Query-Driven Video Event Processing for the Internet of Multimedia Things

Piyush Yadav, Dhaval Salwala, Felipe Arruda Pontes, Praneet Dhingra, Edward Curry
Insight SFI Research Centre for Data Analytics, Data Science Institute
National University of Ireland Galway, Ireland
{piyush.yadav,dhavalvinodbhai.salwala,f.arrudapontes1,praneet.dhingra,edward.curry}@nuigalway.ie

ABSTRACT

Advances in Deep Neural Network (DNN) techniques have revolutionized video analytics and unlocked the potential for querying and mining video event patterns. This paper details GNOSIS, an event processing platform to perform near-real-time video event detection in a distributed setting. GNOSIS follows a serverless approach where its component acts as independent microservices and can be deployed at multiple nodes. GNOSIS uses a declarative query-driven approach where users can write customize queries for spatiotemporal video event reasoning. The system converts the incoming video streams into a continuous evolving graph stream using machine learning (ML) and DNN models pipeline and applies graph matching for video event pattern detection. GNOSIS can perform both stateful and stateless video event matching. To improve Quality of Service (QoS), recent work in GNOSIS incorporates optimization techniques like adaptive scheduling, energy efficiency, and content-driven windows. This paper demonstrates the Occupational Health and Safety query use cases to show the GNOSIS efficacy.

PVLDB Reference Format:
Piyush Yadav, Dhaval Salwala, Felipe Arruda Pontes, Praneet Dhingra, Edward Curry. Query-Driven Video Event Processing for the Internet of Multimedia Things. PVLDB, 14(12): 2847 - 2850, 2021.
doi:10.14778/3476311.3476360

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at http://gnosis-mep.org/.

1 INTRODUCTION

The event processing domain focuses on mining patterns from the data stream in a timely fashion. The event processing systems continuously monitor the incoming stream and generate alerts and notifications whenever an interested pattern is detected. These systems are distributed, decoupled, and are characterized by low-latency, high throughput and real-time performance [8, 11]. With the advent of the Internet of Multimedia Things (IoMT) [1], there is an explosive increase in unstructured data like videos and images. Visual sensors such as smartphones and CCTV cameras are now pervasive and generating continuous streams of video data. Recently, deep learning techniques have achieved a breakthrough in resolving fundamental video analytics issues such as object detection and classification. Presently, event processing systems have limited support to query video streams due to their unstructured data model, spatiotemporal dynamics and require flexible video pipelines to realize distributed video intelligence. The recent works in video analytics focus on specific techniques such as video querying [3, 5, 13, 16], resource efficiency [12, 19], offline optimizations and video pipelines [10]. This paper demonstrates GNOSIS, a native distributed event processing engine to query and process video streams.

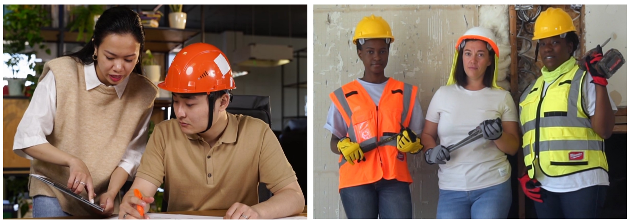


Figure 1: Occupational health and safety scenario : (Left) Non-compliance with safety regulations as one worker is not wearing hard hat, (Right) Workers and Managers classification based on hard hat color (*yellow* and *white*).

Motivating Example. Consider an Occupational Health and Safety (OHS) scenario where the safety of workers is of utmost importance at construction and manufacturing sites. Different safety guidelines have been issued by the health and safety regulatory authorities regarding the usage of Personal Protective Equipment (PPE) to prevent mishaps, construction hazards, and accidents. As per the Bureau of Labor Statistics (BLS), lack of safety helmets resulted in 84% of head injuries among workers. Mainly these sites are located at remote places and are monitored using closed-circuit cameras. As an OHS supervisor, performing manual inspection for safety compliance's (such as wearing a hard hat) from each video camera is time-consuming, tedious, and error-prone (Figure 1). Similarly, there can be other scenarios like counting the number of workers and managers for a specific day. Performing such event-driven tasks leads to multiple challenges such as unstructured video representation, event querying, deploying video pipelines, distributed deployment, and event reasoning. To address the challenges mentioned above, this paper presents GNOSIS, an online, distributed, and near-real-time video event processing framework.

