AUTOMATED SYSTEM FOR IDENTIFYING USABLE SENSORS IN A LARGE SCALE SENSOR NETWORK FOR COMPUTER VISION APPLICATIONS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Aniesh Chawla

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVED BY

Dr. Yung-Hsiang Lu, Chair

School of Electrical and Computer Engineering

Dr. David S. Ebert

School of Electrical and Computer Engineering

Dr. Edward J. Delp

School of Electrical and Computer Engineering

Dr. Pedro Irazoqui, Thesis Form Head

School of Electrical and Computer Engineering

ACKNOWLEDGMENTS

I take great pleasure in expressing my sincere gratitude towards my thesis advisor Prof. Yung-Hsiang Lu, who gave me the opportunity to work with him. I would like to thank him for the inspiration, innovative ideas, constant support and encouragement throughout the project. He has helped me in innumerable ways from the conception of the problem to its implementation. I also thank Prof. David S. Ebert and Prof. Edward J. Delp for their advice, critical insights and help in the project. I sincerely thank Abhinav Goel for his constant support and invaluable input. His timely help and advice made me move forward through this project. I would like to acknowledge Ryan Dailey, Andrew Liu, Joshua Majors, Sripath Mishra, and Ling Zhang. This work would not have been possible Without their help.

Finally, I would like to thank my wife, my parents, my family, and my friends. They have been my constant support and strength in this process. Without their help, it would have not been possible to reach this goal.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Chawla, Aniesh M.S., Purdue University, May 2019. Automated System for Identifying Usable Sensors in a Large Scale Sensor Network for Computer Vision Applications. Major Professor: Yung Hsiang Lu.

Numerous organizations around the world deploy sensor networks, especially visual sensor networks for various applications like monitoring traffic, security, and emergencies. With advances in computer vision technology, the potential application of these sensor networks has expanded. This has led to an increase in demand for deployment of large scale sensor networks. Sensors in a large network have differences in location, position, hardware, etc. These differences lead to varying usefulness as they provide different quality of information. As an example, consider the cameras deployed by the Department of Transportation (DOT). We want to know whether the same traffic cameras could be used for monitoring the damage by a hurricane. Presently, significant manual effort is required to identify useful sensors for different applications. There does not exist an automated system which determines the usefulness of the sensors based on the application. Previous methods on visual sensor networks focus on finding the dependability of sensors based on only the infrastructural and system issues like network congestion, battery failures, hardware failures, etc. These methods do not consider the quality of information from the sensor network. In this paper, we present an automated system which identifies the most useful sensors in a network for a given application. We evaluate our system on 2,500 real-time live sensors from four cities for traffic monitoring and people counting applications. We compare the result of our automated system with the manual score for each camera. The results suggest that the proposed system reliably finds useful

sensors and it output matches the manual scoring system. It also shows that a camera network deployed for a certain application can also be useful for another application.

1. INTRODUCTION

Consider a scenario: Houston is enduring a hurricane and many parts of the city are flooded. There are hundreds of cameras deployed by Houston Department of Transportation [1] providing real-time images. First responders could leverage the information from these cameras to monitor the flooding situation. Consider another scenario: There is a big parade in Manhattan. A boy who comes with his family to watch the parade, gets lost. Satyanarayanan [2] discusses the possibility of finding the boy in the crowd using crowd sourced images from mobile phones. However, instead of waiting for people to send images, authorities can also use traffic cameras to search for the boy. Both these scenarios require an automated system which quickly finds set of useful cameras for the intended application. We design such a system called Method to Analyze Geo-tagged Images for Camera Classification (MAGICC), which takes input as a list of cameras and an application like identifying flooding, finding a person, traffic congestion, etc. and finds the useful cameras for the intended application as shown in Fig. 1.1.

