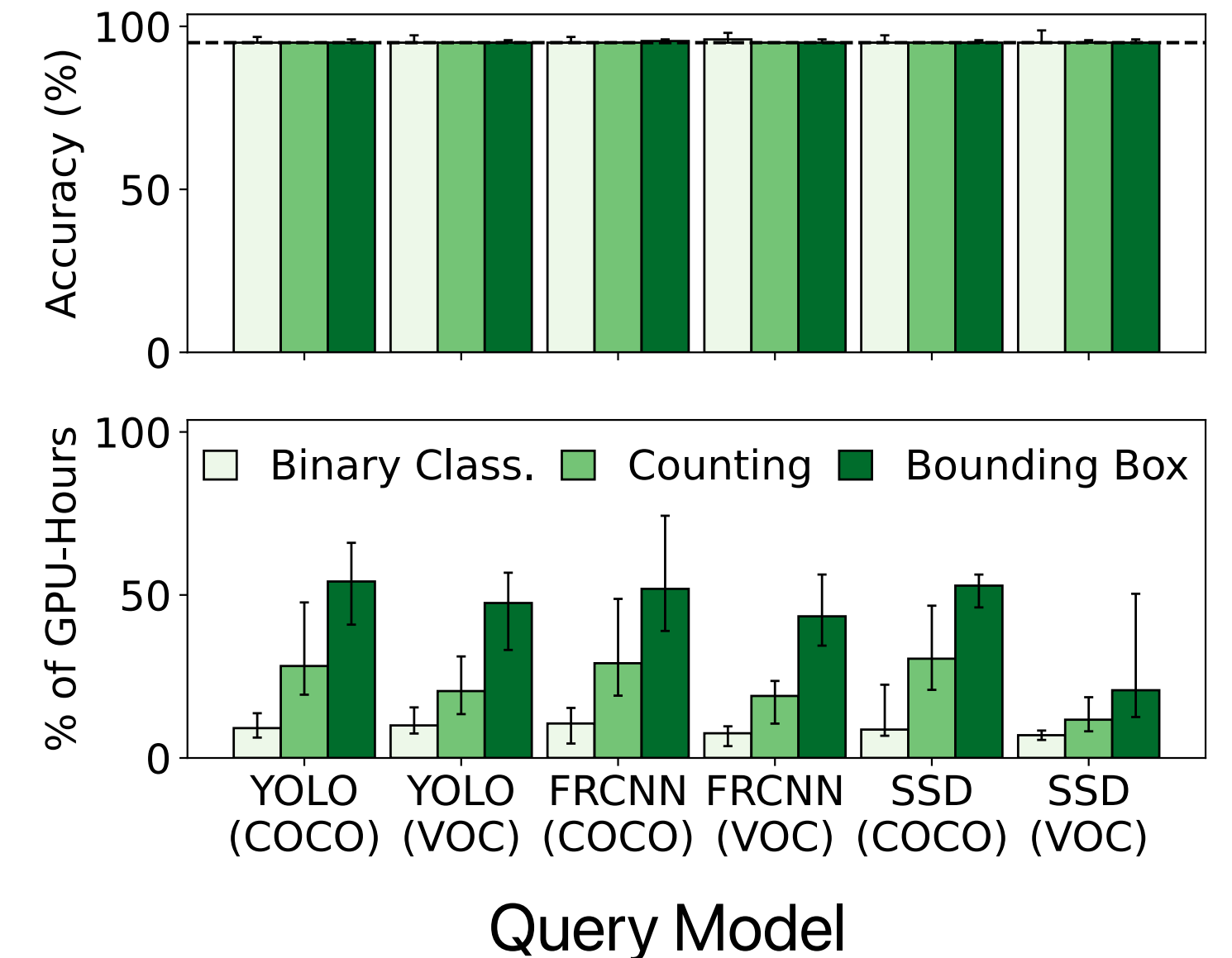
Accuracy Target: 80%



Accuracy Target: 90%

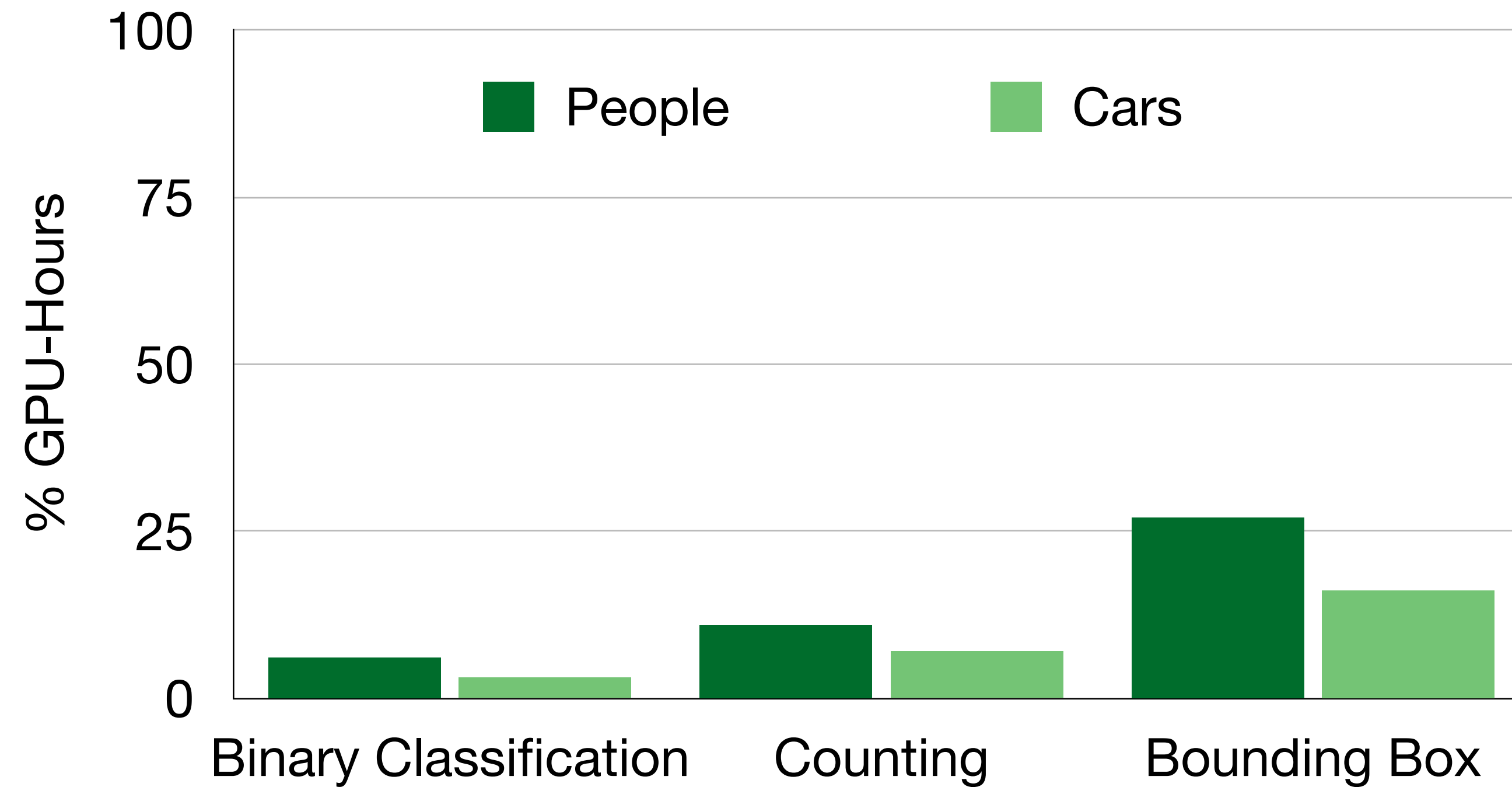


Accuracy Target: 95%



Finer-grained queries and higher accuracy targets -> Run query model on more frames

Different Object Types — People vs. Cars



