IEEE International Conference on Multimedia Big Data 2015

Harvest the Information from Multimedia Big Data in Global Camera Networks

Wei-Tsung Su
Department of Computer Science and Information Engineering,
Aletheia University
New Taipei City, Taiwan
au4451@au.edu.tw

Yung-Hsiang Lu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
yunglu@purdue.edu

Ahmed S. Kaseb
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
akaseb@purdue.edu

***Abstract*—**Many network cameras have been deployed for various purposes, such as monitoring traffic, watching natural scenes, and observing weather. The data from these cameras may provide valuable information about the world. This paper describes the unexplored opportunities and challenges to harvest the information in the global camera networks. A cloud-based system is proposed to harvest the information from many cameras in an efficient way. This system provides an application programming interface (API) that allows users to analyze the multimedia big data from many cameras simultaneously. Users may also store the data for off-line analysis. This system uses the cloud for computing and storage. Experiments demonstrate the ability to process millions of images from thousands of cameras within several hours.

Keywords-multimedia big data, camera networks, cloud computing, application programming interface

I. INTRODUCTION

a) Big Data and Multimedia

Big Data has become one of the hottest topics in recent years. There are different interpretations of "Big Data". Gartner defines Big Data by three Vs: velocity, variety, and volume. The potential of Big Data is to use the large quantity of data for discovering new knowledge and unexpected relationships. Big Data may include many different types. Multimedia data could be a type of Big Data since multimedia data have the three Vs. At multiple frames per second, multimedia data pass through networks at high velocity. Multimedia data also have wide variety: from instructional video in MOOC (massive open on-line course) to commercial movies, from surveillance videos to home videos taken at birthday parties. Besides, storing multimedia data requires large capacity.

b) Network Cameras

Since the introduction of commercial digital cameras in late 1990s, the sales of digital cameras have been growing rapidly. Digital cameras can be divided into two categories. The first includes portable cameras in smartphones, pocket cameras, and single-lens reflex (SLR) cameras. The second category is cameras that are always connected to the Internet. They are generally called *network cameras* or *webcams* because the data from these cameras are intended to be seen on web browsers.

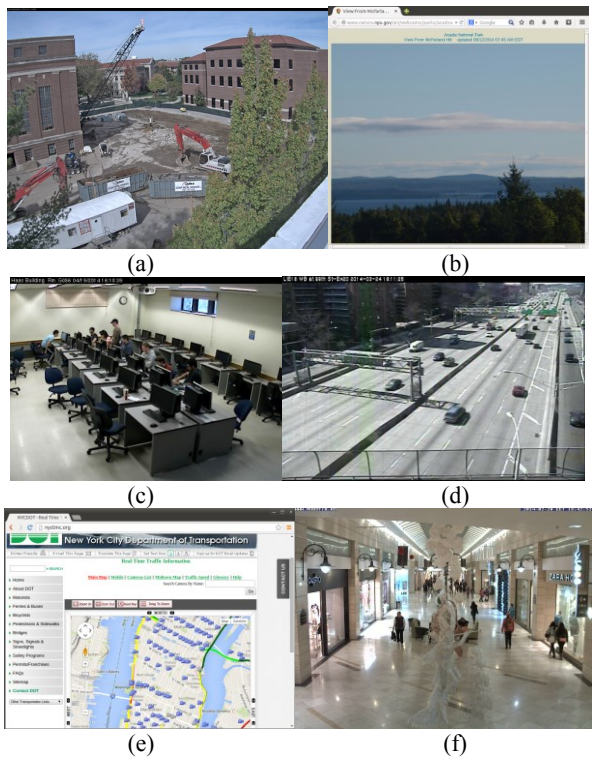

Figure 1. Examples of network cameras. (a) a construction site. (b) a national park. (c) a computer classroom. (d) a highway. (e) traffic cameras in New York City. (f) a shopping mall. These cameras are connected to the Internet and the data are accessible to the public.

Network cameras can be further divided into two types: IP cameras and non-IP cameras. The former has HTTP servers built-in and each camera has a unique IP

address. An IP camera can be connected to the Internet directly. A report [1] estimates that 28 million network cameras will be sold in 2017, at 27.2% growth over the five year from 2012 to 2017. Some network cameras are connected to private networks and the others are connected to the Internet accessible to the general public. A non-IP camera is connected to the Internet through a computer; such a camera has no built-in HTTP server.