First responders and various organizations are also aided by the current increase of deployment of sensors. This increase is due to low-cost availability of visual sensors along with improvements in computer vision technology. It is expected that in the next decade there will be substantial growth in the deployment of visual sensor networks, specifically camera networks [3]. In this paper, we focus only on one type of a visual sensor network, i.e., a camera network. These camera networks are managed by numerous organizations for various applications such as monitoring traffic, tourism, forest, etc. Such diversity in cameras creates differences in two main characteristics of a camera, i.e., resolution and frame rates. Fig. 1.2 shows such diversity in resolution and frame rates for camera networks in four cities. Fig. 1.2(a) shows cameras in city1,

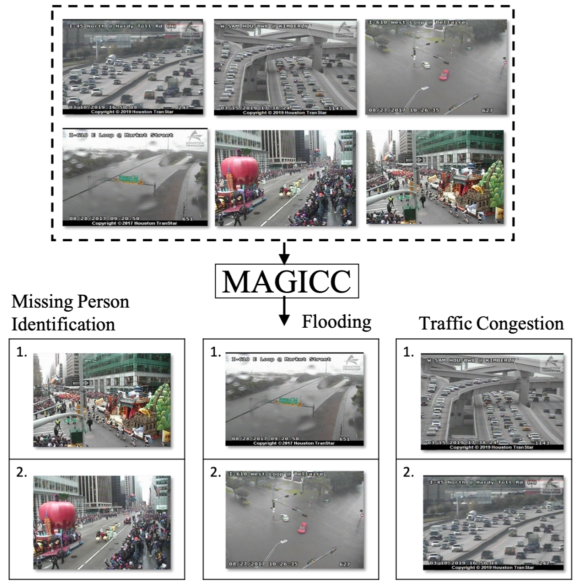
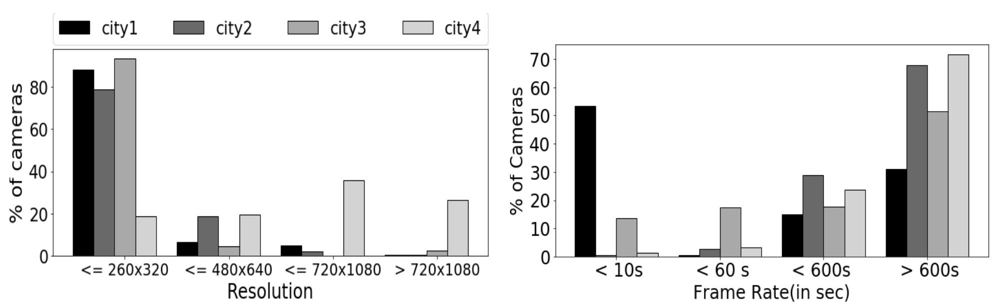


Figure 1.1. MAGICC: Method to Analyze Geotagged Images for Camera Classification takes input as a list of cameras, and a required computer vision application. It outputs a rank list of the most useful cameras for that particular application.

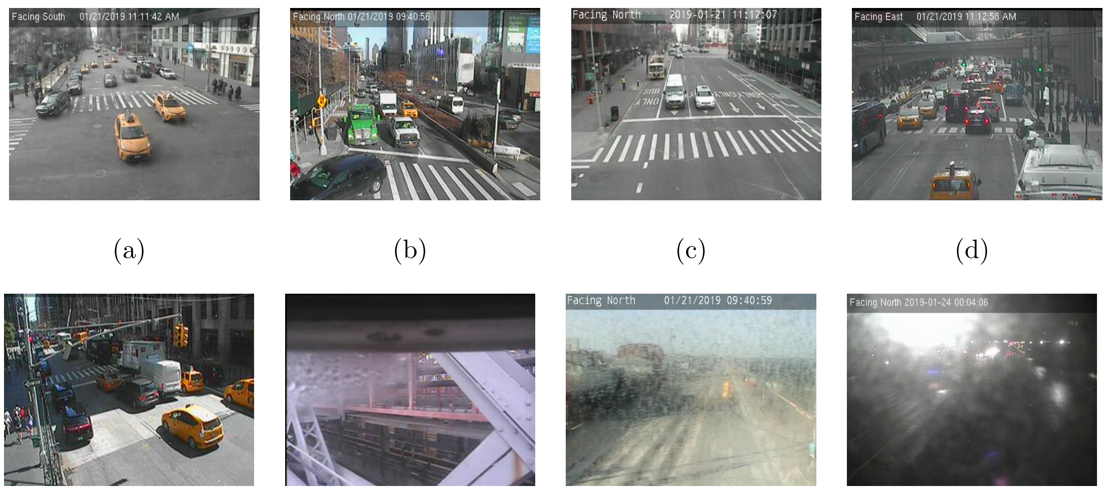
city2, and city3 have low resolution, and the cameras in city2 and city3 have higher frame rates compared to cameras in city1 as shown in Fig. 1.2(b).



