RESEARCH FEATURE

See the World Through Network Cameras

Yung-Hsiang Lu, Purdue University
George K. Thiruvathukal, Loyola University Chicago
Ahmed S. Kaseb, Cairo University
Kent Gauen, Damini Rijhwani, and Ryan Dailey, Purdue University
Deeptanshu Malik, Microsoft
Yutong Huang and Sara Aghajanzadeh, Purdue University
Minghao Guo, University of Illinois at Urbana–Champaign

Millions of network cameras have been deployed worldwide. Real-time data from many network cameras can offer instant views of multiple locations for many applications. We describe the real-time data available from these cameras and potential applications.

The first network camera was, perhaps, deployed at the University of Cambridge, United Kingdom, in 1993 for watching a coffeepot.[1] Millions of stationary cameras (also called *surveillance cameras* or *webcams* in some cases) have been installed at traffic intersections, laboratories, shopping malls, national parks, zoos, construction sites, airports, country borders, university campuses, classrooms, building entrances, and so on. These network cameras can provide visual data (image or video) continuously without human intervention. The data from some (but not all) cameras are recorded for postevent (also called *forensic*) analysis. This article explores the opportunities for analyzing data streams from thousands of network cameras simultaneously. Real-time data may be used in emergency responses; archival data may be used for discovering long-term trends. Figure 1 shows several examples of visual data from network cameras. As can be seen in these examples, the content varies widely from indoor to outdoor and urban to natural environments. This article considers analyzing the data from many heterogeneous network cameras in real time. For more about these cameras, see "Network Cameras."

Digital Object Identifier 10.1109/MC.2019.2906841
Date of publication: 24 September 2019

POTENTIAL APPLICATIONS

Analyzing visual data (image or video) has been an active research topic for decades. Historically, researchers analyze the data taken in laboratories. In recent years, media hosting services (that is, Flickr, YouTube, and Facebook) have made sharing visual data much easier. Researchers started using the data acquired from the Internet to create data sets, such as ImageNet[2] and Common Objects in Context (COCO).[3] Most studies are offline: the analysis is conducted long after the data have been acquired, and there is no specific restriction on the execution time. Often, only pixels are available, and there is no time or location information about the data. As a result, it is not possible to link the data with the context, such as breaking news or a scheduled event. Furthermore, these data sets do not differentiate data taken from city downtowns or national parks. One exception uses periodic

snapshots to observe seasonal trends in environments.[4] The study considers low refresh rates (a few images from each camera per day). In contrast, this article considers data at much higher refresh rates (video or snapshots every few minutes). Adding time and location information can have profound impacts on how the data can be used, as explained in the following examples.

Virtual tour

The World Bank estimates that the number of international tourists reached 1.2 billion in 2015. Nothing can replace the personal experience of visiting a place and enjoying the culture and the local food. However, the hassle of traveling can be unpleasant. Many tourist attractions, such as Yellowstone National Park and the National Zoo, install network cameras and provide real-time data, as shown in Figure 1(d). Through these cameras, it is possible to provide virtual tours to visitors. Furthermore, it is also possible to use network cameras to watch scheduled events. Figure 1(b) and (c) shows images taken in New York City during the Thanksgiving Day Parade in 2014.

Air quality

The U.S. National Park Service (NPS) deploys network cameras monitoring air quality.[5] Each camera takes one image every 15 min and posts the image on the NPS website. The data are archived, and they can be used to study phenology. The data can be cross-referenced with other sources of data, such as the archive of weather data (humidity, temperature, and cloudiness) and rare events (e.g., wildfires). In addition to these cameras in national parks, many TV stations deploy cameras to watch cities. These network cameras