Network cameras are deployed for various reasons, for example,

- to watch the progress of constructions, as shown in Figure 1 (a)
- to observe air quality, as shown in Figure 1 (b)
- to monitor traffic on highways, as shown in Figure 1 (d)
- to attract potential customers, as shown in Figure 1 (f)

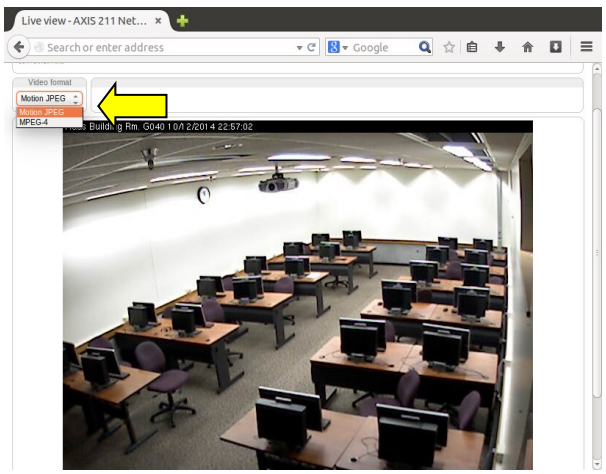

Figure 2. An example of a network camera providing MJPEG and MPEG-4.

c) Network Cameras' Data Formats

Most network cameras provide JPEG as the standard format for images. Each JPEG image requires an individual request. Some cameras also provide videos using MJPEG (motion JPEG), or MPEG-4, or both. Figure 2 shows a network camera that provides both MJPEG and MPEG-4.

MJPEG and MPEG-4 provide multiple frames per request. MPEG-4 detects the changes among adjacent frames. Thus, MPEG-4 is more efficient using network bandwidths when there are few changes (also called motions) between subsequent frames. For cameras that have high motions (such as PTZ, pan-tilt-zoom), MPEG-4 would not be beneficial. In contrast, MJPEG does not detect the changes between frames and each frame is an independent JPEG image. MJPEG is more robust to data loss over the network because each frame is independent. These two formats have advantages and disadvantages. The following table compares these two formats.

	MJPEG	**MPEG-4**
Data	Independent images	Video with motion detection between successive frames
Compression	Intra frame	Intra and inter frame
Data Rate (Mbps)	Higher	Lower
Computation	Lower	Higher
Robustness	Higher	Lower
At Low Rate	Reduce Frame Rates	Repeat frames to keep the frame rates
Lost Frame	No impact on subsequent frames	May impact subsequent frames

d) Volume from Digital Cameras

These velocity and variety of multimedia data are well understood. The volume of multimedia data requires further analysis. Hundreds of millions of photographs are posted on social networks everyday. Every minute, one hundred of hours of videos is uploaded to YouTube [2]. A report by Cisco predicts that in 2018, one million minutes of video content will cross the Internet every second and video will be 79 percent of all consumer Internet traffic [3].

What is the volume of multimedia data? The following is a "Fermi approximation" to understand the volume. "Fermi approximation" is a methodology to obtain quick and rough estimation, suggested by Enrico Fermi. We estimate the volume of data from the network cameras first. Based on the recommendation from Netflix for video [4], 1.5 Mbps is recommended and 5 Mbps is needed for HD (720 scan lines or higher). Some network cameras do not provide video and take snapshots once every few seconds to every few minutes. Many network cameras do not provide HD quality but some have much higher quality. For example, AXIS P1428-E has 3840×2160 resolution. As first-order approximation, consider that each camera produce 10% of 1.5 Mbps = 0.15 Mbps on average. Ten million cameras can produce $10^7 \times 0.15$ Mbps = 1.5 Tbps. Over one day, 1.5 Tbps × 86,400 s ÷ 8 bits/byte = 16,200TB.

To store 16,200TB on 4TB disks, approximate 4,000 disks are needed per day. Over one year, 4,000×365 ≈ 1.5 million disks are needed. The entire hard disk industry ships about 500 million disks per year [5]. This does not include solid-state storage, commonly called flash memory. Thus, storing the data from ten million network cameras requires only 0.3% of the hard disks manufactured each year. This estimation may be too low because some network cameras have high resolutions or frame rates (or both). Even if this number is decupled, storing the data requires only 3% hard disks. One 4TB disk costs about $120 today and 1.5 million disks costs about $200M. Even though this is not cheap, it is technologically and

financially feasible if an organization decides to store the data from all cameras in the world. As cameras' resolutions increase, the total amount of data will also increase. As a result, 1.5 million disks may be insufficient. Even if the number is decupled, at $2B per year, the cost is still within the reach of many large companies.

