Scaling Video Analytics Systems to Large Camera Deployments

Paper # 173

ABSTRACT

New computer vision techniques, which enable accurate extraction of insights from videos, and the deployment of cameras *en masse* have made many previously inconceivable applications possible. Scaling video analytics to massive camera deployments, however, presents a tremendous challenge, as cost and resource consumption grow proportionally to the number of camera feeds. This paper is driven by a simple question: can we scale video analytics in such a way that *cost grows sublinearly* (or even remains constant) as we deploy more cameras, *while the accuracy of the analytics remains stable* (or even improves)? We believe the answer is yes. Our key insight is that as cameras are densely installed in a physical space, their video feeds become increasingly *correlated* with each other, correlations that can be harnessed to improve the cost efficiency and accuracy of multi-camera video analytics. By explaining use-cases and technical challenges, we shed light on the potential of leveraging *cross-camera correlations* to scale out video analytics to many cameras, and call on the community to unlock this potential.

1 Introduction

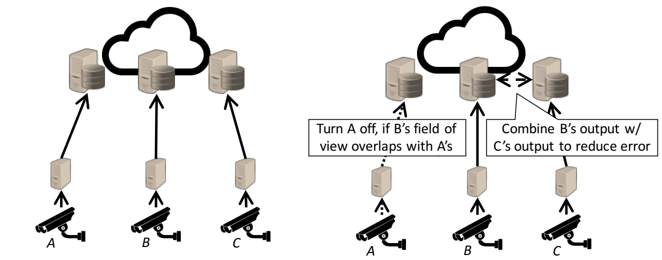
The whole is greater than the sum of its parts.
-Aristotle

Driven by plummeting camera prices and advances in computer vision (DNNs), organizations are deploying cameras at scale for applications ranging from surveillance (*e.g.,* airport or campus security) to flow control (*e.g.,* at traffic lights, building entrances) and retail planning (*e.g.,* inventory management) [29, 21, 41]. Processing the video feeds from these large deployments, using new analytics techniques, however, is becoming prohibitively resource intensive.

A key reason for the rising costs is the fact that in today's video analytics systems, video streams are analyzed *independently*. As a result, the compute required to process the videos grows linearly with the number of cameras. We believe there is an opportunity to both stem this trend of linearly increasing costs, and improve the accuracy of the video analytics, by viewing the cameras *collectively*. Our initial evaluation on a real-world dataset shows that using cameras collectively can yield resource savings of at least 51%, while also improving inference accuracy.

This position paper is based on a simple observation—the dense deployment of cameras over a physical area means that their video feeds are increasingly *correlated*, spatially and temporally. For instance, they may see the same objects, though from different angles or at different points in time.

Our key insight is that these cross-camera correlations can be harnessed, so as to use *substantially lower resources* and/or achieve *higher inference accuracy* than a system that runs complex inference on all video feeds independently. For example, when searching for a specific person, if that person is identified in one camera feed, we can then exclude the possibility of the person appearing in a far-away camera within a short time period, eliminating extraneous queries and reducing false positive detection (Figure 1(a)). Similarly, one can improve accuracy by combining the perspectives of multiple cameras that monitor the same objects at different angles (Figure 1(b)). Moreover, one can opportunistically utilize the compute resources of multiple cameras to collectively alleviate hot spots in the inherently dynamic video analytics workloads. More such opportunities are outlined in §3.



(a) Multi-camera video analytics: Videos are analyzed separately

(b) Saving resources (dotted) & increasing accuracy (dash) via cross-camera correlations

Figure 1: ***Contrasting (a) the traditional per-camera video analytics with (b) the proposed approach that leverages cross-camera correlations.***

With the recent increase in interest for systems infrastructure for video analytics [38, 8, 18, 32, 16], we believe the important next step for the community is designing video analytics stacks for cross-camera applications over a collection of cameras. Current video analytics systems generally analyze video streams independently *even while* useful cross-camera correlations exist [38, 18, 16]. The computer vision literature has optimized for specific applications over a group of cameras (*e.g.,* tracking [29, 41, 33]), but has failed to address the growing cost of inference itself.

We propose to re-architect the video analytics stack so that it can scale to many cameras by fully leveraging these cross-camera correlations. We identify several architectural aspects that are critical to improving resource efficiency and accuracy but are missing in current video analytics systems. Firstly, we illustrate the need for a new module that dynamically generates and maintains up-to-date *spatio-temporal correlations* across cameras. Secondly, we discuss online pipeline reconfiguration and composition, where video pipelines incorporate information from other correlated cameras (*e.g.,* eliminating redundant inference to save cost, or serving as an model ensemble to improve accuracy). Finally, we note the need to process small segments of stored video at faster-than-real-time rates, alongside analytics on live video.