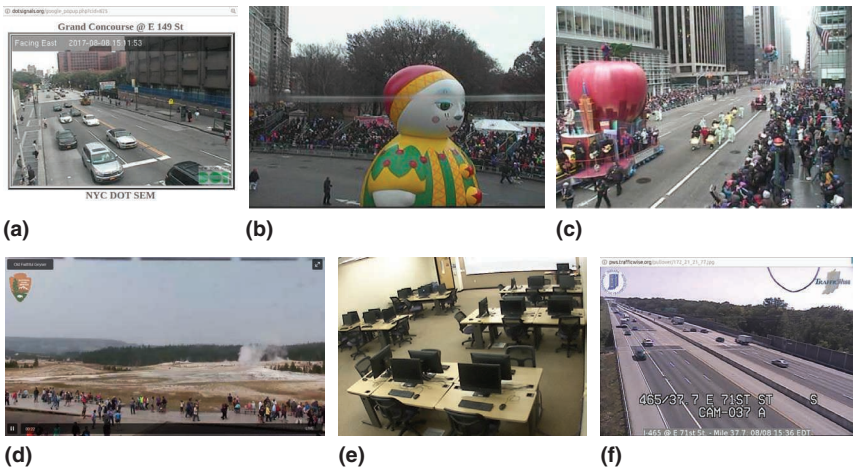

FIGURE 1. The images of (a) New York City, (b) and (c) the Thanksgiving Day Parade in New York City, (d) Yellowstone National Park, (e) a computer lab, and (f) the Interstate 465 highway in Indianapolis. [Sources: (a)–(c) New York City Department of Transportation, used with permission; (d) National Park Service, used with permission; (e) Purdue University, used with permission; and (f) Indiana Department of Transportation, used with permission.]

NETWORK CAMERAS

There is no universally accepted definition of *network cameras*. This article adopts the following definition: a network camera is a camera that is connected to a network (the Internet or an intranet) and can capture visual data automatically and indefinitely without human effort. A network camera may have movement (or pan-tilt-zoom) capability. The cameras may send video streams continuously, take periodic snapshots, or acquire data when events are triggered (such as motion detection). Most network cameras are stationary (that is, their locations are fixed). It is also possible to have mobile network cameras; some cruise ships take periodic snapshots of oceans and transmit the data through satellite networks. Some dashcams have network interfaces, and they may transmit data while the vehicles are moving or parked.

may also be used to assess the air quality in the cities.

Transportation management and urban planning

Improving transportation efficiency is a significant challenge in many cities. Network cameras are widely deployed at traffic intersections. Currently, the real-time data allow city officials to monitor traffic congestion. In the future, these processes could be automatically optimized based on the real-time traffic information provided by the network cameras. Figure 1(f) is an example of a traffic camera in Indianapolis, Indiana. Figure 2(a)–(c) shows the locations of traffic cameras in London, Seattle, and New York City.

Safety and emergency response

Using network cameras, it is possible to monitor large-scale emergencies. Figure 3(a) and (b) shows the flood in Houston on 18 April 2016. Because network cameras continuously acquire data, it is possible to conduct a before-and-after comparison, as shown in Figure 3(c) and (d), when the highways returned to the normal conditions. Our recent study[6] suggests that data from network cameras can complement postings on social networks during emergencies. Network cameras continuously acquire and transmit data without human efforts. Thus, network cameras can be used to monitor locations that have already been evacuated.

Human activities

For 24 hours, an experiment tracked the moving features in a video stream from a camera at Purdue University, West Lafayette, Indiana.[7] The experiment