(a) (b)

Figure 1.2. Diversity in (a) Resolution and (b) Frame rates of cameras across different cities. This shows that cameras have different hardware across cities.

Diverse cameras, along with the differences in their resolutions, frame rates, locations, and orientations, lead to differences in their usefulness. We determine the usefulness of a camera based on the quality of information it provides for the intended application. For example, consider the cameras deployed by the New York City Department of Transportation in Fig. 1.3. Among these, the cameras (a)-(e) provide quality information on the traffic flow. The cameras (f) and (g) are blocked or blurry; therefore, they provide no valuable information for traffic monitoring. With the development in computer vision technology, computer vision programs can be used to extract the information from the camera images.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1.3. Example of Cameras in New York City Department of Transportation network showing that all cameras do not provide useful information for traffic monitoring application.

Previous studies on visual sensor networks do not consider information from the sensors in their analysis. These studies are evaluated on either a small sensor network [4] consisting of less than ten nodes or a simulated sensor network. These studies propose methods to find reliability and availability [5–9] based only on the infrastructural and system issues like low battery [8, 10], network congestion [11, 12], hardware failures, etc. Such methods do not incorporate the quality of information to quantify the usability of the sensor network. Thus, presently it requires a significant amount of laborious manual work to find the useful cameras in the network.

In this paper, we propose a system to analyze geo-tagged images for classifying cameras. MAGICC leverages quality of information for an intended application along with infrastructure and system conditions to find usability of a camera. Quality of information depends on the application requirements, e.g. for a traffic monitoring application, it is required to detect vehicles from the images. Therefore, we use vehicle detection as a measure to quantify the quality of information for a traffic monitoring application. Similarly, any people counting application requires identification of people in the image; therefore we use detection of people as a measure to quantify the quality of information for such an application. To detect a vehicle(s) or a person(s), MAGICC uses YOLOv3 [13], a state-of-the-art object detector. We evaluate MAGICC on 2,500 real time cameras from four cities. The results shows that MAGICC is effective in finding useful cameras. The output from MAGICC shows that a camera network which is setup for an entirely different purpose can be useful for other applications as in the scenario discussed above where traffic cameras may be used to find a missing person. We further show the variation in spatial distribution of the cameras and the data from the camera effect the usability of the camera.

2. RELATED WORK

Table 2.1.
Comparing various studies on sensor networks to our system, MAGICC. None of the studies consider information from the sensor as a parameter to consider the usefulness of sensor networks. These studies are also done on a very small scale(< 10 cameras) or on a simulated environment.

Methods	Parameters in Sensor Model			Large Scale Testing
	Network Congestion	Hardware Failure	Information from Sensor	
Liang et al. [11]	✓			
Zheng et al. [12]	✓			
Silva et al. [5]	✓			
Costa et al. [14]		✓	✓	
Jesus et al. [6]	✓	✓		
Tusher et al. [15]		✓		
Costa et al. [8]	✓	✓		
Munir et al. [9]		✓		
MAGICC	✓		✓	✓

2.1 Infrastructure and System

Silva et al. [5] propose a method to evaluate reliability and availability for industrial setup based on network failure conditions and faults in the sensors. Mahmood et al. [4] discuss various methods to improve reliability of data transmission in a sensor network. Elghazel et al. [16] define reliability as continuous information being sent by the sensors and consider data loss only as a major factor. Liang et al. [11] improve availability by reducing packet losses in resource constraint sensor networks. Zheng et al. [12] propose a method to increase the life of sensors and thereby their availability by reducing their power consumption.

Tusher et al. [15] propose a method to reliably detect faults in a sensor network by using fall curve of the voltages. Lajara et al. [10] estimate the state of health of sensor networks through the battery-related information. Costa et al. [14] find availability issues of sensors based on node failures and field of view of the visual sensors. Bruneo et al. [7] estimate reliability of the sensor network by taking reliability of a group of sensors rather than a single sensor. Costa et al. [8] assess availability of a sensor network for monitoring a target. They consider failure conditions as network and battery failures only.