e) Unexplored Opportunities

Even though the data from many network cameras (traffic cameras, national parks, etc.) are publicly available, the valuable information embedded in the visual data (including video and image) have not be fully explored. The unexplored opportunities occur due to many reasons as described as follows. First, some data are not archived. For example, the data from California's traffic cameras are not recorded [6]. Second, even if the data are stored and archives are available, the archives are scattered around the world. It is not easy to retrieve data from multiple sources that use different protocols and formats. Third, analyzing large quantities of visual data requires significant amounts of computing resources. For some applications, the analyses must be performed on-line while the data are captured. Traffic monitoring is an example. Most drivers would be interested knowing the locations of current accidents; few drivers would be interested in the locations of yesterday's accidents. Moreover, such analyses may be performed during only rush hours on weekdays. Some analyses are seasonable, such as studying snow coverage. It would be uneconomic to have dedicated computing and storage resources that are rarely fully utilized. It would be desirable to allocate the resources on demand. Thus, the motivation of this paper is to provide a cloud-based system to help explore the opportunities to harvest valuable information embedded in multimedia big data from multiple sources.

f) Contributions

This paper has the following contributions. First, it described the unexplored opportunities retrieving valuable information from the global camera networks. Second, it surveys some existing environmental studies using camera networks. Third, this paper presents the challenges obtaining valuable information from the camera networks. Fourth, this paper proposes a cloud-based system to solve some of the problems of analyzing multimedia big data in global camera networks.

II. Environmental Studies using Camera Networks

Cameras have been used by many researchers for studying various topics related to the environment [7]. Ruzicka *et al.* [8] use a webcam to monitor foam formation downstream of wastewater treatment. Winkler *et al.* [9] detect foam to measure water quality [10]. Goddijn-Murphy *et al.* [11] use the colors (optical properties) to evaluate the composition of water. Gilmore *et al.* [12] use cameras to measure water levels. Migiavacca *et al.* [13] use greenness from images to estimate CO_2 intake. Kataoka *et al.* use webcam images to detect colored macro plastic debris [14]. The Phenocams in the University of New Hampshire [15] contains 164 cameras and have shown the values of using cameras to monitor vegetation. Babari *et al.* [16] use cameras to estimate air visibility. Sawyer *et al.* [17] use webcams to teach geographical changes. The AMOS project [18] has retrieved more than 400 million images from 18,000 cameras since 2006. These studies indicate that camera networks can provide valuable information for observing and understanding the environment. Meanwhile, these studies also suggest restrictions of existing approaches using the data from camera networks. (1) Most studies use only a few cameras. (2) Each study has a specific goal and needs to obtain the data from the selected cameras. (3) All studies require low frame rates (several frames per day) to observe long-term trends. Many challenges arise at high frame rates. (4) Existing studies store the visual data for off-line analyses. It is challenging to perform on-line analysis. (5) Off-line analyses at low frame rates do not pose stringent requirements for computing and storage resources. Resource management would become essential when analyzing the streaming data from many cameras for on-line analyses at high frame rates. The following sections explain these challenges and a proposed solution to solve these problems.

III. Challengines in Harvesting Information from Camera Networks

To harvest the value of global camera networks for environmental studies, one must solve the problems mentioned above. First, there is a need of a central repository where researchers are able to find network cameras. Currently, if a researcher wants to utilize the camera networks from different research institutions, the researcher has to develop programs that can retrieve data from many different sources. For example, in USA, the departments of transportation in different states have different methods providing traffic data. Even for IP cameras, different brands require different paths for HTTP GET requests. Such heterogeneity is a challenge. Moreover, different institutions may configure the cameras differently. Some provide video streams of high frame rates and the others set the frame rates low. For some studies (such as phenology [19][20]), low frames are sufficient. For some other studies (such as monitoring traffic and observing wildlife), high frame rates are necessary. Researchers have to select the right

cameras for their studies. Thus, a central repository is required to help researchers find the appropriate cameras for their studies.

If a researcher wants to use the global camera network, the researcher has to

a) find the appropriate cameras and then retrieve data from these cameras
b) store the data for off-line analysis or perform on-line analysis on streaming data
c) manage computing resources
d) analyze the data to extract useful information

Among the four steps, only d) is specific to individual studies. The others are common for different studies. Even though many studies have demonstrated the values of the global camera network, we are unaware of any software framework that provides common functionalities for a)-c). As a result, every researcher has to repeat the similar steps in a)-c) and the true values of the global camera network have not been fully exploited. A preferred solution is a framework that solves a)-c), while providing an application programming interface (API) so that researchers can focus on writing computer programs for d).