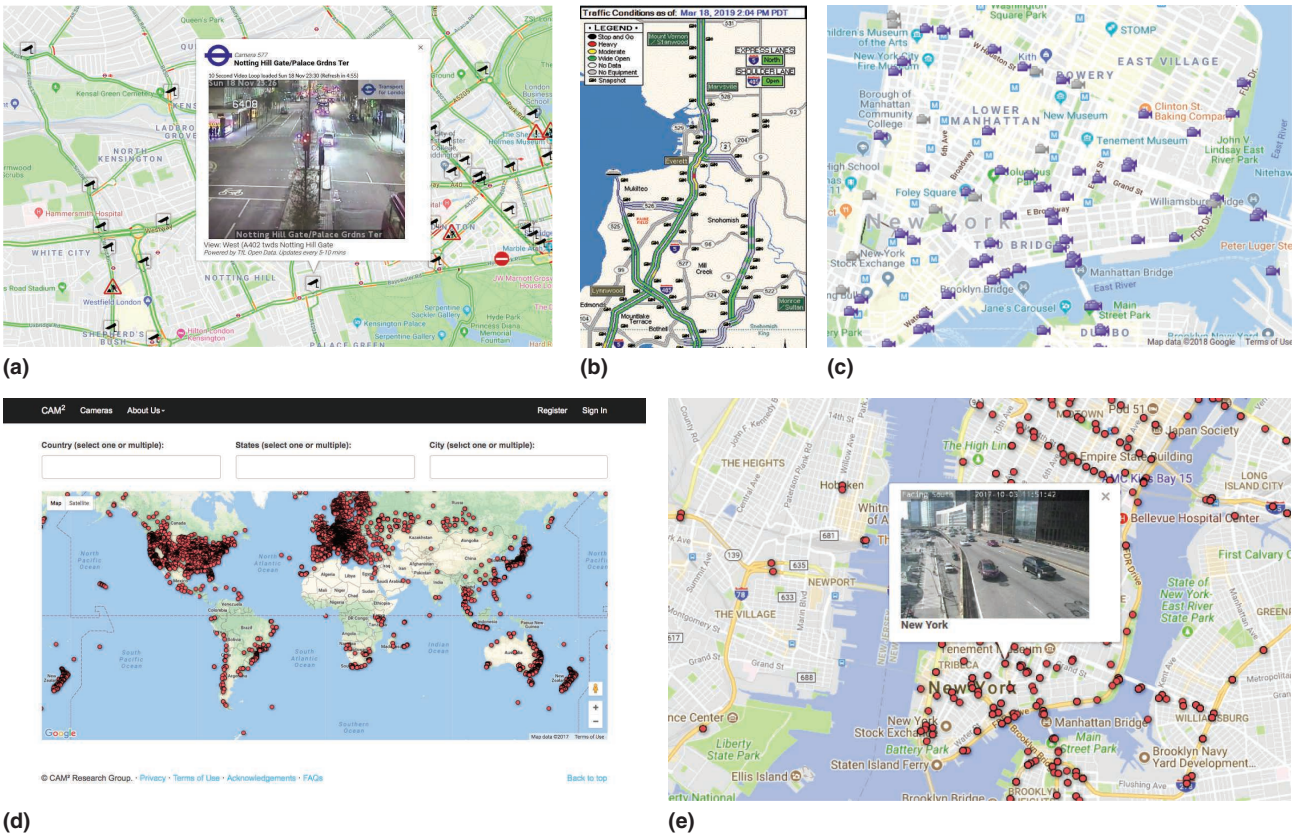

FIGURE 2. The maps of traffic cameras in (a) London, (b) Seattle, and (c) New York City. (d) Screenshot from the CAM2 website: the worldwide camera location map. (e) The New York City camera location map with one real-time image. [Sources: (a) London Department of Transport, used with permission; (b) Washington State Department of Transportation, used with permission; (c) and (e) New York City Department of Transportation, used with permission.]

analyzed 820,000 images (approximately 10 frames per second) from a single camera. Figure 4 shows that more moving features are detected during the day, especially the starting times of lectures. This experiment demonstrates that it is possible to gain insights about the behavior of people using relatively simple analysis programs.

Versatile data from network cameras

Computer vision has made significant progress in recent years. One factor contributing to this success is large data sets with thousands or millions of images and labels. Different data sets may have specific emphases.[8] For example, images posted on social networks tend to have faces at or near the images' centers. Video captured by dashcams tends to have pedestrians at the horizon. Traffic cameras usually look downward from three stories high. These characteristics are a result of the sampling images from different data distributions. The difference among data sets can be called *distinctiveness*. Distinctiveness can be desirable because data sets focus on specific purposes—for face recognition, data from traffic cameras may not be useful. Data from network cameras provide a wide variety of content: indoor or outdoor, city streets or national parks, shopping malls or highways, etc., and these cameras are a rich source for data that are not always easily available in research laboratories.

PURDUE UNIVERSITY'S CONTINUOUS ANALYSIS OF MANY CAMERAS PROJECT

In the previous section, many examples were provided in which analyzing the data from network cameras (real-time images or video streams) can be helpful. In this section, we describe the Continuous Analysis of Many CAMeras (CAM^2) research project at Purdue University, which constructs a system to continuously analyze visual data from many network cameras. Specifically, this section outlines how the research software and services discover network cameras as well as how to retrieve data and metadata from them. It also discusses the back end required for analyzing data in real time and gives a close inspection of the resource manager required for scaling computational needs of analysis programs.

Discover network cameras

The applications described previously require data from many geographically distributed network cameras. This article summarizes the process of finding network cameras and aggregation websites.[9] Internet Protocol (IP) cameras usually have built-in web servers, and the data can be viewed through web browsers (for more information, see "Internet Protocol and Non-Internet Protocol Cameras"). These cameras support HTTP. Different brands have different paths for retrieving data using the GET commands. Several methods can be used to find IP cameras. One obvious

FIGURE 3. The images of (a) and (b) the Houston Flood on 4 April 2016 and (c) and (d) the normal condition on 14 February 2017. (a) and (c) were taken by the same camera, and (b) and (d) were taken by another camera. (Source: Transtar; used with permission.)