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476360

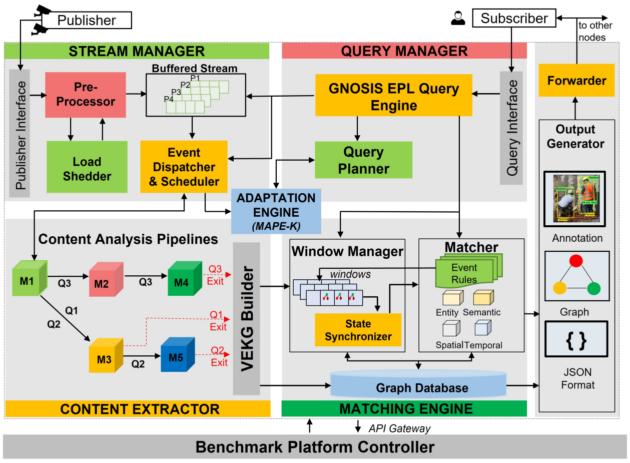


Figure 2: GNOSIS components act as independent microservices that converts the incoming videos as a structured graph stream using DNN models' pipeline and performs the graph-based event matching.

2 GNOSIS

GNOSIS is designed to enable users to write expressive visual queries for video event pattern mining. Figure 2 shows a high-level GNOSIS Architecture which is divided into seven major components- 1) *Stream Manager*: handles publisher video streams, 2) *Query Manager*: deploys GNOSIS Event Processing Language (EPL), 3) *Content Extractor*: extract video content and create video graph stream, 4) *Matching Engine*: perform stateless and stateful video event matching, 5) *Adaptation Engine*: autonomic workload balancing, 6) *Benchmark Platform Controller*: evaluate the system performance and 7) *Output Generator*: generate different query output formats. Some of the key GNOSIS characteristics are as follows:

Complex DNN Models Pipeline Topology. GNOSIS generates flexible DNN and ML models cascade (or pipeline) at runtime, which are pre-trained on specific datasets. The *Scheduler* component creates a control flow of a model pipeline using the CONTENT clause of the GNOSIS EPL. The *Content Extractor* component fed pipeline information as a Directed Acyclic Graph (DAG) where ML models act as nodes. The edges refer to the input and output flow of data from one node to another node. GNOSIS provides a flexible video processing pipeline where each DNN model act as an independent function (Function as a Service) and can be deployed across multiple nodes.

Declarative Video Event Query Language. GNOSIS provides an interactive query interface for querying events over video streams. The GNOSIS EPL is built over powerful openCypher [1] graph query language and enables users to query video events in SQL-like declarative syntax. GNOSIS EPL provides a variety of event rules [18] enabling a wide range of video event detection capabilities such as identifying objects, attributes, and complex spatiotemporal relationships. As shown in Figure 2, the Query Manager component stores, parse and instantiate other GNOSIS components (like windows, matcher, and event dispatcher) as per query metrics. The users can write, add, and delete GNOSIS EPL statements via query interface. The standard GNOSIS EPL syntax is shown below and related examples are demonstrated in section 3 for the Occupational Health and Safety use case.

```
REGISTER QUERY [query name]
OUTPUT [ANN_IMAGE_BBOX | ANN_IMAGE_QUERY_OUTPUT |
K_GRAPH_IMAGE | K_GRAPH_DICT]
CONTENT Service(s)Name
MATCH [openCypher Match query]
WHERE [predicates on Match clause]
FROM [publisher Id]
WITHIN [TUMBLING_COUNT_WINDOW | TUMBLING_TIME_WINDOW
| SLIDING_COUNT_WINDOW | SLIDING_TIME_WINDOW]
RETURN [reference id list, aggregation operators]
```

Video as a Graph Stream. GNOSIS converts the unstructured video data into a structured format using Video Event Knowledge Graph (VEKG) schema [14, 15]. VEKG models the incoming videos as a continuous evolving graph stream with spatial and temporal edges. The spatial edges represent the intraframe relationship between video objects, while temporal edges model the interframe relationships of objects across the frames. In Figure 2 the VEKG Builder component receives the extracted content information from the DNN pipeline and creates a VEKG graph. The VEKG relationships are updated in the windows using *Event Rules*.