Different types of studies have different requirements on the frame rates. Observing seasonal changes by leaf colors can be accomplished by several frames per day [21][22]. Observing a snow storm requires higher frame rates. Monitoring wildlife requires even higher frame rates. Even though many network cameras can provide videos, existing environmental studies do not use high frame rates partly because higher frame rates produce much more data and require more storage space. The larger quantities of data take longer to process.

As mentioned above, existing studies analyze archived visual data off-line [23][24]. Significant efforts are needed moving analysis programs from off-line post-event to on-line. Processing streaming data is challenging. Many applications can benefit from on-line processing and obtaining timely information, for example, detecting traffic congestion or wildfire. Some network cameras provide HD videos at 30 frames per second. At this frame rate, an analysis program must be able to process each frame within 33 ms. This is a stringent constraint for non-trivial image processing or computer vision programs. To meet this constraint, programs may adopt various strategies, such as dividing the programs into multiple stages and pipelining the stages. Parallel computing is another option [25]. We foresee that meeting the timing constraint would be one of the most serious barriers analyzing streaming data.

Resource management is yet another challenge for on-line processing of streaming data. Some environmental studies, such as detecting the arrival of spring by analyzing leaves' colors, are seasonal. Analyzing the data from traffic cameras may be valuable only during rush hours on weekdays. Some studies may be triggered by unexpected events, such as monitoring air quality after a volcanic eruption. These requirements make adaptive resource management essential.

IV. Cloud Computing

The challenges described above suggest that cloud computing would be suitable for harvesting the values of the data from the global camera networks. Cloud computing can allocate resources on demand and provide an economic method for studies that are seasonal or unscheduled. Researchers may launch cloud instances that have multiple cores for running parallel analysis programs for meeting the timing constraints. Cloud computing has the options of launching instances at preferred geographical locations. For example, Amazon Web Services are available in Beijing, Sydney, Singapore, Ireland, Frankfurt, Brazil, Oregon USA, and Virginia USA. Microsoft Azure is available in more than 10 locations. These locations provide options to researchers. Cloud storage can be used as data repositories shared for research communities.

The different geographical locations contribute to different round-trip time (RTT) between the cloud instances and the cameras. Even though RTT is not a linear function of the geographical distances, longer geographical distances usually have longer RTT. Long RTT may reduce the data rates when using TCP. This can be observed when using MJPEG. If higher frame rates are desirable, the cloud instances should be geographically closer to the data sources, i.e., the cameras. This imposes constraints on resource management. The principle is to "Move programs. Do not move data".

Moreover, the prices of cloud instances vary based on geographical locations. The effects of RTT and the frame rates have additional implications when researchers intend to consolidate computing resources and reduce the costs. If simple analyses are performed, a single cloud instance may suffice for handling the data from multiple cameras. However, if high frame rates are desired and the cameras are geographically scattered, it may be necessary to allocate multiple cloud instances that are closer to the cameras. Further investigations are needed for developing solutions that can meet the requirements of higher computing performance, higher frame rates, and lower costs.

V. Cloud-based System for harvesting Information from Camera Networks

To solve the problems mentioned above, we have been building a cloud-based system for harvesting the valuable information from camera networks. Without

the proposed system, the researcher has to tackle the two problems. First, the researcher must find the appropriate cameras. Second, the researcher must access the heterogeneous cameras with different types, brands, models, frame rates, etc. Our system provides a common platform to solve these problems on behalf of the researcher. This system has been operational for nearly one year. We have external collaborators outside our universities adding new features and external users testing this system.

A. System Description

This system has three components: (1) camera interface, (2) an application programming interface (API), and (3) cloud resource management [26]. The camera interface communicates with heterogeneous cameras from diverse sources. This interface hides the brands, models, and sources of the cameras. Researchers can select the cameras for their studies without knowing the details of the cameras. This interface can retrieve individual images from cameras in JPEG or videos in MJPEG. We are currently integrating H.264 into the system. The API is event-driven. A user provides a callback function that is invoked when a new frame arrives. This API completely hides the details of data retrieval. Users do not have to handle the different network protocols needed to retrieve data from heterogeneous cameras. This framework uses cloud instances for computing. A user may submit a program that analyzes images. This program is copied to cloud instances which retrieve data from the cameras and execute the analysis programs.