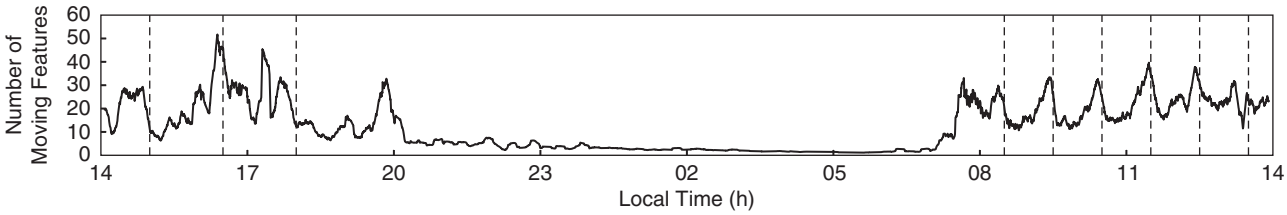

FIGURE 4. Tracking moving features in a video stream of a camera at Purdue University. More moving features exist during the day, especially before the starting times of lectures (indicated by the vertical dashed lines).

INTERNET PROTOCOL AND NON-INTERNET PROTOCOL CAMERAS

Many network cameras can be connected to the Internet directly and have unique Internet Protocol (IP) addresses. They are called *IP cameras* in this article. Some cameras (such as webcams) are connected to computers that make data available on the Internet. They are called *non-IP cameras* in this article because the cameras themselves do not have their own IP addresses. A network camera may have an IP address but not necessarily expose itself, and it may rely on a computer to act as a proxy. In this case, the IP address is the computer's IP address, not the camera's IP address. Many organizations have web servers that show the data from multiple cameras. Because the IP addresses are the web servers' addresses, these cameras are also considered non-IP cameras.

method queries search engines. This method, however, has a low success rate because search engines usually return vendors of network cameras, not IP addresses that can provide real-time data streams. Another method scans IP addresses by sending the GET commands of the known brands. If an IP address responds to the commands positively ("200 OK"), then the IP address becomes a candidate. The candidate is further inspected by the Purdue team.

Currently, this process is manual for two reasons. First, some IP addresses respond to the GET commands even though they are not network cameras (a false positive). Second, the Purdue team inspects the discovered camera and keeps it only if the camera data are from a public location (i.e., a traffic intersection, a park, or a university campus). CAM^2 is actively investigating the automation of discovering network cameras (see the section "Automatically Adding Network Cameras"). To automate privacy filtering in the future, we anticipate that deep-learning models may become capable of scene classification of private versus public locations.

Metadata aggregation

Following network camera discovery, collecting additional information (called *metadata*) about the cameras is important (and possibly required) for data analysis. In this project, metadata include (but are not limited to) the cameras' locations, methods to retrieve data, the format of the data (i.e., MP4, flash, Motion JPEG, JPEG, and portable network graphics), and the information about the refresh rate of the network cameras. Metadata may also describe the data's content (e.g., indoor/outdoor, highways, parks, university campuses, and shopping malls). Three particularly important pieces of metadata are location, data quality, and data reliability.

Location information is required for many of the applications described

earlier. In many cases, the owner of a camera provides the precise location (with longitude and latitude) of the camera. In other cases, street addresses are given. It is also possible to use IP addresses to determine geographic locations, but this approach may be inaccurate for several reasons. An organization (e.g., a university) may have a large campus. Knowing that an IP address belongs to this organization may not provide sufficient details about the camera's location. Furthermore, as mentioned previously, some organizations show multiple data streams on websites. The web servers' locations do not reflect the cameras' locations. In the future, accurate locations may be estimated by cross-referencing information from the network camera images with other resources. Some examples include 1) the time of day, given the sunlight as well as the direction and length of shadows;[10] 2) current events (such as parades); and 3) significant landmarks.

Data quality is critical for analysis and can be measured by many metrics. One is the resolution (number of pixels); another is the refresh rate (frames per second). The data quality may also be determined by the purpose of the applications. For example, if a camera is deployed to monitor traffic, then the data quality is determined by whether it can see congestion clearly or if the view is blocked by trees. In contrast, if a camera is deployed to monitor air quality, it is more important to evaluate whether the view has high visibility.

Reliability refers to the availability of the network camera data. For example, some network cameras provide data only during the daylight hours and not during nighttime hours. Some network cameras are available 24/7. Others may be disconnected for various reasons, such as being damaged during a hurricane.

Web user interface

CAM^2 is designed as an open research tool, available for other researchers to use. It has a web interface (https://www.cam2project.net/) for users to select cameras based on locations. Figure 2(d) is a screenshot of the website. The locations of the cameras are shown as markers on a map (using Google Maps). When a marker is clicked, a snapshot is displayed, as shown in Figure 2(e). The website allows users to select cameras based on country, state (for the United States), and city. The map automatically zooms into the selected country. The markers in Figure 2(d) were originally implemented using the Google Maps client application programming interface (API). However, as the number of cameras in CAM^2 grows, this is no longer a scalable solution. Experiments showed that loading 10,000 markers would take nearly 20 s for rendering the map. To improve scalability, the CAM^2 website uses Google Fusion Tables, which supports tile-based rendering of the markers on Google Maps. The rendering time for 100,000 markers is lower than 2.5 s.