Querying for **people** requires **more model inference** than querying for **cars**.

Comparison to Model-Specific Preprocessing

Comparison to Model-Specific Preprocessing

Focus (OSDI '18) leverages model-specific preprocessing to accelerate binary classification queries.

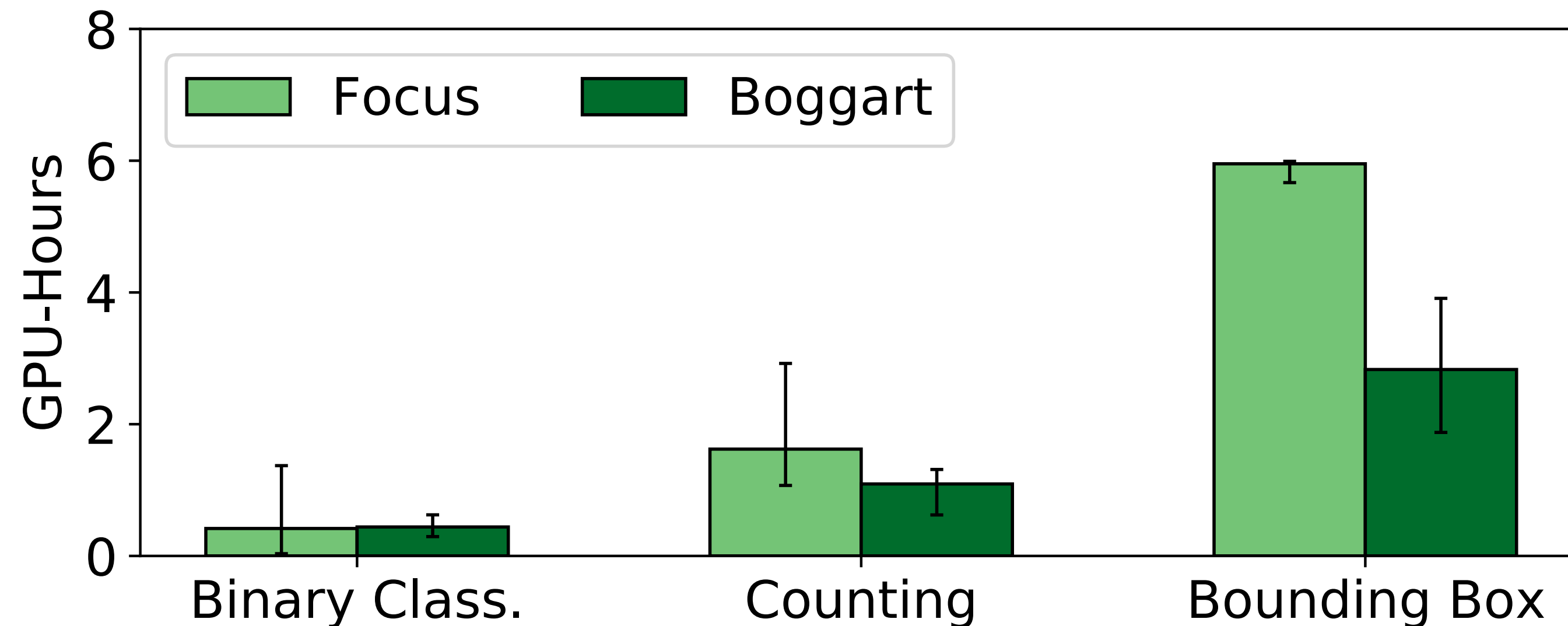
Comparison to Model-Specific Preprocessing