The procedure of using this system is described below:

1. Select the cameras for analysis according to researchers' need, such as location and time zone, as shown in Figure 3.
2. Set the execution configuration: the desired frame rate and the duration.
3. Upload an analysis program. The system has 16 pre-written analysis modules, as shown in Figure 4, for corner detection, motion detection, sunrise detection, etc. These modules serve the purpose of sample programs. The system currently supports Python with OpenCV. Researchers can write their own analysis modules with the API. Their programs may save results in a variety of formats, e.g. text and images.
4. Execute the program.
5. Download the execution results.

To simplify the use of this system, all of the above steps 1-5 of using this system can be done through the website, http:// cam2.ecn.purdue.edu/.

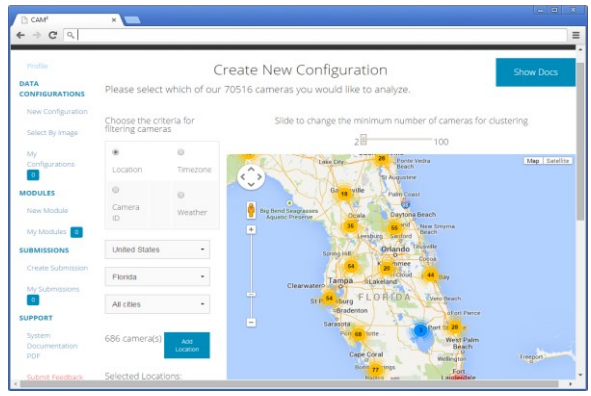

Figure 3. This figure shows the system has 686 cameras in Florida USA. A user may select cameras from different areas.

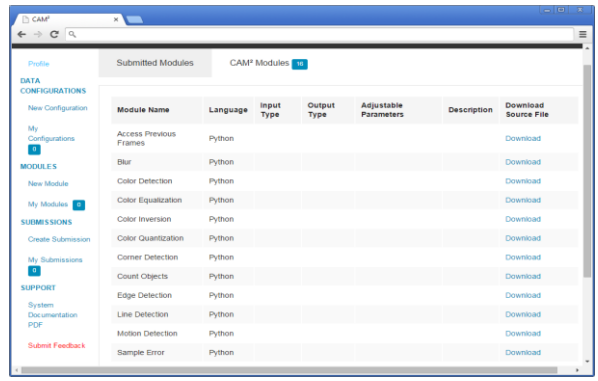

Figure 4. The system has 16 pre-written analysis modules. A user may use these modules as the basis for the analysis program.

B. Case Study

A simple case study, circle detection, is used to illustrate the procedure of using the proposed system in 5 steps through the web UI. In Step 1, the researchers can select the cameras for analysis by location and time zone as shown in Figure 3. In addition, the researchers can select cameras directly by image as shown in Figure 5.

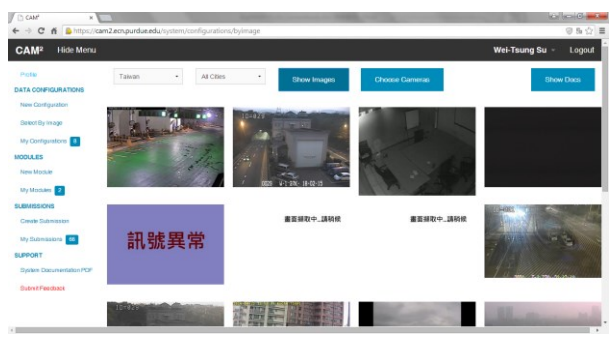

Figure 5 Select cameras directly by image

In step 2, the researchers can create a configuration which includes the desired duration and frame rate as shown in Figure 6.

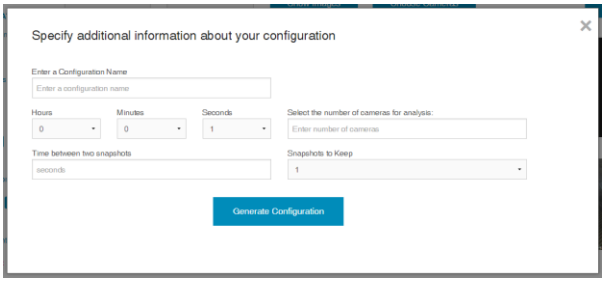

Figure 6 Configuration setup

In step 3, the researchers can upload their analysis programs using OpenCV-Python based on the proposed API. Here, we use circle detection as an example to illustrate the structure of an analysis program. An analysis program may import *FrameMetadata* and *CameraMetadata* classes if the information about this frame retrieved from this camera is required. Each analysis program in the proposed system must implement the *Analyzer* class, which is consisted of three methods as shown in Table 1.

a) The method *initialize* is called once at the beginning of the execution. The parameters or variables could be initialized in this method.

b) The method *on_new_frame* will be called every time a new frame is retrieved from the selected cameras.The main computer vision algorithm must be implemeted in this method.

c) The method *finalize* is called once after all frames are analyzed based on the configuration. The final calculation (such as summarizing the information from all frames) could be done and the final results can be saved as text files or images in this method.