System architecture

Figure 5 shows the three primary components of CAM^2: the user interface, camera interface, and computing platform.[11] The user interface is made up of two access points. First, apps can be programmed with our Python API[12] to access the CAM^2 system. Second, the user can access CAM^2 through a web portal. Aside from the user interaction, the entire CAM^2 system is automated. The web portal allows users to select the camera data streams for analysis, specify the desired analysis parameters (e.g., frame rate and duration), and submit the analysis programs. In other words, the web portal grants users access to the other two essential features of CAM^2.

The camera interface is accessed through the user interface. The camera database provides access to the network cameras. It is a Structured Query Language database that stores the uniform resource locator of the network cameras along with other metadata information. The network cameras

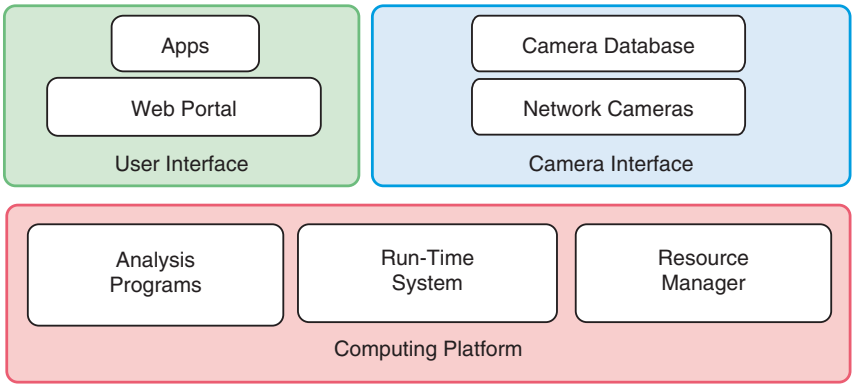

FIGURE 5. CAM^2 has three primary components: the user interface, camera interface, and computing platform.

themselves are, of course, already deployed globally. After or during data collection, the user can run an analysis program from the computing platform. For example, an analysis program can be used to track people (see Figure 4).

The computing platform contains three major components, all of which enable users to run analysis programs on cloud servers.[12] Analysis programs are either created by users or selected from an available list of provided programs. The run-time system is event driven for processing images. After a new image (or video frame) is acquired, a call-back function is invoked. Currently, the data streams are treated independently. Thus, this system is intrinsically parallel and can scale up to process thousands of data streams simultaneously. The resource manager allocates the appropriate number of cloud instances to execute the analysis programs. The cloud instances are responsible for retrieving the visual data from the cameras and executing the analysis programs in real time. The instances may include graphics processing units (GPUs).

Resource manager

Inside the computing platform, the resource manager is a crucial component of CAM^2 for automatically scaling the computational resources to meet analysis programs' demands. Some applications (such as transportation management and emergency response) need to analyze data only at certain time periods (rush hours or when a disaster occurs). Thus, the resource manager needs to adjust the allocated resources as the needs rise and fall. Many factors can affect the resource manager's decisions. Cloud vendors offer dozens of instance types with various amounts of available processor cores, memory, GPU cores, storage, and so on. Furthermore, cloud instances of the same capability (the same number of cores and same amount of memory) have up to a 40% difference in cost.[7]

When the required computation and monetary costs are known for an analysis program, the optimal solution can be determined via a convex optimization problem.[13] It is assumed that computation and memory use scale linearly with the number of cloud instances. This is a reasonable assumption because this is the guarantee provided by the host of a cloud instance. It is shown that the optimal cloud instance is the minimum ratio between the cost of a given cloud instance and the provided computation power (in terms of memory and CPU speed).

To make the problem even more challenging, resource requirements depend on the content of the data as well as the analysis programs. A study suggests using multidimensional bin packing to model the relationships between the needs of analysis programs and the characteristics of cloud instances.[7] The method reduces overall costs by up to 60%.

However, when the geographical distance (thus, the network round-trip time) increases, the data refresh rate may decline.[14,15] The network camera's image quality can suffer. As a result, it is necessary to select a data center close to the network cameras if a high refresh rate is desired. This is an issue, as network cameras are deployed worldwide and cloud data centers are located in many different parts of the world. Therefore, the cost, location, and required image quality for analysis must be considered together for determining the proper cloud instance.[16] The new study shows that modifying the original bin-packing method[7] causes a 56% reduction in cost when compared with selecting the nearest location, and this further improves the original method by 36%.

OPPORTUNITIES AND CHALLENGES

To realize the applications outlined in the "Potential Applications" section, the following research opportunities and challenges must be investigated.