Our goal is not to provide a complete solution, but to motivate the design of an accurate and cost-efficient *multi-camera video analytics* system. In the rest of the paper, we will discuss the trends that drive the need to scale video analytics to swarms of cameras (§2), enumerate the performance benefits of using cross-camera correlations (§3), and outline the technical challenges and potential approaches to building such a system (§4). Our hope is to inspire the practical realization of these ideas in the near future.

2 Camera trends & applications

This section sets the context for using many cameras collaboratively by discussing (1) trends in camera deployments, and (2) the increased recent interest in cross-camera applications.

Dramatic rise in smart camera installations: Organizations are deploying cameras *en masse* to cover physical areas. While enterprises are fitting cameras in office hallways, store aisles, and building entry/exit points, government agencies are deploying cameras outdoors for surveillance and planning. Two key trends are contributing to this trend:

1. *Falling camera costs* allow more enterprises and business owners to install cameras, and at higher density. For instance, today, one can install an HDTV-quality camera with on-board SD card storage for $20 [36], where as three years ago the industry's first HDTV camera cost $1,500 [30]. Driven by the sharp drop in camera costs, camera installations have grown exponentially, with 566 PT of data generated by *new* video surveillance cameras worldwide *every day* in 2015, compared to 413 PT generated by newly installed cameras in 2013 [31].

   There has been a recent wave of interest in "AI cameras" – cameras with compute and storage on-board – that are designed for processing and storing the videos [6, 1, 25, 2]. These cameras are programmable and allow for running arbitrary deep learning models as well as classic computer vision algorithms. AI cameras are slated to be deployed at mass scale by enterprises.
2. *Advances in computer vision*, specifically in object detection and re-identification techniques [40, 27], have sparked renewed interest among organizations in camera-based data analytics. For example, transportation departments in the US are seeing a huge push towards using video analytics for traffic efficiency and planning [20]. A key advantage of using cameras is that they are relatively easy to deploy and can be purposed for multiple objectives.

Increased interest in cross-camera applications: We focus on applications that involve video analytics *across* cameras. While many cross-camera video applications were envisaged in prior research, the lack of one or both of the above trends made them either prohibitively expensive or insufficiently accurate for real-world use-cases.

We focus on a category of applications we refer to as *spotlight search*. Spotlight search refers to detecting a specific type of activity and object (*e.g.,* shoplifting, a person), and then tracking the entity as it moves through the camera network. Compared to detection, tracking tends to require more expensive techniques such as face recognition and person re-identification [41]. Note that objects can be tracked both in the forward direction ("real-time tracking"), and in the backward direction ("investigative search") on recorded video. Spotlight search represents a broad template, or a core building block, for many cross-camera applications. Cameras in a retail store use spotlight search to monitor customers flagged for suspicious activity. Likewise, traffic cameras use spotlight search to track vehicles exhibiting erratic driving patterns. In this paper, we focus on spotlight search on live camera feeds as the canonical cross-camera application.

Metrics of interest: The two metrics of interest in video analytics applications are inference *accuracy* and *cost* of processing. Inference accuracy is a function of the models used for the analytics, the labeled data used for training, and video characteristics such as frame resolution and frame rate [38, 16, 18]. All the above metrics also influence the *cost* of processing – larger models and higher quality videos result in higher accuracy, at the price of increased resource consumption or, equivalently, processing latency.

3 New opportunities in camera deployments: *The whole is greater than the sum of its parts!*

Next, we explain the key benefits – in efficiency and accuracy – of cross-camera video analytics. The key insight is that scaling video analytics to many cameras does not necessarily stipulate a linear increase in cost; instead, one can significantly improve cost-efficiency as well as accuracy by leveraging the spatio-temporal correlations across cameras.

3.1 Key enabler: Cross-camera correlations

A fundamental building block to enabling cross-camera collaboration is the profiles of *spatio-temporal correlations* across cameras. At a high level, these spatio-temporal correlations capture the relationship between the content of camera A and the content of camera B over a time delta Δt.[1] This correlation manifests itself in at least three concrete forms in real-world contexts. Firstly, the same object can appear in multiple cameras, *i.e., content* correlation, at the same time (*e.g.,* cameras in a meeting room) or at different points in time (*e.g.,* cameras placed at two ends of a hallway in a building); Secondly, multiple cameras may share similar characteristics, *i.e., property* correlation, such as *e.g.,* the types, velocities, and sizes of contained objects. Thirdly, one camera may have a different viewpoint on objects than another, resulting in a *position* correlation, *e.g.,* some cameras see larger/clearer faces since they are deployed closer to eye level.