State Management. GNOSIS can perform both stateless and stateful video event processing. Stateless event matching is a frame level analytics, primarily focusing on detecting objects and attributes. As shown in Figure 2, the *Windows Manager* component in GNOSIS constitutes different windows operators (such as sliding and tumbling time windows) to handle the stateful video event analytics. Thus, GNOSIS can handle both simple and complex video event patterns that prevail across multiple frames in spatial and temporal dimensions.

Graph-based Video Event Matching. GNOSIS treats video event detection as a graph matching problem. The *Matcher* component receives the event message (VEKG) state from the Windows Manager. For each VEKG state, the matcher updates the VEKG graph inside the RedisGraph [2] database. The registered GNOSIS EPL query is parsed through a Cypher parser to produce an equivalent openCypher query. Later, each openCypher query gets executed on the associated VEKG graph inside the RedisGraph database for evaluation. Based on the OUTPUT EPL clause, the results are fed back to the *Output Generator* event pipeline to visualize the results in different formats such as JSON, graphs, and image annotations. The *Forwarder* component then forwards the result to the query subscriber or route the results to other nodes for further processing.

Adaptive Optimization Service. In GNOSIS, some services can manage themselves to achieve and maintain predefined Quality of Service goals in specific environments. The *Adaptation Engine* allows the system to monitor its components (currently Content Extractor and Scheduler) and analyze their behaviour, to plan and execute the necessary changes using MAPE-K approach. Currently, these adaptations were based on the user queries and the available services to reduce the latency, bandwidth [17] and energy usage [9], with a slight trade-off in result accuracy.

[1] http://opencypher.org/

[2] https://oss.redislabs.com/redisgraph/

Benchmarking and Tracing. GNOSIS constitutes Benchmark Platform Controller (BPC) to evaluate the performance in a controlled and consistent environment. In BPC, each task has its parameters required to run it and execute a list of actions, such as: adding a publisher, adding subscriber and exporting traces. Jaeger, a distributed tracing framework is used to captures traces and once all of these executions are finished, the results are sent using an HTTP POST request.

3 DEMONSTRATION

GNOSIS User Interface. Figure 4 shows the GNOSIS user interface where video streaming source can be published using the *Media Source* component. The publish/unpublish button will register the video stream with the system. The subscribers can write GNOSIS EPL via *Query Editor* and can add multiple queries using *Subscribed Queries* section. The query results are shown in the *QueryOutput* section by selecting the registered queries.

3.1 Use case: Occupational Health and Safety

This use case demonstrates three safety compliance queries to generate hard hat related events for the OHS supervisor.

Q1- Count no. of workers wearing a hard hat. Query 1 (Q1) uses a single content service HardHatDetection to count the number of workers wearing a hard hat. The CONTENT clause refers to DNN model pipelines. Figure 4 shows the worker count message (ANN_IMA GE_QUERY_OUTPUT) and bounding boxes (ANN_IMA GE_BBOX) which are generated from the OUTPUT clause.

```
Q1: REGISTER QUERY CountHardHatWorkers
OUTPUT ANN_IMAGE_BBOX, ANN_IMAGE_QUERY_OUTPUT
CONTENT HardHatDetection
MATCH (worker_hat:HAT)
FROM video_Q1.mp4
RETURN COUNT(worker_hat) as WorkerCount
```

Q2- Detect Hard Hat Compliance Event. Query 2 (Q2) detects a complex high-level HardHatCompliance event using a single HardHatDetection model over a count window of three frames. The RETURN clause limits the required output and sent compliance status as TRUE (if there are workers and everyone is wearing a hat) or False (Figure 5(top)).

```
Q2: REGISTER QUERY HardHatCompliance
OUTPUT ANN_IMAGE_BBOX, ANN_IMAGE_QUERY_OUTPUT
CONTENT HardHatDetection
MATCH (worker_hat:HAT)OR(non_worker_hat:NOT_HAT)
FROM video_Q2.mp4
WITHIN TUMBLING_COUNT_WINDOW(3)
RETURN COUNT(DISTINCT worker_hat)>0
AND COUNT(DISTINCT non_worker_hat)=0 as ComplianceStatus
```