Automatically adding network cameras

Adding network cameras to the CAM^2 database must be further automated to utilize the vast number of network camera data still yet to be discovered. The challenges of using public network camera data leave this valuable data source largely unused. Network camera discovery is challenging due to the lack of common programming interfaces of the websites hosting network cameras. Different brands of network cameras have different programming interfaces. Different institutions organize the data in different ways. Such heterogeneity hinders the usability of the real-time data in emergencies. In other words, network camera data are not readily indexed. For example, there is no easy way to generate images from all of the live public network cameras in New York City. A web search will yield websites that point to camera data in New York City. But the data are spread across many websites, and it is not clear how to easily aggregate images from relevant cameras. To solve this problem, the CAM^2 team is building 1) a web crawler to work with many different website interfaces and 2) a database to provide a uniform interface via a RESTful API. The current version of the RESTful API has been released, and we continues to improve the original version.

Contextual information as weak labels

The proper acquisition of metadata related to each network camera provides useful functionality for the future. The location and time of day provide useful information for automatic data set augmentation. The information can be called *contextual information* of the image/video data. For example, a camera deployed on a busy highway is unlikely to see rhinos or buffalo. If such animals do appear on the highway, this unusual event will likely be reported by the news (also, it can be looked up by time and location). In contrast, a network camera watching a waterfall in a national park should not see semitrucks. Time also provides contextual information about the visual data. The streets in New York City are usually filled with vehicles. On rare occasions, such as the parade shown in Figure 1(b) and (c), the streets are filled with people. Thus, this network camera data can provide almost correct labels by simply assuming that vehicles exist in the data. Modifying the label with cross-referenced news reports (of a parade) and other anomaly detection can form a type of weak supervision.[17]

Weak supervision refers to using labels that can 1) be automatically generated using partially true rules (as in the aforementioned examples), 2) utilize related ground-truth labels that are not for exactly the same task, 3) boost, and 4) hand label with unreliable, nonexpert annotators. Current research demonstrates how different types of weak supervision can be used to improve the accuracy of machine-learning models.[17] Contextual information provides weak labels similar to those generated using partially true rules, and we suspect that future work will also improve model accuracy for image and video tasks (i.e., classification and object detection).

This contextual information can be easily derived if a GPS receiver is included in every camera deployed outdoors. GPS receivers are already in every mobile phone, and it is expected that all future network cameras will also be equipped with them. Time and location may be referenced by sunlight and sun location. Location can be even further refined by significant landmarks. For cameras deployed indoors, methods also exist for positioning them.[18]

Other contextual information (e.g., indoor/outdoor and urban/rural) can be derived from a set of images using a variety of available computer-vision methods. If needed, a new data set of contextual information can be created by training a computer-vision method on it. Although this is only an approximate solution, it is feasible that this will be sufficient for weak labels.

Network camera data are distinct

Commonly used data sets are distinct from each other.[8] For example, the images used in ImageNet[2] can be distinguished from images used in COCO.[3] For object detection tasks, labeled objects are more centrally concentrated for ImageNet than for COCO. This difference is a result of the different data distributions. Existing computer-vision solutions tend to focus on developing accurate models for a small number of data distributions. Even when models are compared across many different data sets, the solutions' applicability beyond these data sets is unclear. The testing error of recently developed models can be overly optimistic even for samples from the same data distribution.[19]

Because the visual data from network cameras may be considered distinct from other data sets, repurposing a model's weights from a similar task may be insufficient. Instead, more sophisticated transfer-learning methods may be required to mitigate the differences among data sets. Additional evidence would be needed to demonstrate that such techniques can handle the wide range of visual data from thousands of network cameras. Future work can investigate the degree to which models trained on available data can be transferred and look for the best method for transferring the model's information. If needed, this may require an expansion of the existing CAM[2] data set for each specific application.[8]

Improving models for emergency response

The data seen during emergency events is uncommon. Thus, the accuracy of machine-learning models when responding to emergencies is likely poor. We propose three methods to improve machine learning in the event of an emergency. The first is to periodically record data before a disaster for an anomaly detection system. Next is to connect CAM[2] to infrastructure for crowdsourcing to gather labeled data on short notice from locations known to have an impending emergency situation. For example, crowdsourcing was used in the Haiti earthquake of 2010.[20] It is conceivable to create a similar infrastructure for images and video in an emergency. Finally, network cameras need to have a uniform interface for easy access in emergency situations, given the proper privacy and legal constraints.[21]