Table 1 The structure of an analysis program

```
from analyzer import Analyzer
from frame_metadata import FrameMetadata
from camera_metadata import CameraMetadata

import datetime
import numpy as np
import cv2
import cv2.cv as cv

class MyAnalyzer(Analyzer):

  def initialize(self):
  """ Called once at the beginning """

  def on_new_frame(self):
  """ Called when a new frame arrives """

  def finalize(self):
  """ Called once in the end """
```

For circle detection, the parameters of Hough circle detection and variables helpful for final caculation are defined in initialize method as shown in Table 2. In on_new_frame method, it is simple to use Hough circle detection which is a built-in algorithm in OpenCV. To retrieve a frame, it is easy to call `self.get_frame()` without considering the brand and type of this camera. The researchers can obtaine the information of frame by calling `self.get_frame_metadata()`. Finally, the caculation, such as the total and average numbers of circles detected can be done and saved in finalize method. Then, the researchers can upload the analysis programs via web UI as Figure 7.

Table 2 Sameple program of circle detection

```
def initialize(self):
  # Initialize parameters
  self.BLUR = 25
  self.DIST = 50
  self.PARA1 = 50
  self.PARA2 = 30

  # Initialize values
  self.total_circles = 0
  self.total_frames = 0
  . . .

def on_new_frame(self):
  # Get frame
  frame = self.get_frame()
  . . .

  # Get frame metadata
  frame_metadata =
     self.get_frame_metadata()

  # Get date/time of frame, time is UTC
  date_time =
     frame_metadata.datetime.strftime(
        '%Y-%m-%d_%H-%M-%S')

  # Get camera id
  camera_id =
   frame_metadata.camera_metadata.camera_id
  . . .

  # counting circles
  . . .

  circles = cv2.HoughCircles(
     frame_gray, cv.CV_HOUGH_GRADIENT,
     1, self.DIST,
     param1=self.PARA1, param2=self.PARA2,
     minRadius=0, maxRadius=0)
     . . .

  # Save results
  filename =
     str(camera_id) + '_' + date_time
  self.save('results_' + filename + '.jpg',
     frame_annotated)

def finalize(self):
  # Calculate average number of circles
  avg_circles = float(self.total_circles) /
     self.total_frames

  # Put results in a string
  results_str += 'Average number of circles
      per frame:\t%.2f' % avg_circles
  . . .
```

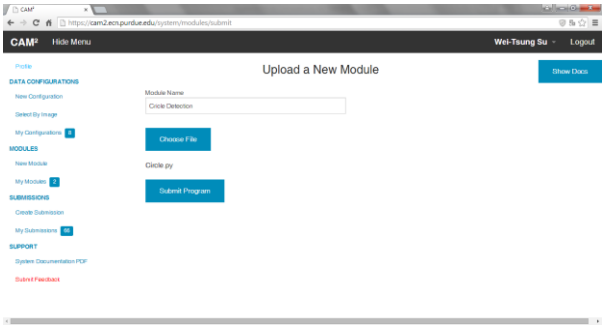

Figure 7 Upload an analysis program via web UI

In step 4, the researchers are able to execute the problem with their analysis programs with a specific configuration as shown in Figure 8. The progress of the execution will be dynamically updated in the web UI. After the execution is finished, the researchers can easily download the results for post analysis as shown in Figure 9.

