Efficient Video Object Detection

Huizi Mao
Advisor: Bill Dally
Object Detection from Video

- Object detection is commonly needed for video understanding
- Object detection is computation-intensive
Video Object Detection (VOD)

- Objective of VOD: improve the accuracy-speed trade-off
  - With the aid of temporal information
Video Object Detection Metrics

- How to define **accuracy** and **speed** for VOD?
- Object detection for images:
  - Accuracy -> average precision (AP)
  - Speed -> hardware computation time, FLOPs

- We argue that these metrics are insufficient
  - A video should not be treated as independent images
  - Accuracy -> AP and average delay (AD)
  - Speed -> average latency (AL)
Metrics for swift video object detection

- **Computational latency**: hardware computation time
Metrics for swift video object detection

- **Computational latency**: hardware computation time
- **Algorithmic delay**: number of frames for an algorithm/model to correctly identify an object

Swift processing ≠ Swift detection!
Measuring algorithmic delay

- Delay should be measured and compared under constant false positive (FP) rates
- We define a metric for algorithmic delay - Average Delay (AD)
  - Like average precision
  - Harmonic mean of delays over multiple FP rates

\[
AD = \frac{1}{\bar{p}} - 1 = \frac{1}{\frac{1}{R} \sum_r D_r^{**} + 1} - 1
\]

\(D_r^{**}\) is the mean of all objects’ delay under FP rate \(r\)

Measuring computational latency

- Specify the computation budget in terms of FLOPS, time, etc.
- Average latency (AL) is defined similar to average delay (AD)
  - Measure computational latency when objects are initially detected, and take a harmonic mean.
Evaluating Algorithmic Delay

- Deep Feature Flow (DFF) computes features at key frames, and propagates features for non-key frames.
- Flow-guided Feature Aggregation (FGFA) aggregates features across multiple frames to improve robustness.

![Graph showing performance metrics for different models. The graph includes data points for DFF R-FCN, R-FCN, and FGFA R-FCN, with annotations for 1:N - 1 keyframe out of N frames.](image)
Observations from FGFA

- Fusing information in a temporal window causes slower responses to new objects
Takeaways

- Swift video object detection is not only a computation problem
- Average delay describes the swiftness of an algorithm that average precision cannot tell
Build a swift detection system

- How do we reduce overall detection latency?
  - With a fixed computation budget (in FLOPs)
Algorithmic delay optimization

- A more accurate backbone almost always leads to a lower delay
  - But it comes with a larger computation budget
- Can we leverage larger backbones with the same budget?
  - Allocating computation wisely is the key!
Algorithmic delay optimization

- In the **spatial domain**, reduce regions to compute
  - Find high-interest regions with tracking information

ResNet-18 as backbone

ResNet-101 as backbone
Standard object detection pipeline

- Tracking is a post-processing step for detection

Multi-object tracker (MOT): Associate detected objects by location/appearance
Cascaded Tracked Detector (CaTDet)

- A detector cascade with temporal feedback
  - Crop and run the detector on high-interest regions only

A small but inaccurate detector

Predict locations of previous objects

*Mao, et al. "CaTDet: Cascaded tracked detector for efficient object detection from video." MLSys 2019*
Algorithmic delay optimization

- In the **temporal domain**, reduce detection frequency enables larger models
  - Find previously detected objects by template matching

Small model
High FPS

Large model
Low FPS

TP: true positive
PatchNet - very efficient template matching

- Patches as filters - fine-grained template correlation
- A shallow CNN model on correlation score maps instead of pixel maps

Advantages of PatchNet

- Very robust for key-frame detection and template matching
  - Only 0.7% AP drop when key-frame interval is 5
  - ImageNet VID, R-FCN ResNet-101

- Very efficient and GPU-friendly
  - 58MFLOPs vs 1-3GFLOPs (SiamFC[6], SiamRPN[7])
Algorithmic delay optimization

- Combine CaTDet and PatchNet for spatial-temporal reduction
  - Outperforms model scaling

ImageNet VID, Faster R-CNN
Average precision comparison

- The advantage is even larger when measuring average precision
Computational latency revisit

- We look at computational latency with non-constant detection time
- Average computation time ≠ average computational latency
Computational latency revisit

- Example: key-frame methods, e.g., Deep Feature Flow, PatchNet

Average computation time < frame interval

Average computational latency > frame interval
Resource-aware adaptive scheduling

- **Problem:** Both CaTDet and PatchNet suffer from workload imbalance.
- **Solution:** Track or detect based on the remaining computing resources.

Do template tracking when:
remaining time < $\alpha \cdot$ budget
A swift detection system

- Combining spatial skipping and temporal skipping

CaTDet system

SwiftDet system
Results on VIDT dataset

- Combing CaTDet and PatchNet lowers algorithmic delay
- Dynamic scheduling lowers computational latency

Baseline detector: ResNet-101, Faster R-CNN
Hardware constraint: 70GFLOPS

CaTDet-x: filter regions with confidence < x
PatchNet@N: do detection every N frames
Summary

▪ New evaluation metrics for video object detection

- Average precision
- Average runtime
- Average precision
- Average delay
- Average latency

▪ A swift detection system
  › Spatial-temporal reduction for high AP and low AD
  › Adaptive scheduling for low AL
Reference