Data-set distinctiveness for active learning

Finding the right subset of data to label is a general concern in machine learning, especially during an emergency when time is short. The problem is also applicable to nonemergency scenarios when the cost of labeling data is the constraint rather than the time. As network camera data are distinct, we wish to further investigate if data set distinctiveness can be used to improve active-learning methods. In this article, active learning is described as follows. Given a large amount of unlabeled data, we must identify the right subset of data to label. A general framework for active-learning methods is often given as the balance between measures of 1) how representative (or typical) the data sample is relative to the true data distribution and 2) how to maximize variance reduction of the model (and equivalently, how to minimize the true risk). Often, the product of the two measures is used to identify the best samples. With an input–output pair, this can be thought of as 1) modeling only the data and 2) modeling only the conditional distribution. Multiplying them together can be thought of as modeling the unnormalized posterior. Because network camera data are distinct from the existing data sets, the use of this new data source may improve the current active-learning methods.

Adaptive and programmable network cameras

Another improvement is to make network cameras self-aware of the context being seen to automatically execute relevant programs. It may be possible for stationary cameras to determine the visual information being captured and install/execute computer-vision programs specialized for the content. Furthermore, network cameras may need to be reprogrammed in emergencies. For example, street cameras in Figure 1(b) and (c) may be specialized for detecting congestion and accidents in normal conditions. During a parade, the cameras may need to be reprogrammed to search for a lost child.[22]

Network cameras provide rich information about the world. The visual data have many applications, including real-time emergency response and discovery of long-term trends. This article presents CAM^2, a software infrastructure constructed at Purdue University for acquiring and analyzing data from network cameras. The article suggests many opportunities using the data and challenges to be conquered.

ACKNOWLEDGMENTS

This project is supported in part by Facebook, Intel, and the National Science Foundation OISE-1427808, IIP-1530914, OAC-1535108, OAC-1747694. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. The authors would like to thank Amazon and Microsoft for providing the cloud infrastructure as well as the organizations that provide the camera data. Many Purdue students in the Vertically Integrated Projects program have contributed to the development of the software components used in CAM^2.

REFERENCES

1. R. Kesby, "How the world's first webcam made a coffee pot famous," BBC News, Nov. 22, 2012. [Online]. Available: http://www.bbc.com/news/technology-20439301
2. ImageNet. [Online]. Available: http://www.image-net.org/
3. T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. European Conf. Computer Vision*, 2014, pp. 740–755.
4. J. D. O'Sullivan, A. Stylianou, A. Abrams, and R. Pless, "Democratizing the visualization of 500 million webcam images," in *Proc. IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, Oct. 2014, pp. 1–5.
5. National Park Service, "NPS Air Quality Web Cameras," [Online]. Available: https://www.nature.nps.gov/air/webcams/
6. C. Surakitbanharn et al., "Cross-referencing social media and public surveillance camera data for disaster response," in *Proc. IEEE Symp. Technologies for Homeland Security*, Oct. 2018. doi: 10.1109/THS.2018.8574200.
7. A. S. Kaseb, A. Mohan, Y. Koh, and Y.-H. Lu, "Cloud resource management for analyzing big real-time visual data from network cameras," *IEEE Trans. Cloud Comput.*, vol. 4, p. 1, 2017. doi: 10.1109/TCC.2017.2720665.
8. K. Gauen et al., "Comparison of visual datasets for machine learning," in *Proc. IEEE Conf. Information Reuse and Integration*, 2017. doi: 10.1109/IRI.2017.59.
9. R. Dailey et al., "Creating the world's largest real-time camera network," in *Proc. Electronic Imaging, Imaging and Multimedia Analytics in a Web and Mobile World*, 2017, pp. 5–12.
10. I. N. Junejo and H. Foroosh, "Estimating geo-temporal location of stationary cameras using shadow trajectories," in *Computer Vision—ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman,

ABOUT THE AUTHORS

YUNG-HSIANG LU is a professor in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana. His research interests include computer vision, distributed systems, and low-power systems. Lu received a Ph.D. from Stanford University, California. He is a Senior Member of the IEEE and a Distinguished Scientist of the Association for Computing Machinery. Contact him at yunglu@purdue.edu.

GEORGE K. THIRUVATHUKAL is a professor in the Department of Computer Science at Loyola University Chicago and visiting faculty at Argonne National Laboratory, Lemont, Illinois. His research interests include parallel computing, software engineering, and programming languages. Thiruvathukal received a Ph.D. from the Illinois Institute of Technology, Chicago. He is a member of the *Computer* Editorial Board. Contact him at gkt@cs.luc.edu.

AHMED S. KASEB is an assistant professor in the Faculty of Engineering at Cairo University, Egypt. His research interests include cloud computing, big data analytics, and computer vision. Kaseb received a Ph.D. from Purdue University, West Lafayette, Indiana. Contact him at akaseb@eng.cu.edu.eg.