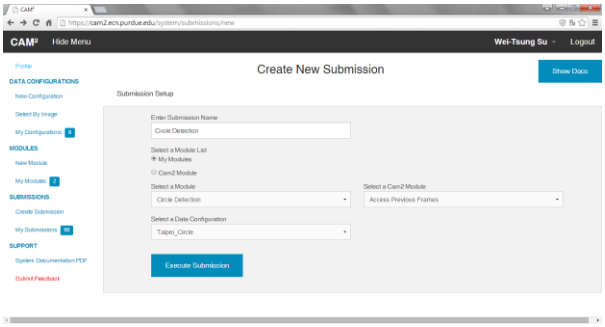

Figure 8 Execute the analysis program

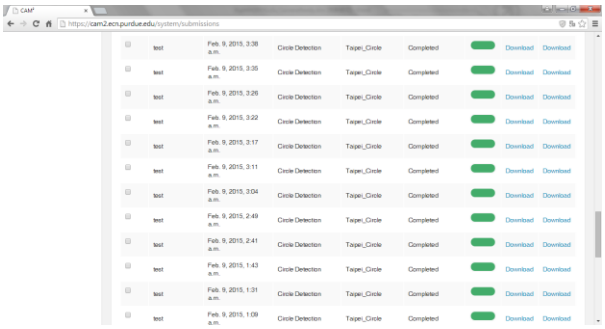

Figure 9 Download the results for post analysis

Figures 10 and 11 show another practical case study of object detections using this system. As can be seen in these two figures, the system is capable of detecting vehicles. This system can be used in studying transportation. This system has demonstrated the ability to simultaneously retrieve data from one thousand cameras at one frame every ten seconds and analyze the streaming data. The program uses 15 cloud instances and obtains data rate exceeding 100 Mbps [26].

Figure 10. This system can detect foreground object (a vehicle) by using background subtraction.

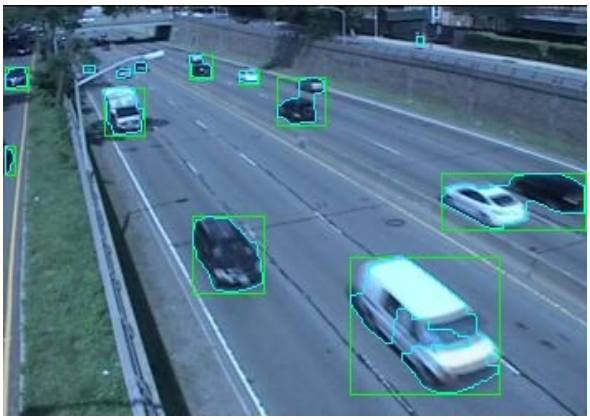

Figure 11. The system can detect moving objects. This figure marks the vehicles on a highway.

VI. Conclusion

In this paper, we propose a cloud-based system to harvest valuable information from camera networks. This system helps researchers easily and efficiently select, configure, and analyze the multimedia big data in camera networks. This system has demonstrated its convenience and efficiency. As described above, this system can analyze the multimedia big data from 1,000 network cameras at the same time. Moreover, the data rate exceeds 100 Mbps using 15 cloud instances. In the future, the problem of reducing cost and improving performance could be further studied while using cloud resources.

References

[1] Marketandmarkets.com. *Network camera and video analytics market*. September 2012. Report Code: SE 1238.

[2] YouTube.com. *YouTube Statistic*. Available: https://www.youtube.com/yt/press/statistics.html

[3] Cisco (June 2014). *Cisco visual networking index: Forecast and methodology*, 2013-2018.

[4] Netflix.com. *Internet Connection Speed Recommendations*. Available: https://help.netflix.com/en/node/306

[5] Western Digital Annual Report 2013.

[6] California Department of Transportation. *Frequently Asked Questions*. Available: http://video.dot.ca.gov/faq.htm