Q3- Worker vs Manager Classification using DNN model pipelines and spatial operator. There are compliances where the construction and manufacturing companies issue different color hats for different stakeholders. For example, workers and managers can be classified based on yellow and white color hats, respectively. Query 3 (Q3) uses three DNN model services- PersonDetection(to detect person), HardHatDetection (to detect hat), and ColorDetection (to detect the hat color) and returns the number of workers based on hat color. The OVERLAP_TOP operator is used to find the spatial alignment of 'hat' and 'person' objects. Figure 3 shows the VEKG construction flow of Q3 with OVERLAP_TOP operator adding a boolean relation of 0 and 1 to the VEKG edge. The Q3 shows the true GNOSIS potential where events can be detected using a combination of DNN pipelines and spatiotemporal operators (Figure 5(bottom)).

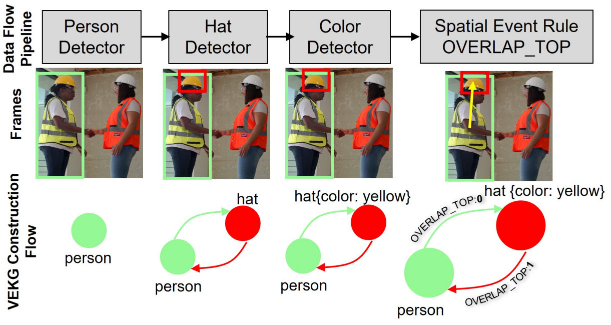


Figure 3: Query 3 content dataflow and graph construction.

```
Q3: REGISTER QUERY CountWorkerManagerCompliance
OUTPUT ANN_IMAGE_BBOX, ANN_IMAGE_QUERY_OUTPUT,
K_GRAPH_IMAGE, K_GRAPH_DICT
CONTENT PersonDetection, HardHatDetection, ColorDetection
MATCH (worker1:PERSON)-[SPATIAL1:OVERLAP_TOP]->(worker_hat1:
HATcolor:'YELLOW') OR (worker2:PERSON)-[SPATIAL2:OVERLAP_TOP]
-> (worker_hat2:HATcolor:'WHITE')
WHERE OVERLAP_TOP=60%
FROM video_Q3.mp4
RETURN COUNT(DISTINCT SPATIAL1) as WorkerCount,
COUNT(DISTINCT SPATIAL2) as ManagerCount
```

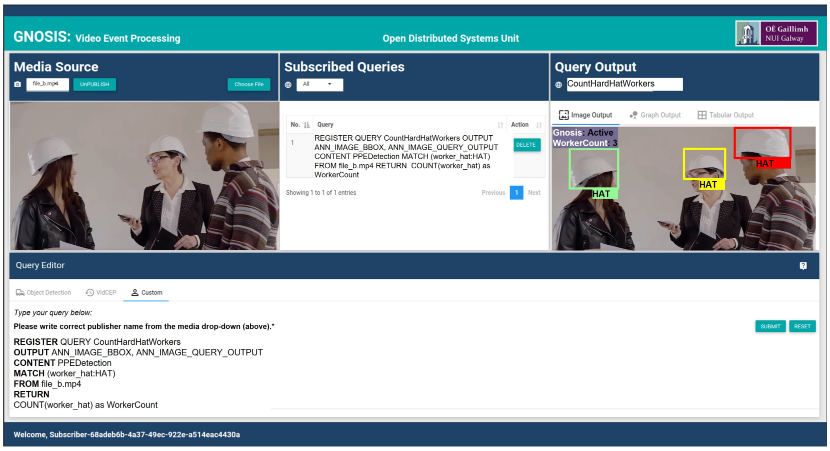
Figure 6 shows the per service latency (seconds) at GNOSIS benchmark dashboard. The PPEDetectionService reports a latency of 0.027,0.031, and 0.033 seconds for Q1,Q2,Q3 respectively. Thus, it can be evident that per service latency increases with increase in the query complexity and number of DNN models in a pipeline. The GNOSIS system is quite robust and can handle large window state depending on the available memory.