2.2 Markov Process and Reliability Improvement

Jesus et al. [6] model cameras using Continuous Time Markov Chain (CTMC) based on hardware and battery states of wireless visual sensors. They use Fault Tree analysis and find dependability for wireless visual sensor networks. Bruneo et al. [17] also propose a Markov model for sensor networks using CTMC, based on battery discharge to find reliability of sensor networks. Aditi et al. [18] estimate state of a sensor using Linear Mixture Model to detect an anomalous node in the system. SanMiguel et al. [19] create a unifying framework to model a camera with three states. They also propose a method to reduce the energy consumption of the camera in order to improve its reliability. Munir et al. [9] propose a Markov based camera model to find reliability of sensor networks based on sensor failures.

2.3 Emergency Response and Surveillance

Quality of information is immensely important for applications like emergency response, surveillance, etc. Koh et al. [20] show that public camera networks are useful for public safety, but they quantify the usefulness. Semertzidis et al. [21] present a system for automatic traffic monitoring by including multiple data sources. Cayirci et al. [22] propose a sensor architecture for rescue operations. In this architecture, sensors are randomly deployed to detect individuals needing rescue. They perform simulations to test their architecture.

2.4 Camera Positioning

Song et al. [23] propose a system for dynamically changing camera network reconfiguration to optimize scene-analysis. Erdem et al. [24] outline a method to cost effectively place camera for improving coverage for an application. Hengel et al. [25] describe a method to optimally place cameras in a building for improving surveillance. Kritter et al. [26] discuss various optimal camera placement solutions.

2.5 Our Contribution

Table 2.1 compares our Method to Analyze Geo-tagged Images for Camera Classification, MAGICC, to previous studies on sensor networks. MAGICC analyzes the sensor network on systems and infrastructural conditions like network congestion, hardware failure, and the quality of information from the sensor. It is evaluated on real-time camera networks from four cities consisting of 2,500 cameras. The contributions of this paper are as follows:

1. This paper presents an automated system, MAGICC, that measures the usefulness of a camera for a given application.

2. MAGICC provides a list of most useful cameras for a given application.

3. To our knowledge, it is the first large scale study on more than 2,500 live cameras generating real-time data.

3. STATE TRANSITION MODEL OF A CAMERA

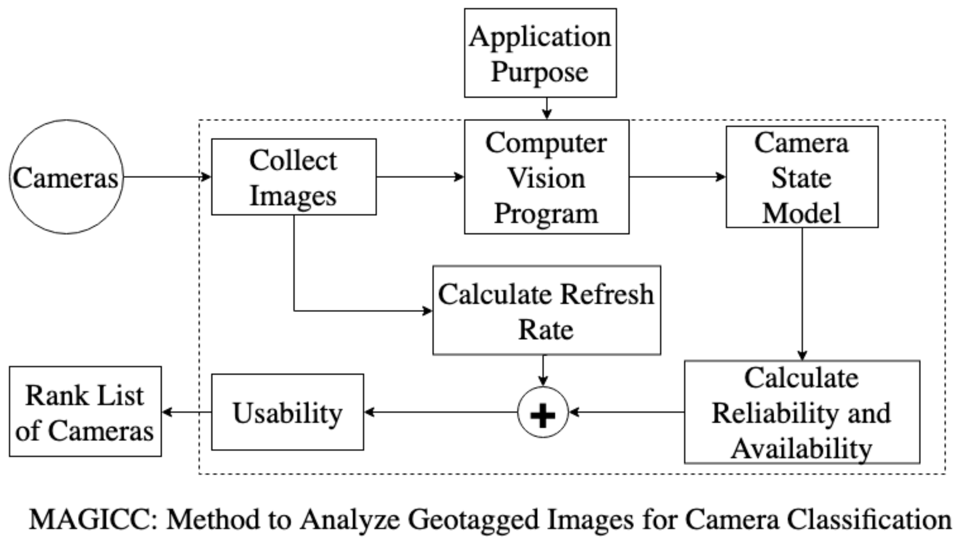


Figure 3.1. Flow Chart of MAGICC to find the usability of a camera. It takes a list of cameras as an input. It uses YOLOv3 to detect car and people from the images collected from the camera. It calculates usability for each camera to output a ranked list of most useful cameras.

