DEMONSTRATION OF THE BLAZEIT VISUAL QUERY ENGINE USING SPECIALIZED NEURAL NETWORKS

Daniel Kang [1] **Peter Bailis** [1] **Matei Zaharia** [1]

ABSTRACT

Video volumes are growing and analysts can query this video to understand the real world. However, analysts are unable to watch or manually annotate the sheer volumes of video. As a result, they have turned to automatic analysis in the form of deep learning. However, this approach suffers two key problems: deep neural networks are extremely computationally expensive and they often require complex, imperative code to deploy. In response, we have built a visual query engine, BLAZEIT, in the Stanford DAWN lab. BLAZEIT features a query language and novel uses of specialized neural networks and fast filters that can deliver order-of-magnitude speedups. We propose a live demonstration of BLAZEIT. We will have an interactive interface where users can input queries in BLAZEIT's query language, FRAMEQL, and visualize outputs.

1 INTRODUCTION

Video volumes are growing: London alone has over 500,000 CCTVs (bbc, 2015). Analysts can query this video to understand the real world. For example, an urban planner could ask which times of the day are the busiest at specific intersections or an autonomous vehicle data scientist could find rare clips of pedestrians jaywalking for simulation purposes.

Analysts are increasingly turning to automated analysis in the form of deep neural networks (DNNs), as manually watching or annotating the video is prohibitively expensive and time-consuming. For example, object detection DNNs can return the position and classes of objects in a frame of video (He et al., 2017). This data can subsequently be used to answer aggregate statistics over videos or find rare events.

Unfortunately, DNN-based video analytics has two major problems. First, DNNs can be extremely expensive: state-of-the-art object detection DNNs (e.g., Mask R-CNN) can execute up to 10× slower than real time. Second, deploying DNNs often requires writing complex, imperative code (e.g., requiring use of libraries such as OpenCV and PyTorch), which is difficult for non-experts.

To address the challenges of DNN-based video analytics, we have built a visual query engine, BLAZEIT, that features novel uses of data-dependent model specialization and fast filters for DNN-based video queries (Kang et al., 2018). BLAZEIT contains a query language, FRAMEQL, and an optimizing execution engine that optimizes and efficiently executes FRAMEQL queries.

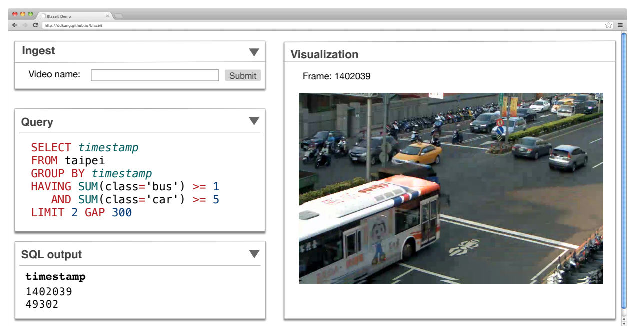

Figure 1. An example query with example output and visualized frame.

FRAMEQL alleviates usability challenge by providing a SQL-like interface for specifying queries. Analysts need only be familiar with SQL and do not need to write complex, imperative code to specify queries.

BLAZEIT's query optimization and execution engine alleviates scalability challenges by *automatically* optimizing user-provided queries. BLAZEIT leverages specialized neural networks (Kang et al., 2017) and fast filters inferred from queries. BLAZEIT's optimized query plans can provide up to three orders of magnitude improvements over naive plans.

We propose a demonstration of the BLAZEIT query engine. We show a schematic of the demonstration in Figure 1 and describe the demonstration below.

[1]Stanford DAWN Project. Correspondence to: Daniel Kang <blazeit@cs.stanford.edu>.

Proceedings of the 2^{nd} SysML Conference, Palo Alto, CA, USA, 2019.

```
SELECT FCOUNT(*)
FROM taipei
WHERE class = 'car'
ERROR WITHIN 0.1
AT CONFIDENCE 95%
```

(a) The FRAMEQL query for counting the frame-averaged number of cars.

```
SELECT timestamp
FROM taipei
GROUP BY timestamp
HAVING
    SUM(class='bus')>=1
AND SUM(class='car')>=5
LIMIT 10 GAP 300
```

(b) The FRAMEQL query for selecting 10 frames of at least one bus and five cars, with each frame at least 10 seconds apart.

```
SELECT *
FROM taipei
WHERE class = 'bus'
  AND redness(content) >= 17.5
  AND area(mask) > 100000
GROUP BY trackid
HAVING COUNT(*) > 15
```

(c) The FRAMEQL query for selecting all the information of red buses at least 100,000 pixels large, in the scene for at least 0.5s (at 30 fps, 0.5s is 15 frames). The last constraint is for noise reduction.

