# SVQ: Streaming Video Queries

Ioannis Xarchakos
University of Toronto
xarchakos@cs.toronto.edu

Nick Koudas
University of Toronto
koudas@cs.toronto.edu

## ABSTRACT

Recent advances in video processing utilizing deep learning primitives achieved breakthroughs in fundamental problems in video analysis such as frame classification and object detection enabling an array of new applications.

In this demo we present *SVQ* a system capable of executing declarative queries on streaming video. The system utilizes a set of approximate filters to speed up queries that involve objects of specific type (e.g., cars, trucks, etc.) on video frames with associated spatial relationships among them (e.g., car left of truck). The resulting filters are able to assess quickly if the query predicates are true to proceed with further analysis of the frame or otherwise not consider the frame further avoiding costly object detection and localization operations. The filters utilize extensible deep neural architectures and are easy to deploy and utilize.

We demonstrate that the application of our filtering techniques in the context of SVQ enable declarative queries on video streams increasing dramatically the frame processing rate and speed up query processing by at least two orders of magnitude depending on the query.

**ACM Reference Format:**
Ioannis Xarchakos and Nick Koudas. 2019. SVQ: Streaming Video Queries. In *Proceedings of ACM SIGMOD conference (SIGMOD'19).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

In the last few years, Deep Learning (DL) [8, 11] has become a dominant artificial intelligence (AI) technology in industry and academia. Although by no means a panacea for everything related to AI it has managed to revolutionize certain important practical applications such as machine translation, image classification, image understanding, video query answering and video analysis.

Video data abound; as of this writing 300 hours of video are uploaded on Youtube every minute. The abundance of mobile devices enabled video data capture en masse and as a result more video content is produced than can be consumed by humans. This is especially true in surveillance applications. Thus, it is not surprising that a lot of research attention is being devoted to the development of techniques to analyze and understand video data in several communities. The applications that will benefit from advanced techniques to process and understand video content are numerous ranging from video surveillance and video monitoring applications, to news production and autonomous driving.

Declarative query processing enabled accessible query interfaces to diverse data sources. In a similar token we wish to enable declarative query processing on streaming video sources to express certain types of video *monitoring queries*. Recent advances in computer vision utilizing deep learning deliver sophisticated object classification [7, 14] and detection algorithms [1–3, 13]. Such algorithms can assess the presence of specific objects in an image, assess their properties (e.g. color, texture), their location relative to the frame coordinates as well as track an object from frame [6] to frame delivering impressive accuracy. Depending on their accuracy, state of the art object detection techniques are far from real time [3]. However current technology enables us to extract a *schema* from a video by applying video classification/detection algorithms at the frame level. Such a schema would detail at the very minimum, each object present per frame, their class (e.g., car) any associated properties one is extracting from the object (e.g., color), the object coordinates relative to the frame. As such one can express numerous queries of interest over such a schema.

SVQ is an embodiment of our research [5] on declarative query processing over streaming video going beyond detection of frames with objects of a specific class or with set properties [4]. In particular we focus on queries involving *count* and *spatial constraints* on objects detected in a frame. Considering for example the image in Figure 1(a) we would like to be able to execute a query of the form [1]:

```
SELECT cameraID, frameID, C1 (F1 (vehBox1)) AS vehType1,
C1 (F1 (vehbox2)) AS vehType2, C2 (F2 (vehBox1)) AS vehColor
```

[1] Adopting query syntax from [9]


*SIGMOD'19, June 2019, Amsterdam Netherlands NL*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

```
FROM (PROCESS inputVideo PRODUCE cameraID, frameID,
vehBox1, vehBox2 USING VehDetector)
WHERE vehType1 = car AND vehColor = red AND
vehType2 = truck AND (ORDER(vehType1, vehType2) = RIGHT
```

that identifies all frames in which a red car has a truck on its right. In the query syntax, $C_i$ are classifiers for different object types (vehicle types, color, etc) and $F_i$ are features extracted from areas of a video frame in which objects are identified (using *vehDetector* which is an object detection algorithm). Naturally queries may involve more than two objects. Numerous types of spatial constraints exist such as *left, right, above, below*, as well as combinations thereof. Categorization of such constraints from the field of spatial databases are readily applicable [10]. Our interest in not only to capture constraints among objects but also constraints between objects and areas of the visible screen in the same fashion (e.g., bicycle not in bike lane, where bike lane is identified by a rectangle in the screen). We assume that the query runs continuously and reports frames for which the query predicates are true.