These spatio-temporal correlations are abundant in settings where cameras are densely deployed. As we will show next, the prevalence of these cross-camera correlations enables key

[1] The correlation reduces to a "spatial-only" correlation as $\Delta t \to 0$.

opportunities to use cameras collaboratively. We defer the details and discussion of practical challenges to §4.

Moving forward, we will assume that the cameras have the compute (CPU, GPU) and memory (RAM, SSD) resources to run video analytics modules, and the network connectivity and bandwidth to communicate with each other [6, 25, 3].

3.2 Better cost efficiency

Leveraging cross-camera correlations improves the *cost efficiency* of multi-camera video analytics—one can use significantly fewer resources than a system that runs expensive analytics separately on each feed, without reducing accuracy.

C1: Eliminating redundant inference

In cross-camera applications like spotlight search, there are often far fewer objects of interest than cameras. Hence, ideally, query resource consumption over multiple cameras should not grow proportionally to the number of cameras. We envision two potential ways of doing this by leveraging content-level correlations across cameras (§3.1).

- When two spatially correlated cameras have (partly) overlapping views (*e.g.,* cameras in the same room or covering the same hallway), the overlapped region need only be analyzed once.
- When an object leaves a camera, only a small set of relevant cameras (*e.g.,* cameras likely to see the object in the next few seconds), identified via their spatio-temporal correlations, need search for the object.

In spotlight search, for example, once a suspicious activity is detected and the individual to be tracked is identified, we can selectively trigger multi-class object detection or person re-identification models only on the cameras that the individual is likely to traverse. In other words, we can use spatio-temporal correlations to *forecast* the trajectory of objects.

We analyze the popular "DukeMTMC" video dataset [28], which contains footage from eight cameras on the Duke University campus. Figure 2 shows a map of the different cameras, along with the percentage of traffic leaving a particular camera i that next appears in another camera j. Figures are calculated based on manually annotated human identity labels. As an example observation, within a time window of 90 minutes, 89% of all traffic leaving Camera 1 appears in Camera 2. At Camera 3, an equal percentage of traffic, about 45%, leaves for Cameras 2 and 4. Gains achieved by leveraging these spatial traffic patterns are discussed in §3.4.

C2: Pooling resources across cameras

Since objects/activities of interest are usually sparse, most cameras do not need to run analytics models all the time. Instead, models can be intelligently triggered on demand. This creates a substantial heterogeneity in workloads across different cameras. For instance, one camera may monitor a central hallway and detect many candidate persons, while another camera detects no people in the same time window.

Such workload heterogeneity provides an opportunity for dynamic offloading, in which more heavily utilized cameras

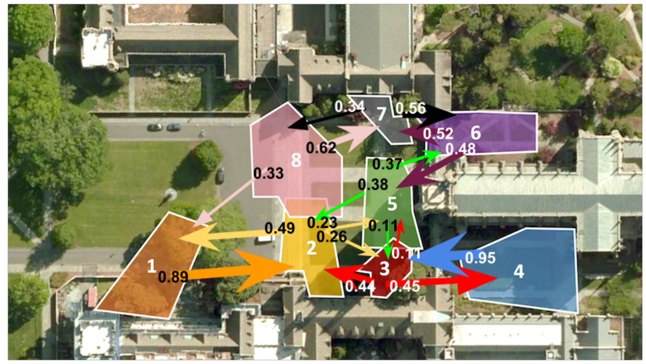


Figure 2: ***Camera topology and traffic flow in the DukeMTMC dataset.***

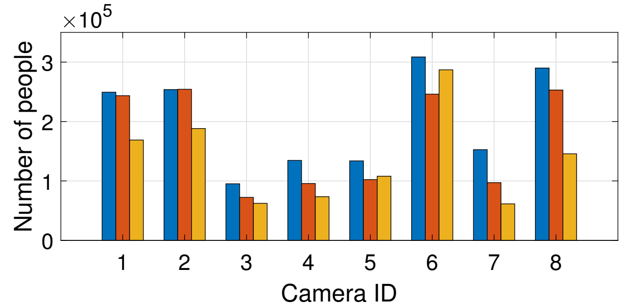


Figure 3: ***Number of people detections on different cameras in the DukeMTMC dataset.***

transfer part of their analytics load to less-utilized cameras. For instance, a camera that runs complex per-frame inference can offload queries on some frames to other "idle" cameras whose video feeds are static. By pooling available resources and balancing the workloads among multiple cameras, one can greatly reduce the amount of resources provisioned on each camera (*e.g.,* size/number of GPUs), from an allocation that would support peak workloads. Such a scheme could also reduce (or altogether eliminate) the need to relegate compute loads to the cloud, a capability constrained by available bandwidth, and in some applications, privacy concerns. *Content* correlations, §3.1 directly facilitate this offloading as they foretell query trajectories, and by extension, workloads on different cameras.