4 RELATED WORK

NoScope [6] focuses on fast binary object detection using specialized DNN trained on archived video. FOCUS [4] provides low cost and low latency video event detection on an indexed video dataset. VideoStorm [19] processes live videos on a large cluster where queries are predefined as computation graphs. The systems mentioned above do not concentrate on expressive user queries, state management, and spatiotemporal event patterns. Sprocket [2] proposes a serverless video analytics approach but do not have any query language and lacks spatiotemporal video event matching aspects. SVQ++ [3], Optasia [7], and BlazeIT [5] propose an expressive vision query by adopting SQL syntax but currently lack spatiotemporal matching and state management which are key GNOSIS functionalities.

5 CONCLUSION AND FUTURE WORK

GNOSIS showcases the ongoing work in the field of video event processing and querying. GNOSIS provides an interactive query interface for querying events over video streams. The GNOSIS EPL is built over the powerful openCypher graph query language



No.	Query	Action
1	REGISTER QUERY CountHardHatWorkers OUTPUT ANN_IMAGE_BBOX, ANN_IMAGE_QUERY_OUTPUT CONTENT PPEDetection MATCH (worker_hat:HAT) FROM file_b.mp4 RETURN COUNT(worker_hat) as WorkerCount	DELETE

Showing 1 to 1 of 1 entries Previous 1 Next

Type your query below:

Please write correct publisher name from the media drop-down (above).*

```
REGISTER QUERY CountHardHatWorkers
OUTPUT ANN_IMAGE_BBOX, ANN_IMAGE_QUERY_OUTPUT
CONTENT PPEDetection
MATCH (worker_hat:HAT)
FROM file_b.mp4
RETURN
COUNT(worker_hat) as WorkerCount
```

Welcome, Subscriber-68adeb6b-4a37-49ec-922e-a514eac4430a

Figure 4: GNOSIS user interface with query Q1 visualization.

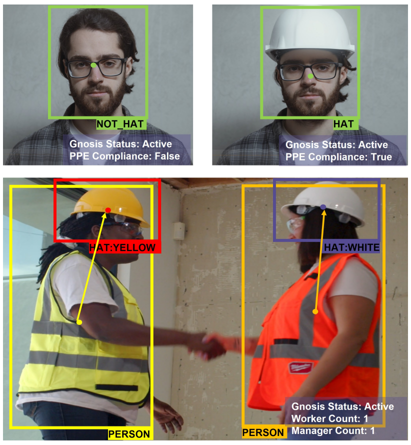


Figure 5: Query Q2 and Q3 visualization.

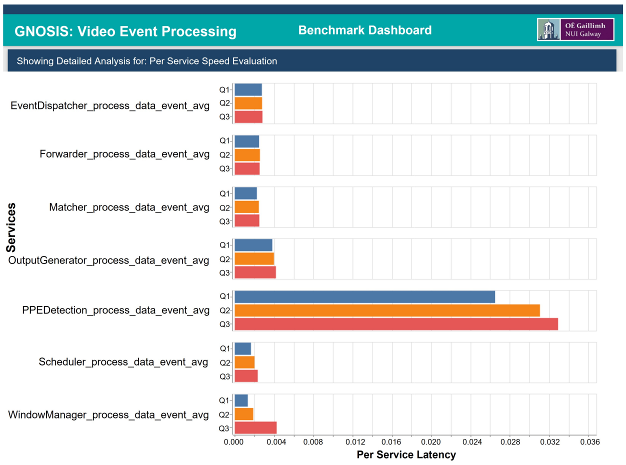


Figure 6: GNOSIS per service latency (seconds).

and enables users to query video events in SQL-like declarative syntax. GNOSIS supports primitive spatial and temporal operator, which enables users to write their own event rules. The paper demonstrates the Occupational Health and Safety use case queries. In future, multimodal data integration, edge-centric optimizations and developing more complex spatiotemporal operators are the key focus area of development in GNOSIS.

ACKNOWLEDGMENTS

This work was supported with the financial support of the Science Foundation Ireland (SFI) grant SFI/12/RC/2289_P2.