Focus (OSDI '18) leverages model-specific preprocessing to accelerate binary classification queries.

Model: YOLOv3+COCO,
Accuracy Target: 90%

**Low cost for
generalization**

Query Execution

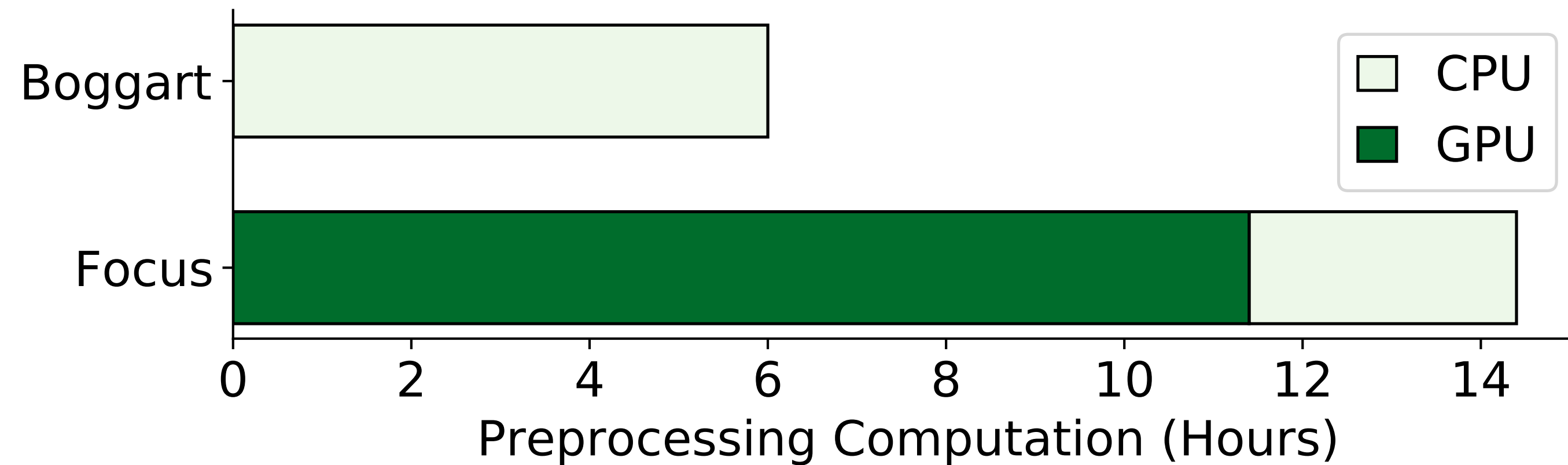


Comparison to Model-Specific Preprocessing

Focus (OSDI '18) leverages model-specific preprocessing to accelerate binary classification queries.

Low cost for generalization

Preprocessing



Evaluation Axes

- ▶ Query-execution speedups
- ▶ Comparison to existing systems
- ▶ Performance on downsampled video
- ▶ Resource scaling
- ▶ Storage costs
- ▶ Parameter sensitivity
- ▶ Generalizability

Boggart

- ▶ A general-purpose accelerator for retrospective querying with diverse user-provided models
- ▶ Leverages model-agnostic computer vision techniques to generate trajectories of areas of motion
- ▶ Despite its generality, its speedups match (and most often, exceed) existing approaches

Source code available at github.com/neilsagarwal/boggart