Figure 3 shows the total number of people detected at 60 fps over three consecutive 16.5 minute intervals. Evidently, there is a huge imbalance in traffic both over time and across cameras. Notably, provisioning compute resources based on ideal load balancing here (*i.e.,* splitting the entire workload equally across each camera and time interval), as opposed to peak workload without temporal and cross-camera load balancing, would offer *potential resource savings of* 89.5%.

C3: Collaborative adaptation

Recent work has shown that varying the configurations of video analytics pipelines (*e.g.,* frame rate, resolution) can substantially reduce the resource consumption (by up to two orders of magnitude [38]) with little drop in accuracy; *e.g.,* a low frame rate is adequate to track vehicles during peak hours. As video content varies over time, configurations must also *adapt dynamically*. As stated in [38], however, profiling is an expensive operation, taking several CPU hours, even after many search space optimizations.

Instead of profiling the cameras individually, we can leverage the fact that some cameras have correlated *properties*, §3.1. For instance, cameras in a city will have similar characteristics, such as the size and speed of objects, etc. Traffic likely surges at roughly the same time on most of the cameras within a neighborhood. By exploiting the correlation between cameras, we only need to profile a few (or just one) of these correlated cameras; other cameras can simply re-use this profile. Our initial experiments [16] on the object detection pipeline and traffic camera feeds show that we can reduce the cost of profiling different configurations at near-linear rates relative to the number of traffic cameras in a mid-tier US city, *i.e.,* the profiling cost is reduced by $\sim 7\times$ when the number of correlated cameras grows by $10\times$.

3.3 Higher inference accuracy

In addition to better cost efficiency, we also observe opportunities to leverage cross-camera correlations to improve inference accuracy, without increasing resource usage.

A1: Collaborative inference

Using an ensemble of identical DNNs to render a prediction is an established method for boosting inference accuracy [14]. The technique also applies to model ensembles consisting of multiple, correlated cameras (*e.g.,* with different perspectives on an object). Inference can also benefit from hosting dissimilar models on different cameras. For instance, camera A with limited resources uses a specialized model for car detection, whereas camera B uses a general, heavyweight model for object classification. Then camera A can offload its video to camera B to cross-validate inference results when B is idle.

Two cameras can also be correlated in a *mutually exclusive* manner, which can help reduce false positives. In spotlight search, for instance, if a person p is identified in camera A, we can preclude a detection of the same person p in another camera with no overlap with A. Knowing where an object is likely *not* to show up can improve both recall and precision over a naïve baseline that searches all cameras for a query object. In our initial experiments on the Duke dataset, recall improved from 66% to 82% and precision from 30% to 52% on a test set of five queries, simply by applying a spatial filter on the search space. Removing unlikely candidates from the re-identification gallery reduced false positive matches, which tended to dislodge subsequent tracking, and bring down both accuracy metrics (see §3.4).

A2: Cross-camera model refinement

One source of video analytics error stems from the fact that objects sometimes look differently in real-world settings than in training data. For instance, some surveillance cameras are installed on ceilings, which reduces facial recognition accuracy, due to the oblique viewing angle [23]. These errors can be alleviated by retraining the analytics model, using the output of another camera with an eye level view as the "ground truth". Opportunities for such cross-camera model refinement generally stem from position correlations, §3.1. As another example, traffic cameras under direct sunlight or strong shadows tend to render poorly exposed images, resulting in lower detection and classification accuracy, whereas cameras without lighting interference yield better inference performance [12]. Since lighting conditions change over time, two such feeds can complement each other, both spatially and temporally, via collaborative model training.

3.4 Preliminary results

Table 1 contains a preliminary evaluation of our spotlight search scheme on the Duke dataset [28], which consists of 8 cameras. We expect resource savings to scale linearly with camera count, as the number of cameras correlated with a particular camera does not increase with the surface area of the network. Note that applying a more aggressive spatial filter (*e.g.,* only searching cameras that receive $\geq$10% of the traffic from a source camera) results in both greater savings (cost-efficiency) and higher recall/precision (accuracy).