A camera network is installed for an intended application e.g. traffic monitoring, surveillance, etc. As cameras are diverse in a large network, all cameras may not provide the same quality of information required for the intended application. MAGICC incorporates intended application requirements as a measure to define usability of a camera as shown in Fig. 3.1. It models the camera into various states based on whether it can get quality information from the camera. A camera moves between these states with a certain set of rates of transitions. These rates also form an integral part of its transition model. Every camera has its own set of transition rates. MAGICC calculates these rates from state transitions probability distributions for each camera. It does not assume any specific probability distribution, but identifies the best distribution using various statistical tools. In order to find the best distribution, MAGICC collects the images from the cameras as shown in Fig. 3.1. It collects images for a period of 7 days as to remove any single day event like power outages, sports event, etc. which could lead to change in the usability of a camera.

3.1 Camera States

We create a state model of a camera based on the quality of information we receive from the camera. The quality of information is measured by counting the number of application- specific detections. For example, the quality of the information from a camera for traffic monitoring would be determined by if a vehicle is detected.

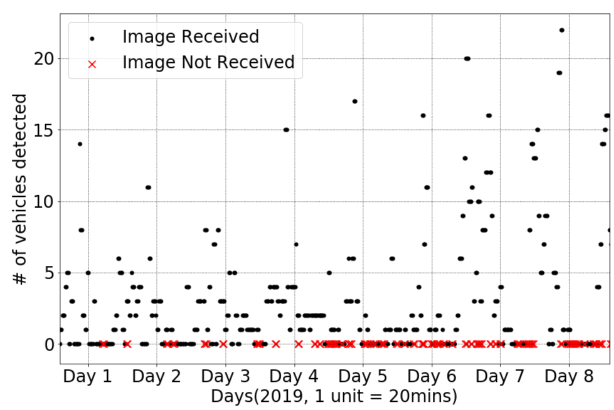


Figure 3.2. Number of vehicles detected over time by one of the city1 cameras. The camera exists in three states: (1) One or more vehicles detected (2) Vehicle not detected (3) Image not received.

The model of the camera consists of three states describing the type of information received from a camera. These three states are *Useful*, *Not Useful*, and *Down*. These states of a camera are also found when analyzing camera a for people counting application. We define these states of a camera as follows:

1. *Not Useful*: A camera is in this state if the system receives an image from the camera but the image is lacking information in regards to the application.

2. *Useful*: A camera is in this state when our system can receive an image from the camera, and the image has quality information.

3. *Down*: A camera is in this state when the system does not receive any image from the camera or it receives an image which is exactly the same as the previous image received. This state occurs because of infrastructure or system issues e.g. network congestion, hardware issue, power outage, battery down, etc.

Fig. 3.2 shows these three states of a camera for a traffic monitoring application. It shows that a camera is in the *Not Useful* state when an image was received from a camera but there is no vehicles detected by YOLOv3; a camera is in *Useful* state when there is at least one vehicle detected; a camera is down when a image is not received from the camera.

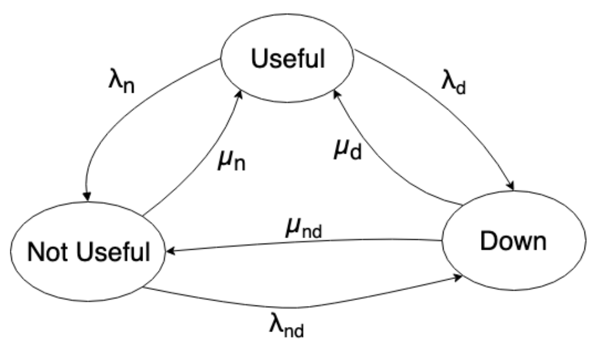


Figure 3.3. Camera model showing transition of a camera between three states over time. λ_i's and μ_i's are the rates of the transition from one state to another

From Fig. 3.2 we observe that a camera can transition to any state irrespective of its previous state. Thus, we establish a state diagram of a camera as shown in Fig. 3.3. All the λ_i's and μ_i's in Fig. 3.3 are the rates of state transitions of a camera. Though Fig. 3.3 resembles a Markov process we cannot assume that the model follows a Markov process. The system should be memoryless for it to be a Markov process. For a memoryless system the probability of the states comes from an exponential distribution. Fig. 3.4 shows the probability plot for each of the states of a camera