Figure 2. Three FRAMEQL example queries.

2 DEMONSTRATION

Overview. We will highlight the ease of use and efficiency of BLAZEIT through an interactive, live demonstration. Our demonstration will contain an interface for users to input FRAMEQL queries and a visualization tool to display metadata and visualize results. We describe the techniques we will showcase, the interface, and the interactive, live components of the demonstration in turn.

We will require no special equipment. Queries will be executed on the presenter's laptop and displayed on a monitor.

Techniques showcased. We will showcase BLAZEIT's query optimization in the form of model specialization and inferring fast filters.

A key technique that BLAZEIT uses is model specialization (Kang et al., 2017), in which a larger neural network (e.g., Mask R-CNN) is used to train a smaller, more efficient neural network on a restricted task or distribution (e.g., full object detection vs counting the number of cars in a frame). Instead of naively applying object detection to all frames of a video, BLAZEIT can automatically train specialized models for a wide class of visual queries. We will showcase the efficiency of BLAZEIT's specialized models by executing the specialized neural networks on the presenter's *laptop* and compare to Mask R-CNN executing on a GPU.

BLAZEIT can also infer fast filters from query contents. In queries where predicates are applied to the video (e.g., Figure 2c), BLAZEIT can train filters to discard irrelevant frames or safely discard frames. We will showcase this process by displaying query statistics.

Interface. Our demonstration will have an interactive user interface where users can input and execute FRAMEQL queries, we show a schematic in Figure 1. The result of a FRAMEQL query is a relation, which will be displayed in the interface. In cases where the query returns frames, the demonstration will also contain a visualization tool that will display the returned frames and extra metadata, depending on the query.

The demonstration will be *interactive* and *live*: users will be able to submit queries to our interface and the queries will be run live.

Use case and queries. Our demonstration will focus on an urban planning use case. We will use real video collected from urban web cams. The user will be able to input several queries related to analyzing traffic. We highlight three examples below; we show the FRAMEQL queries in Figure 2.

We show in Figure 2a how to count the average number of cars per frame in FRAMEQL. The urban planner may be interested in understanding aggregate traffic behaviors and run such a query. BLAZEIT can provide up to **8.7**$\times$ speedups over naive AQP and orders of magnitude speedups over naive query plans.

We show in Figure 2b how to find instances of at least one bus and at least five cars. The urban planner may be interested in qualitatively understanding how public transit interacts with congestion and searches for such clips. BLAZEIT can provide up to **500**$\times$ speedups over AQP and recent work on optimizing video analytics.

We show in Figure 2c how to find instances of large red buses. The urban planner may notice that red buses are indicative of tour buses and searches for instances of such buses to understand tourism in the city. BLAZEIT can provide up to **50**$\times$ speedups over naive plans.

Acknowledgments

This research was supported in part by affiliate members and other supporters of the Stanford DAWN project—Google, Intel, Microsoft, NEC, Teradata, and VMware—as well as DARPA under No. FA8750-17-2-0095 (D3M), industrial gifts and support from Toyota Research Institute, Juniper Networks, Keysight Technologies, Hitachi, Facebook, Northrop Grumman, NetApp, and the NSF under grants DGE-1656518 and CNS-1651570.

REFERENCES

Cctv: Too many cameras useless, warns surveillance watchdog Tony Porter, 2015. URL http://www.bbc.com/news/uk-30978995.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *ICCV*. IEEE, 2017.

Kang, D., Emmons, J., Abuzaid, F., Bailis, P., and Zaharia, M. Noscope: optimizing neural network queries over video at scale. *PVLDB*, 10(11), 2017.

Kang, D., Bailis, P., and Zaharia, M. Blazeit: Fast exploratory video queries using neural networks. *arXiv preprint arXiv:1805.01046*, 2018.