Table 1: ***Spotlight search results for various levels of spatial correlation filtering,*** *i.e.,* ***minimum visit probability 0-10%. Figures for 100 tracking queries on 8 cameras.***

Visit prob. (%)	Savings (actual, %)	Savings (max., %)	Recall (%)	Precis. (%)
0%	0.0	0.0	59.1	56.8
1%	35.9	43.8	59.2	58.2
3%	48.3	56.3	59.2	58.7
10%	**51.2**	**62.5**	**59.4**	**59.0**

4 Architecting for cross-camera analytics

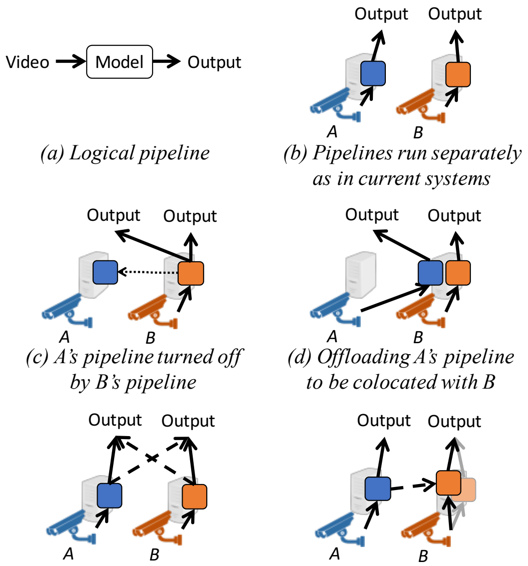
We have seen that exploiting spatial/temporal correlations across cameras can improve both the cost efficiency and accuracy of video analytics in multi-camera settings. Realizing these benefits in practice, however, requires re-architecting the underlying video analytics stack. This section articulates the key missing pieces in current video analytics systems, and outlines the core technical challenges that must be addressed to realize the benefits of cross-camera correlations.

4.1 What's missing in today's video analytics?

Although realizing each optimization in §3.2 and §3.3 requires substantial changes to today's single-camera-oriented analytics systems, we observe that the proposed system can be built on the following four functionalities.

#1: Cross-camera correlation database

Firstly, a new system module needs to be added to determine the spatio-temporal correlations between any pair of cameras and maintain an up-to-date view of these correlations (§3.1). Physically, this module can be a centralized service, decentralized (with each camera maintaining a local copy),



(a) Logical pipeline

(b) Pipelines run separately as in current systems

(c) A's pipeline turned off by B's pipeline

(d) Offloading A's pipeline to be colocated with B

(e) Combining A's result with B's result (and vice versa)

(f) Using A's result to re-train a new model for B

Figure 4: ***Illustrative examples of peer-triggered reconfiguration (c, d) and pipeline composition (e, f) using an example logical pipeline (a) running on two cameras (b).***

or a hybrid system. Different correlations can be represented in various ways. For example, content correlations can be modeled as the *conditional probability* of detecting a specific object in camera B at time t, given its appearance at time $t - \Delta t$ in camera A, and stored as a discrete, 3-D matrix in a database. This database of cross-camera correlations must be dynamically updated, because the correlations between cameras can vary over time for various reasons: video patterns can evolve, cameras can enter or leave the system, and camera positions and viewpoints may change with time. We discuss the intricacy of discovering these correlations, and the implementation of this new module, in §4.2.

#2: Peer-triggered reconfiguration

Today, the execution of a video analytics pipeline (what resources to use and which video to analyze) is largely preconfigured. To take advantage of cross-camera correlations, however, an analytics pipeline must be aware of the inference results of other relevant video streams, and support *peer-triggered reconfiguration* at runtime. Depending on the content of other related video streams, an analytics task can be assigned to the computing resources of *any* relevant camera to process *any* video stream at *any* time. This effectively separates the logical analytics pipeline from its execution. To eliminate redundant inference (C1 of §3.2), for instance, one video stream pipeline may need to dynamically trigger (or switch off) another video pipeline (Figure 4.c). Similarly, to pool resources across cameras (C2 of §3.2), a video stream may need to dynamically offload computation to another camera, depending on correlation-based workload projections (Figure 4.d). To trigger these reconfigurations, inference results need to be shared in real-time *between* pipelines.