Although object detection algorithms have advanced substantially over the last few years [1–3, 13] their performance on video streams is still far from real time. From a processing point of view if one could afford to execute state of the art object detection and suitable classification for each frame in real time, answering a query as the one above would be relative easy. We would evaluate the predicates on sets of objects at each frame as dictated by the query aiming to determine whether they satisfy the query predicate. After the objects on a frame have been identified along with their locations and types as well as features, query evaluation would follow by applying well established spatial query processing techniques. Such a *brute force* approach is far from real time as currently state of the art object detectors run at a few frames per second [13].

As a result we have proposed [5] a series of relatively inexpensive filters, that can determine if a frame is a candidate to qualify in the query answer. As an example, if a frame only contains one object (*count filter*) or if there is no red car or truck in the image or there is no car right of a truck in the frame (*class location filter*), it is not a candidate to yield a query answer. We fully process the frame with object detection algorithms only if they pass suitably applied filters. Depending on the selectivity of the filters, one can skip frames and increase the rate at which streaming video is processed in terms of frames per second. The proposed filters follow state of the art image classification and object detection methodologies and we precisely quantify their accuracy and associated trade-offs.

Armed with the ability to efficiently answer monitoring queries involving spatial constraints, we embed it as a primitive to answer another important class of video monitoring


(a) (b)

**Figure 1: Example Video Frames: (a) Example Spatial constraints (b) Spatial constraints on a temporal dimension**

queries, namely video streaming aggregation queries involving spatial constraints. Consider for example Figure 1(b). It depicts a car at the left of a stop sign. From a surveillance point of view we would like to determine if this event is true for more than say 10 minutes. This may indicate that the car is parked and be flagged as a possible violation of traffic regulations. We utilize Monte Carlo based techniques to efficiently process such aggregation queries involving spatial constraints between objects.

We focus on single static camera streaming video as this is the case prevalent in video surveillance applications. Moreover since our work concerns filters to conduct estimations regarding objects in video frames and their relationships, we focus on video streams in which objects and their features of interest (e.g. shapes, colors) are clearly distinguishable on the screen for typical image resolutions. As such the surveillance applications of interest in this study consist of frames containing small numbers of objects (e.g., multiples of tens of objects as in city intersections, building security, highway segment surveillance etc) but not thousands of objects. Crowd monitoring applications [15] in which frames may contain multiple hundreds or thousands of objects (sports events, demonstrations, political rallies, etc) are not a focus of our work. Such use cases are equally important but require very different approaches than those we propose herein. They are however important directions for future work.

More specifically we will be able to demonstrate:

- A series of filters that can quickly assess whether a frame should be processed further given video monitoring queries involving count and spatial constraints on objects present in the frame. These include *count-filters (CF)* that quickly determine the number of objects in a frame, *class-count-filters (CCF)* that quickly determine the number of objects on a specific class in a frame and *class-location-filters (CLF)* that predict the spatial location of objects of a specific class in a frame enabling the evaluation of spatial relationships/constraints across

objects utilizing such predictions. In each case we evaluate the accuracy performance trade-offs of applying such filters in a query processing setting.
- Monte Carlo techniques to process aggregate queries involving spatial query predicates that effectively reduce the variance of the estimates. We utilize a generalization of the Monte Carlo based approach to queries involving predicates among multiple objects and demonstrate the performance/accuracy trade-offs of such an approach.
- The architecture of *SVG* and its current modules, including declarative query parsing and execution as well extensible deep learning modules to implement the different types of filters.
- The current user interface and dashboard of *SVG*, the functionality supporting execution of declarative queries involving the proposed deep learning based filters and query types, performance monitors quantifying both the accuracy and performance benefits of the query execution when compared with base lines.