[7] John Porter, Chau-Chin Lin, David E. Smith, and Sheng-Shan Lu, ”Ecological image databases: From the webcam to the researcher,” *Ecological Informatics*, pp. 51–58, Jan., 2010.
[8] Katerina Ruzicka, Oliver Gabriel, Ulrike Bletterie, Stefan Winkler, and Matthias Zessner, “Cause and effect relationship between foam formation and treated wastewater effluents in a transboundary river,” *Physics and Chemistry of the Earth*, vol. 34, no. 8-9, pp. 565–573, 2009.
[9] S. Winkler, M. Zessner, E. Saracevic, K. Ruzicka, N. Fleischmann, and U. Wegricht, “Investigative monitoring in the context of detecting anthropogenic impact on an epipotamal river,” *Water Science & Technology*, vol. 57, no. 7, pp. 1023–1030, 2008.
[10] Heng Ma, Tsueng-Fang Tsai, and Chia-Cheng Liu, “Real-time monitoring of water quality using temporal trajectory of live fish,” *Expert Systems with Applications*, vol. 37, no. 7, pp. 5185–5171, Jul., 2010.
[11] Lonneke Goddijn-Murphy, Damien Dailloux, Martin White, and Dave Bowers. “Fundamentals of in situ digital camera methodology for water quality monitoring of coast and ocean,” *Sensors*, vol. 9, no. 7, pp. 5825–5843, Jul., 2009.
[12] Troy E. Gilmore, Francois Birgand, and Kenneth W. Chapman, “Source and magnitude of error in an inexpensive image-based water level measurement system,” *Journal of Hydrology*, vol. 496, pp. 178–186, Jul., 2013.
[13] Mirco Migliavacca, Marta Galvagno, Edoardo Cremonese, Micol Rossini, Michele Meroni, Oliver Sonnentag, Sergio Cogliati, Giovanni Manca, Fabrizio Diotri, Lorenzo Busetto, Alessandro Cescatti, Roberto Colombo, Francesco Fava, Umberto Morra di Cella, Emiliano Pari, Consolata Siniscalco, and Andrew D. Richardson, “Using digital repeat photography and eddy covariance data to model grassland phenology and photosynthetic CO2 uptake,” *Agricultural and Forest Meteorology*, vol. 151, no. 10, pp. 1325–1337, Oct., 2011.
[14] Tomoya Kataoka, Hirofumi Hinata, and Shinichiro Kako, “A new technique for detecting colored macro plastic debris on beaches using webcam images and CIELUV,” *Marine Pollution Bulletin*, vol. 64, no. 9, pp. 1829–1836, Sept., 2012.
2012.
[15] University of New Hampshire Phenocam. Available: http://phenocam.sr.unh.edu/webcam/.
[16] Raouf Babari, Nicolas Hautiere, Eric Dumont, Roland Brmond, and Nicolas Paparoditis, “A modeldriven approach to estimate atmospheric visibility with ordinary cameras,” *Atmospheric Environment*, vol. 45, no. 30, pp. 5316–5324, Sept., 2011.
[17] Carol F. Sawyer, David R. Butler, and Mary Curtis, “Using Webcams to Show Change and Movement in the Physical Environment,” *Journal of Geography*, vol. 109, no. 6, pp. 109:251–263, Nov., 2010.
[18] Nathan Jacobs, Nathaniel Roman, and Robert Pless, “Consistent Temporal Variations in Many Outdoor Scenes,” in IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[19] Wiebe Nijland, Rogier de Jong, Steven M. de Jong, Michael A. Wulder, Chris W. Bater, and Nicholas C. Coops, “Monitoring plant condition and phenology using infrared sensitive consumer grade digital cameras,” *Agricultural and Forest Meteorology*, vol. 184, no. 15, pp. 98–106, Jan., 2014.
[20] Eric Graham, Erin Riordan, Eric Yuen, Deborah Estrin, and Philip Rundel, “Public internet-connected cameras used as a cross-continental ground-based plant phenology monitoring system,” *Global Change Biology*, vol. 16, no. 11, pp. 30140–3023, Nov., 2010.
[21] Caroline A. Polgar, Richard B. Primack, Jeffrey S. Dukes, Crystal Schaaf, Zhuosen Wang, and Susanne S. Hoeppner, “Tree leaf out response to temperature: comparing field observations, remote sensing, and a warming experiment,” *International Journal of Biometeorology*, vol. 58, no. 6, pp. 1251-1257, Sept., 2013.
[22] Raimund Henneken, Volker Dose, Christoph Schleip, and Annette Menzel, “Detecting plant seasonality from webcams using bayesian multiple change point analysis,” *Agricultural and Forest Meteorology*, vol. 168, pp. 177–185, Jan., 2013.
[23] O. Sonnentag, M. Detto, R. Vargas, Y. Ryu, M. Kelly B.R.K. Runkle and, and D.D. Baldocch, “Tracking the structural and functional development of a perennial pepperweed (*Lepidium latifolium L.*) infestation using a multi-year archive of webcam imagery and eddy covariance measurements,” *Agricultural and Forest Meteorology*, vol. 151, no. 17, pp. 916–926, Jul., 2011.
[24] Nathan Jacobs, Walker Burgin, Richard Speyer, David Ross, and Robert Pless, “Adventures in archiving and using three years of webcam images,” in IEEE CVPR Workshop on Internet Vision, 2009.
[25] H. Hu, Y.G. Wen, T.-S. Chua and X.L. Li, “Towards Scalable Systems for Big Data Analytics: A Technology Tutorial,” *IEEE Access Journal*, vol. 2, pp. 652-687, Jul., 2014,
[26] Ahmed S. Kaseb, Everett Berry, Erik Rozolis, Kyle McNulty, Seth Bontrager, Youngsol Koh, Yung-Hsiang Lu, and Edward J. Delp, “An Interactive Web-based System Using Cloud for Large-Scale Visual Analytics,” in Imaging and Multimedia Analytics in a Web and Mobile World, 2015.