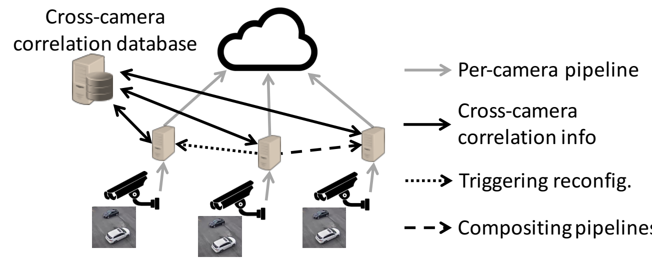


Figure 5: ***End-to-end, cross-camera analytics architecture***

While prior work explores task offloading across cameras and between the edge and the cloud [38, 9, 19], the trigger is usually workload changes on a single camera. In contrast, we argue that such reconfigurations must also consider events on the video streams of other, related cameras.

#3: Video pipeline composition

Analyzing each video stream in isolation also precludes learning from the content of other camera feeds. As we noted in §3.3, by combining the inference results of multiple correlated cameras, *i.e.,* composing multiple video pipelines, one can significantly improve inference accuracy. Figure 4 shows two examples. Firstly, by sharing inference results across pipelines in real-time (Figure 4.e), one can correct the inference error of another less well-positioned camera (A1 in §3.3). Secondly, the inference model for one pipeline can be refined/retrained (Figure 4.f) based on the inference results of another better positioned camera (A2 in §3.3). Unlike the aforementioned reconfiguration of video pipelines, *merging* pipelines in this way actually impacts pipeline output.

#4: Fast analytics on stored video

Recall from §2 that spotlight search can involve tracking an object *backward* for short periods of time to its first appearance in the camera network. This requires a new feature, absent from most video stream analytics systems: fast analysis of stored (past) video data, *in parallel* with live video streams. Stored video must be processed with very low latency (*e.g.,* several seconds), as subsequent tracking decisions depend on the results of the search. In particular, this introduces a new requirement: processing many seconds or minutes of stored video at *faster-than-real-time* rates.

Putting it all together: Figure 5 depicts a new video analytics system that incorporates these proposed changes, along with two new required interfaces. Firstly, the correlation database must expose an interface to the analytics pipelines that reveals the spatio-temporal correlation between any two cameras. Secondly, pipelines must support an interface for real-time communication, to (1) trigger reconfiguration (C1 in §3.2) and (2) share inference results (A1 and A2 in §3.3). This channel can be extended to support the sharing of resource availability (C2) and optimal configurations (C3).

4.2 Technical challenges

In this section, we highlight the technical challenges that must be resolved to fully leverage cross-camera correlations.

1) Learning cross-camera correlations

To enable multi-camera optimizations, cross-camera correlations need to be established in the first place. We envision two basic approaches. One is to rely on domain experts, *e.g.,* system administrators or developers who deploy cameras and models. They can, for example, calibrate cameras [39] to determine the overlapped field of view based on camera locations and the floor plan. A more data-driven approach is to *learn* the correlations from the inference results; *e.g.,* if two cameras identify the same person in a short time interval, they might exhibit a content-dependent correlation.

The two approaches represent a tradeoff— the data-driven approach can better adapt to dynamic correlations, but is more computationally expensive (*e.g.,* it requires running re-id algorithms on all video feeds simply to learn correlations). Note that both approaches will introduce *systematic error* in the correlation database. In spotlight search, for instance, pruned search might fail to track people with abnormal mobility patterns. Resource pooling schemes and collaborative adaptation (C2 and C3 in §3.2) could also degrade with unusual changes in workload and content characteristics.

A potential solution is to let domain experts establish the initial correlation database, and dynamically update it by learning changes from the new inference results. This by itself is an interesting problem to pursue.

2) Resource management in camera clusters

Akin to clusters in the cloud, a set of cameras deployed by an enterprise also represent a "cluster" with compute capacities and network connections. Video analytics work must be assigned to the different cameras in proportion to their resources, while also ensuring high utilization and overall performance. While cluster management frameworks [13, 7] perform resource management, two differences stand out in our setting. Firstly, video analytics focuses on analyzing video *streams*, as opposed to the batch jobs [10, 37] dominant in big data clusters. Secondly, our spatio-temporal correlations enable us to *predict* workload trajectories, and by extension, forecast future resource availability on cameras.

Networking is another important dimension. Cameras often need to share data in real-time (*e.g.,* A1, A2 in §3.3, #3 in §4.1). Given that the links connecting these cameras could be constrained wireless links, the network must also be appropriately scheduled jointly with the compute capacities.