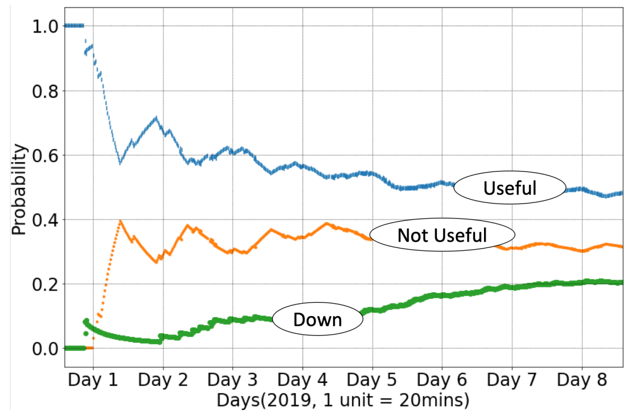


Figure 3.4. Probability of three States of a camera over time. As time increases, State Probabilities becomes constant.

over time. To show that the system probabilities are exponential, we fit an exponential function to the probability of states of a camera. We then evaluate the curve fit using coefficient of determination (R^2) method. R^2 is a statistical tool which tells if the fraction of variance in the data sample can be explained by the testing function. In our case we use exponential function as a testing function. A value of R^2 close to 1 means that an exponential function fits the data correctly and a value close to zero represents the function does not fit the data. Table 1 (in Appendix) shows that the R^2 values for the exponential function on the set of cameras is close to 1. This means the exponential function is a valid approximation of the data.

3.2 State Transition Probability Distribution

In order to find the usability score, we need to determine the rates for state transitions. These rates come from probability distribution of the state transitions in the model of a camera. We use the Kolmogorov-Smirnov (K-S) test [27] to find the best distribution(s) for the states transition. K-S test is a hypothesis test for verifying if different samples of data come from a same distribution. The following hypothesis test is used in K-S test:

- Null Hypothesis, H_0: The sample data comes from the testing distribution

- Alternate Hypothesis, H_1: The sample data does not come from the testing distribution

K-S test outputs a p-value for the Null hypothesis. We reject the Null Hypothesis (H_0) when the p-value is less than 0.05. In our analysis, we test the top 12 distributions which occur most prominently for sensor analysis. Table 2 (in Appendix) shows the list of distributions and the p-values of K-S test on a random sample of 10 cameras from the city1. The K-S test results in Table 2 shows that only four distributions, i.e., Exponential, Bradford, Uniform, and Weibull$_{min}$ pass the p-value threshold. Out of these four distributions, the best distribution is the one with the least error from fitting a distribution on the data samples. We use the least error method as we cannot select the best distribution just based on a higher p-value. Table 3 (in Appendix) shows that the exponential distribution has the lowest error even though it doesn't have the highest p-values. Thus, we accept exponential distribution is the best probability distribution for the transition rates of states for cameras. We verify this result on all 4 cities.

4. USABILITY OF A CAMERA

The quality of information a camera network provides is based on its spatial distribution and whether its cameras can provide quality images in both day and night. Given a camera network with a fixed spatial distribution and a computer vision application, we want to identify the usability of a camera for that application.

We define the usability of a camera by three aspects: (1) **Reliability**: Duration for which the camera continuously sends quality information; (2) **Availability**: Duration after which the camera returns to sending quality information from either not sending the data or sending data with no information; and (3) **Refresh Time**: Duration after which a camera updates the data.

As stated earlier the quality of information depends on the application. For example, in Fig. 1.1 for detecting floods, water detected on the road is quality information; whereas for a traffic congestion application, vehicles detected on the road is quality information. For the use cases in this paper, we define detecting a vehicle and detecting a person in an image as a measure to quantify the quality of information for monitoring traffic and person counting applications respectively.

MAGICC models the cameras into three states described in previous section. It finds the rates of state transitions for each camera by finding the probability distributions of the state transitions. These rates are then used to calculate reliability and availability of a camera. MAGICC additionally finds refresh time for each camera. It then produces a usability score as a function of reliability, availability, and refresh time. MAGICC uses this score to rank the cameras for the input list. This score is also used to compare the cameras both across networks and within a single network. We make this comparison to find the effect of spatial distribution on the usability of a camera.