## 2 FILTERING APPROACHES

We briefly review our filtering proposals. We assume video streams with a set *frames per second (fps)* rate; we also assume access to each frame individually. Resolution of each image frame is fixed and remains the same throughout the stream. Our objective is to process each frame fast applying filters and only activate expensive object detection algorithms on a frame when there is high confidence that it will belong to the answer set (i.e., satisfies the query), to make the final decision. Our first set of filters which we refer to as *Image Classification (IC)* is inspired by image classification algorithms [7, 14, 16] and the second set of filters which we refer to as *Object Detection (OD)* is inspired by object detection algorithms [1–3, 12, 13]. The set of filters we propose are approximate and as such can yield both false positive and false negatives. From a query execution perspective multiple filters may be applicable on a single frame.

## 3 SYSTEM ARCHITECTURE

The current architecture of *SVG* is as follows. The front end accepts SQL queries and also provides various nobs for video source selection, performance monitoring and query results display and comparisons (see Section 4). Queries are dispatched to the back end which is responsible for *parsing the query* and incorporating the supported *deep learning predicates* and *deep learning filters*. Deep learning filters is an extensible module that implements our proposed filter predicates for *count estimation* of objects in a frame, *count estimation of objects of a specific class* in the frame, as well as estimation of the *location of given object classes* in the frames. Deep learning predicates is an extensible module that encompasses popular recent deep learning algorithms for object detection, texture/shape extraction and algorithms for precise object localization on an input video frame. These algorithms are embedded in the parsed query representation and relayed to the query execution engine. The execution module utilizes popular deep learning frameworks to execute the query with the assistance of available GPUs. Frames that pass the filters instantiated in the query are subsequently checked with deep learning predicates and then routed to the front end for display.

## 4 DEMO EXPERIENCE

*SVG* is under rapid development. It currently incorporates our proposed algorithms for filtering frames (based on counts, class based counts and object location), allowing to express semantically meaningful video frame queries in an interactive fashion. Users will have the option of video source selection (out of a collection of available videos) and the ability to expresses their queries in SQL suitably enhanced with UDFs to manipulate video object primitives. In addition they will be able to express location based (spatial) constraints among video frame objects that should be satisfied by the queries. Users can also test the impact of different filters in isolation or in combination and observe the resulting query performance. The impact of filters will be analyzed based on the total query processing time (when compared to a query that does not make use of the filters but instead executes the query in a brute force manner) as well as the resulting frame processing rate. Figure 2 presents the front end. There are options for video source selection (the data set to query) along with options to observe results in normal frame rate or slow motion (to ease comprehension of query results) (Area A). Our filters can be selected in isolation so that participants can experiment with each filter as well as an area to issue a comprehensive SQL query (Areas B and C) that includes, object types of interest, their spatial constraints, along with suitable aggregate constraints. Accuracy and precision/recall results (as applicable) are reported in Areas E (for our algorithms) and H (for brute force), complemented with detailed performance numbers (response times and frame rates) in areas F and I respectively. Finally areas D and G of the UI present the actual query results for our proposed techniques and brute force respectively.