Finally, given the long-term duration of video analytics jobs, it will often be necessary to *migrate* computation across cameras (*e.g.,* C2 in §3.2, #2 in §4.1). Doing so will require considering both the overheads involved in transferring state, and in loading models on the new camera's GPUs.

3) Rewind processing of videos

Rewind processing (#4 in §4.1)—analyzing recently recorded videos—in parallel with live video requires careful system design. A naïve solution is to ship the video to a cloud cluster, but this is too bandwidth-intensive to finish in near-realtime. Another approach is to process the video where it is stored, but a camera is unlikely to have enough resources to do this at a faster-than-real-time rate, while also processing the live video stream.

Instead, we envision a MapReduce-like solution, which utilizes the resources of many cameras (and the cloud) by (1) partitioning the video data and (2) calling on multiple cameras (and cloud servers) to perform rewind processing in parallel. Care is required to orchestrate computation across different cameras, in light of their available resources. Statistically, we expect rewind processing to involve only a small fraction of the cameras at any point in time, thus ensuring the requisite compute capacity on most cameras.

5 Related Work

Finally, we put this paper into perspective by briefly surveying topics that are related to multi-camera video analytics.

Video analytics pipelines: Most video pipelines (*e.g.,* [18, 26, 32, 38, 16]) today focus on using camera, edge, and cloud resources to analyze single video streams. Like this body of work, our paper pursues the objectives of high accuracy and cost efficiency. Our goal, however, is to meet these objectives in a multi-camera setting, a scenario to which existing single-camera-oriented systems scale poorly. Existing techniques for optimizing individual video pipelines are generally orthogonal to our cross-camera analytics platform, and could be co-deployed. Our particular focus is exploiting the opportunities posed by cross-camera correlations, which naturally arise in dense camera clusters.

Camera networks: Multi-camera networks (*e.g.,* [5, 4, 22]) and multi-camera applications (*e.g.,* [17, 11]) have been studied to enable cross-camera communication (*e.g.,* over WiFi) and allow power-constrained cameras to work collaboratively.

Our work is built on these communication capabilities, but focuses on building a custom data analytics stack that spans a cluster of cameras. While some camera networks do perform analytics on video feeds (*e.g.,* [33, 34]), they handle only simple pipelines, and fail to explicitly reduce the resources needed for expensive video analytics, or provide a common interface to support various vision tasks.

Geo-distributed data analytics: Analyzing data stored in geo-distributed services (*e.g.,* data centers) is a related and well-studied topic (*e.g.,* [24, 35, 15]). The key difference in our setting is that camera data exhibits spatio-temporal correlations, which as we have seen, can be used to achieve major resource savings and improve analytics accuracy.

6 Conclusions

The increasing prevalence of enterprise camera deployments presents a critical opportunity to improve the efficiency and accuracy of video analytics via spatio-temporal correlations. The challenges posed by cross-camera applications call for a major redesign of the video analytics stack. We hope that the ideas in this paper both motivate this architectural shift, and highlight potential technical directions for its realization.

7 References

[1] Google Clips. https://store.google.com/us/product/google_clips_specs?hl=en-US.

[2] Hikvision Surveillence. http://blog.hikvision.com/blog/hikvisions-new-deepinview-surveillance-camera-boosts-video-performance.

[3] NVIDIA Metropolis. https://www.nvidia.com/en-us/autonomous-machines/intelligent-video-analytics-platform/, 2018.

[4] K. Abas, C. Porto, and K. Obraczka. Wireless smart camera networks for the surveillance of public spaces. *IEEE Computer*, 47(5):37–44, 2014.

[5] H. Aghajan and A. Cavallaro. *Multi-camera networks: principles and applications*. Academic press, 2009.

[6] Amazon. AWS DeepLens. https://aws.amazon.com/deeplens/, 2017.

[7] Apache Hadoop NextGen MapReduce (YARN). Retrieved 9/24/2013, URL: http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html.

[8] D. Crankshaw, X. Wang, G. Zhou, M. Franklin, J. Gonzalez, and I. Stoica. Clipper: A Low-Latency Online Prediction Serving System. In *USENIX NSDI*, 2017.

[9] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: Making Smartphones Last Longer with Code Offload. In *ACM MobiSys*, 2010.

[10] Dhruba Borthakur, Zheng Shao. Hadoop and Hive Development at Facebook. http://borthakur.com/ftp/hadoopworld.pdf.

[11] B. Dieber, C. Micheloni, and B. Rinner. Resource-aware coverage and task assignment in visual sensor networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(10):1424–1437, 2011.