KENT GAUEN is a doctoral student in the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana. His research interests include computer vision and machine learning. Gauen received a B.S. in electrical and computer engineering from Purdue University. Contact him at gauenk@purdue.edu.

DAMINI RIJHWANI is an undergraduate student in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana. Her research interests include computer vision and machine learning. Contact her at drijhwan@purdue.edu.

RYAN DAILEY is a master's student in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana. His research interests include distributed systems and data mining. Dailey received a B.S. in electrical and computer engineering from Purdue University. Contact him at dailey1@purdue.edu.

DEEPTANSHU MALIK is a software engineer at Microsoft. Malik received a B.S. in electrical and computer engineering from Purdue University. Contact him at malikd@purdue.edu.

YUTONG HUANG is a master's student in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana. His research interests include operating systems and distributed memory management. Huang received a B.S. in electrical and computer engineering from Purdue University. Contact him at huang637@purdue.edu.

SARA AGHAJANZADEH is a master's student in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana. Her research interests include computer vision and machine learning. Aghajanzadeh received a B.S. in computer information technology from Purdue University, West Lafayette, Indiana. Contact her at saghajan@purdue.edu.

MINGHAO GUO is a master's student in data science at the University of Illinois at Urbana–Champaign. Her research interests include data mining, visualization, and statistical data analysis. Guo received a B.S. in computer science from Purdue University, West Lafayette, Indiana. Contact her at minaaa1003@gmail.com.

Eds. Berlin: Springer-Verlag, 2008, pp. 318–331.

11. A. S. Kaseb, Y. Koh, E. Berry, K. McNulty, Y.-H. Lu, and E. J. Delp, "Multimedia content creation using global network cameras: The making of cam2," in *Proc. IEEE Global Conf. Signal and Information Processing (GlobalSIP)*, Dec. 2015, pp. 15–18.
12. A. S. Kaseb et al., "A system for large-scale analysis of distributed cameras," in *Proc. IEEE Global Conf. Signal and Information Processing*, 2014, pp. 340–344.
13. A. S. Kaseb, A. Mohan, and Y. Lu, "Cloud resource management for image and video analysis of big data from network cameras," in *Proc. Int. Conf. Cloud Computing and Big Data (CCBD)*, Nov. 2015, pp. 287–294.
14. W. Chen, A. Mohan, Y.-H. Lu, T. J. Hacker, W. T. Ooi, and E. J. Delp, "Analysis of large-scale distributed cameras using the cloud," *IEEE Cloud Comput.*, vol. 2, no. 5, pp. 54–62, Sept. 2015.
15. A. Mohan, A. S. Kaseb, Y. Lu, and T. Hacker, "Adaptive resource management for analyzing video streams from globally distributed network cameras," *IEEE Trans. Cloud Comput.*, vol. 5, p. 1, 2018. doi: 10.1109/TCC.2018.2836907.
16. A. Mohan, A. S. Kaseb, Y.-H. Lu, and T. J. Hacker, "Location based cloud resource management for analyzing real-time videos from globally

distributed network cameras," in *Proc. IEEE Int. Conf. Cloud Computing Technology and Science*, 2016, pp. 176–183.

17. A. Ratner, B. Hancock, J. Dunnmon, R. Goldman, and C. Ré, "Snorkel MeTaL: Weak supervision for multi-task learning," in *Proc. 2nd Workshop on Data Management for End-To-End Machine Learning, DEEM'18*. New York: ACM, 2018, pp. 3:1–3:4.
18. P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Commun. Surv. Tuts.*, vol. 19, no. 2, pp. 1347–1370, 2017.
19. B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, Do cifar-10 classifiers generalize to cifar-10? 2018. [Online]. Available: arXiv:1806.00451
20. M. Zook, M. Graham, T. Shelton, and S. Gorman, "Volunteered geographic information and crowdsourcing disaster relief: A case study of the Haitian earthquake," *World Med. Health Policy*, vol. 2, no. 2, pp. 7–33, 2010.
21. Y.-H. Lu, A. Cavallaro, C. Crump, G. Friedland, and K. Winstein, "Privacy protection in online multimedia," in *Proc. 25th ACM Int. Conf. Multimedia, MM '17*. New York: ACM, 2017, pp. 457–459.
22. M. Satyanarayanan, "Mobile computing: The next decade," in *Proc. 1st ACM Workshop Mobile Cloud Computing Services: Social Networks and Beyond, MCS '10*. New York: ACM, 2010, pp. 5:1–5:6.

Digital Object Identifier 10.1109/MC.2019.2937700