REFERENCES

[1] Sheeraz A Alvi, Bilal Afzal, Ghalib A Shah, Luigi Atzori, and Waqar Mahmood. 2015. Internet of multimedia things: Vision and challenges. *Ad Hoc Networks* 33 (2015), 87–111.

[2] Lixiang Ao, Liz Izhikevich, Geoffrey M Voelker, and George Porter. 2018. Sprocket: A serverless video processing framework. In *Proceedings of the ACM Symposium on Cloud Computing*. 263–274.

[3] Daren Chao, Nick Koudas, and Ioannis Xarchakos. 2020. SVQ++: Querying for Object Interactions in Video Streams. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2769–2772.

[4] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B Gibbons, and Onur Mutlu. 2018. Focus: Querying large video datasets with low latency and low cost. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. 269–286.

[5] Daniel Kang, Peter Bailis, and Matei Zaharia. [n.d.]. BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. *Proceedings of the VLDB Endowment* 13, 4 ([n. d.]).

[6] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. 2017. NoScope: Optimizing Neural Network Queries over Video at Scale. *Proceedings of the VLDB Endowment* 10, 11 (2017).

[7] Yao Lu, Aakanksha Chowdhery, and Srikanth Kandula. 2016. Optasia: A relational platform for efficient large-scale video analytics. In *Proceedings of the Seventh ACM Symposium on Cloud Computing*. 57–70.

[8] Gero Mühl, Ludger Fiege, and Peter Pietzuch. 2006. *Distributed event-based systems*. Springer Science & Business Media.

[9] Felipe Arruda Pontes and Edward Curry. 2020. Cloud-Edge Microservice Architecture for DNN-based Distributed Multimedia Event Processing.. In *ESOCC Workshops*. 65–72.

[10] Mohammad Salehe, Zhiming Hu, Seyed Hossein Mortazavi, Iqbal Mohomed, and Tim Capes. 2019. Videopipe: Building video stream processing pipelines at the edge. In *Proceedings of the 20th International Middleware Conference Industrial Track*. 43–49.

[11] Michael Stonebraker, Uğur Çetintemel, and Stan Zdonik. 2005. The 8 requirements of real-time stream processing. *ACM Sigmod Record* 34, 4 (2005), 42–47.

[12] Junjue Wang, Ziqiang Feng, Zhuo Chen, Shilpa George, Mihir Bala, Padmanabhan Pillai, Shao-Wen Yang, and Mahadev Satyanarayanan. 2018. Bandwidth-efficient live video analytics for drones via edge computing. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 159–173.

[13] Piyush Yadav. 2019. High-performance complex event processing framework to detect event patterns over video streams. In *Proceedings of the 20th International Middleware Conference Doctoral Symposium*. 47–50.

[14] Piyush Yadav and Edward Curry. 2019. Vekg: Video event knowledge graph to represent video streams for complex event pattern matching. In *2019 First International Conference on Graph Computing (GC)*. IEEE, 13–20.

[15] Piyush Yadav and Edward Curry. 2019. Vidcep: Complex event processing framework to detect spatiotemporal patterns in video streams. In *2019 IEEE International conference on big data (big data)*. IEEE, 2513–2522.

[16] Piyush Yadav and Edward Curry. 2021. *Query-Aware Adaptive Windowing for Spatiotemporal Complex Video Event Processing for Internet of Multimedia Things.* Ph.D. Dissertation. NUI Galway.

[17] Piyush Yadav, Dhaval Salwala, and Edward Curry. 2021. VID-WIN: Fast Video Event Matching with Query-Aware Windowing at the Edge for the Internet of Multimedia Things. *IEEE Internet of Things Journal* (2021).

[18] Piyush Yadav, Dhaval Salwala, Dibya Prakash Das, and Edward Curry. 2020. Knowledge Graph Driven Approach to Represent Video Streams for Spatiotemporal Event Pattern Matching in Complex Event Processing. *International Journal of Semantic Computing* 14, 03 (2020), 423–455.

[19] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J Freedman. 2017. Live video analytics at scale with approximation and delay-tolerance. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. 377–392.