[12] T. Fu, J. Stipancic, S. Zangenehpour, L. Miranda-Moreno, and N. Saunier. Automatic traffic data collection under varying lighting and temperature conditions in multimodal environments: Thermal versus visible spectrum video-based systems. *Journal of Advanced Transportation*, Jan. 2017.

[13] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. Katz, S. Shenker, and I. Stoica. Mesos: A platform for fine-grained resource sharing in the data center. In *NSDI*, 2011.

[14] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[15] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. R. Ganger, P. B. Gibbons, and O. Mutlu. Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds. In *USENIX NSDI*, 2017.

[16] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica. Chameleon: Video Analytics at Scale via Adaptive Configurations and Cross-Camera Correlations. In *ACM SIGCOMM*, 2018.

[17] A. T. Kamal, J. A. Farrell, A. K. Roy-Chowdhury, et al. Information Weighted Consensus Filters and Their Application in Distributed Camera Networks. *IEEE Trans. Automat. Contr.*, 58(12):3112–3125, 2013.

[18] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia. NoScope: Optimizing Neural Network Queries over Video at Scale. In *VLDB*, 2017.

[19] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl. Energy Characterization and Optimization of Image Sensing Toward Continuous Mobile Vision. In *ACM MobiSys*, 2013.

[20] F. Loewenherz, V. Bahl, and Y. Wang. Video analytics towards vision zero. In *ITE Journal*, 2017.

[21] Y. Lu, A. Chowdhery, and S. Kandula. Optasia: A Relational Platform for Efficient Large-Scale Video Analytics. In *ACM SoCC*, 2016.

[22] L. Miller, K. Abas, and K. Obraczka. Scmesh: Solar-powered wireless smart camera mesh network. In *IEEE ICCCN*, 2015.

[23] NPR. It Ain't Me, Babe: Researchers Find Flaws In Police Facial Recognition Technology. https://www.npr.org/sections/alltechconsidered/2016/10/25/499176469.

[24] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl, and I. Stoica. Low latency geo-distributed data analytics. In *ACM SIGCOMM CCR*, volume 45, pages 421–434. ACM, 2015.

[25] Qualcomm. Vision Intelligence Platform. https://www.qualcomm.com/news/releases/2018/04/11/qualcomm-unveils-vision-intelligence-platform-purpose-built-iot-devices, 2018.

[26] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He. Data distillation: Towards omni-supervised learning. *arXiv preprint arXiv:1712.04440*, 2017.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE CVPR*, 2016.

[28] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking. In *ECCV Workshop on Benchmarking Multi-Target Tracking*, 2016.

[29] E. Ristani and C. Tomasi. Features for Multi-Target Multi-Camera Tracking and Re-Identification. In *IEEE CVPR*, 2018.

[30] SecurityInfoWatch. Market for small IP camera installations expected to surge. http://www.securityinfowatch.com/article/10731727/, 2012.

[31] SecurityInfoWatch. Data generated by new surveillance cameras to increase exponentially in the coming years. http://www.securityinfowatch.com/news/12160483/, 2016.

[32] H. Shen, M. Philipose, S. Agarwal, and A. Wolman. MCDNN: An Approximation-Based Execution Framework for Deep Stream Processing Under Resource Constraints. In *ACM MobiSys*, 2016.

[33] B. Song, A. T. Kamal, C. Soto, C. Ding, J. A. Farrell, and A. K. Roy-Chowdhury. Tracking and activity recognition through consensus in distributed camera networks. *IEEE Transactions on Image Processing*, 19(10):2564–2579, 2010.

[34] M. Taj and A. Cavallaro. Distributed and decentralized multicamera tracking. *IEEE Signal Processing Magazine*, 28(3):46–58, 2011.

[35] A. Vulimiri, C. Curino, P. B. Godfrey, T. Jungblut, K. Karanasos, J. Padhye, and G. Varghese. Wanalytics: Geo-distributed analytics for a data intensive world. In *ACM SIGMOD*, 2015.

[36] I. Wyze Labs. WyzeCam. https://www.wyzecam.com/, 2018.

[37] M. Zaharia, M. Chowdhury, M. Franklin, S. Shenker, and I. Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 10–10. USENIX Association, 2010.

[38] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman. Live Video Analytics at Scale with Approximation and Delay-Tolerance. In *USENIX NSDI*, 2017.

[39] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000.

[40] L. Zheng, Y. Yang, and A. G. Hauptmann. Person Re-identification: Past, Present and Future. *CoRR*, abs/1610.02984, 2016.

[41] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian. Person re-identification in the wild. In *IEEE CVPR*, 2017.