## REFERENCES

[1] Ross B. Girshick. 2015. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. 1440–1448. https://doi.org/10.1109/ICCV.2015.169

[2] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision*

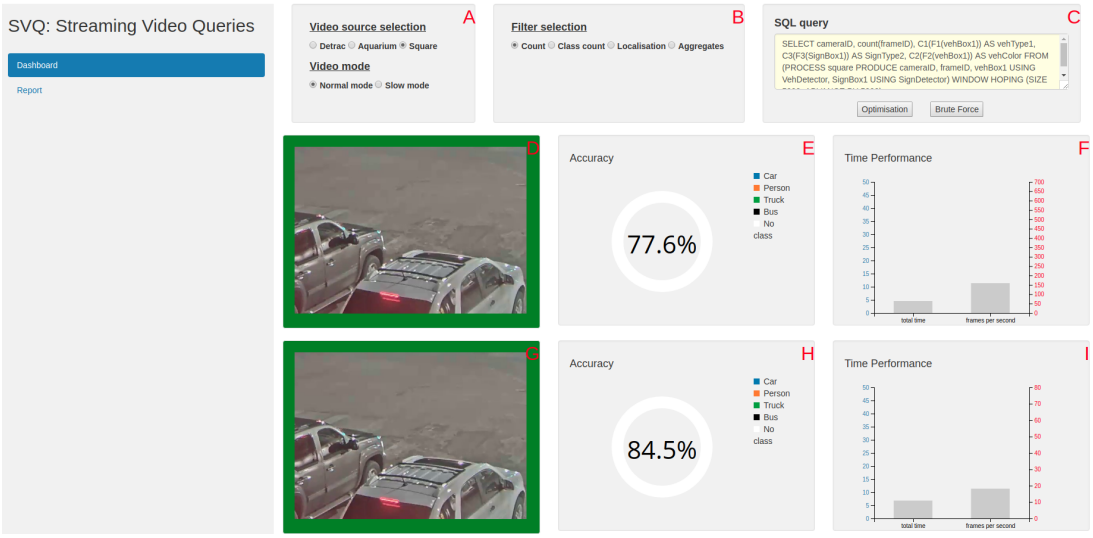


**Figure 2: *SVG* Current front end**

*and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. 580–587. https://doi.org/10.1109/CVPR.2014.81

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 2980–2988. https://doi.org/10.1109/ICCV.2017.322

[4] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: Optimizing Neural Network Queries over Video at Scale. *Proc. VLDB Endow.* 10, 11 (Aug. 2017), 1586–1597. https://doi.org/10.14778/3137628.3137664

[5] Nick Koudas, Raymond Li, and Yannis Xarchakos. 2019. Video Monitoring Queries. In *Submitted for publication.*

[6] Sebastian Krebs, Bharanidhar Duraisamy, and Fabian Flohr. 2017. A survey on leveraging deep neural networks for object tracking. In *20th IEEE International Conference on Intelligent Transportation Systems, ITSC 2017, Yokohama, Japan, October 16-19, 2017*. 411–418. https://doi.org/10.1109/ITSC.2017.8317904

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60, 6 (May 2017), 84–90. https://doi.org/10.1145/3065386

[8] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444. https://doi.org/10.1038/nature14539

[9] Yao Lu, Aakanksha Chowdhery, Srikanth Kandula, and Surajit Chaudhuri. 2018. Accelerating Machine Learning Inference with Probabilistic Predicates. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18)*. ACM, New York, NY, USA, 1493–1508. https://doi.org/10.1145/3183713.3183751

[10] Dimitris Papadias, Timos Sellis, Yannis Theodoridis, and Max J. Egenhofer. 1995. Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-trees. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (SIGMOD '95)*. ACM, New York, NY, USA, 92–103. https://doi.org/10.1145/223784.223798

[11] Marc'Aurelio Ranzato, Geoffrey E. Hinton, and Yann LeCun. 2015. Guest Editorial: Deep Learning. *International Journal of Computer Vision* 113, 1 (2015), 1–2. https://doi.org/10.1007/s11263-015-0813-1

[12] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 779–788. https://doi.org/10.1109/CVPR.2016.91

[13] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

[14] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014). arXiv:1409.1556 http://arxiv.org/abs/1409.1556

[15] Vishwanath Sindagi and Vishal M. Patel. 2017. A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation. *CoRR* abs/1707.01202 (2017). arXiv:1707.01202 http://arxiv.org/abs/1707.01202

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 1–9. https://doi.org/10.1109/CVPR.2015.7298594