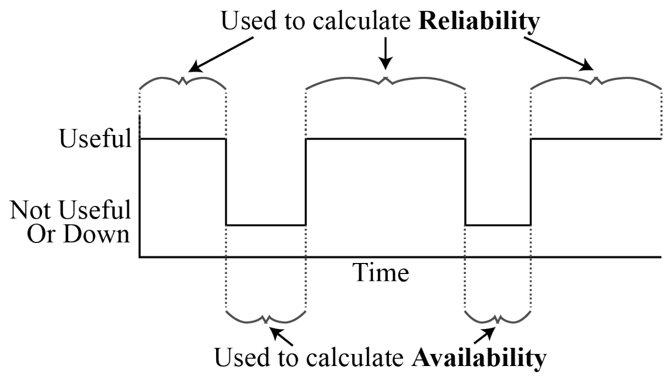


Figure 4.1. The graph illustrates the definition of reliability and availability. The duration that a camera provides useful data contributes to the reliability of the camera, whereas the availability is calculated using the times that a camera spends in the not useful or down states. A camera has higher availability if it spends less time in these states

4.1 Reliability

This aspect gives us the duration for which a camera gives quality information. Consider Fig. 1.3(f), this camera sends data but the information is not relevant to traffic monitoring application. This camera is not reliable for the purpose of traffic monitoring. Fig. 4.1 shows that the time a camera spends in the *Useful* state contributes to the reliability score. Reliability at time t is defined by Rausand et al. [28] as the probability that the camera will remain in the *Useful* state at time $T(> t)$ given that camera is in the *Useful* state at time t.

$$R(t) = P(\mathit{Useful}(T) > t) \tag{4.1}$$

In simpler terms it determines the duration that a camera remains in the *Useful* state. Based on the state model in Fig. 3.3, camera can transition to either *Not-Useful* or *Down* state. Therefore, we take probability of the minimum time after which a camera either goes to the *Not-Useful* or to the *Down* state.

$$P(T > t) = P(min(T_n, T_d) > t) \tag{4.2}$$

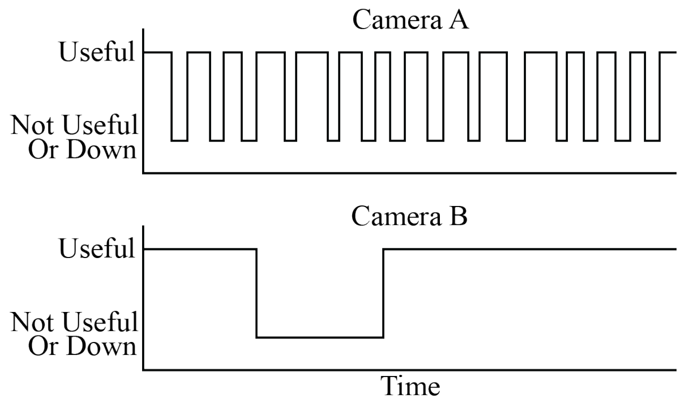


Figure 4.2. Two hypothetical cameras that transition between *Useful* and *Not Useful/Down* state over time. Camera A has low reliability because it provides useful information for a short duration of time but it has high availability as it quickly comes back to sending useful information; whereas camera B has high reliability as it stays in *Useful* state for a longer duration but it has low availability as it also remains in *Not Useful/Down* for a long duration.

where T_n and T_d are time after which a camera transition from the *Useful* state to *Not-Useful* and *Down* state respectively. The model follows a Markov process and state transitions are independent. Therefore Equation (4.2) can be further written as,

$$\begin{aligned} P(min(T_n, T_d) > t) &= P(T_n > t \cap T_d > t) \\ &= P(T_n > t) \cdot P(T_d > t) \end{aligned} \tag{4.3}$$

From the previous section, the probabilities of the state transitions are from the exponential distribution. Therefore, Equation(4.3) translates to

$$\begin{aligned} P(min(T_n, T_d) &= e^{-\lambda_n(t)} \text{ x } e^{-\lambda_d(t)} \\ \Longrightarrow \; R(t) &= e^{-(\lambda_n + \lambda_d)t} \end{aligned} \tag{4.4}$$

Large camera networks are deployed to be used for long term. Therefore, it is important to find the steady state time duration for which a camera remains in *Useful* state. Fig. 3.2 shows that the camera can recover from any of the *Not-Useful* state to the *Useful* state. Therefore, we define the steady state duration in *Useful* state as Mean Time between Failure (MTBF). Equation (4.5) shows derivation of MTBF from reliability of a camera.

$$\begin{aligned} MTBF &= \int_0^\infty R(t)dt \\ &= \frac{1}{\lambda_n + \lambda_d} \end{aligned} \tag{4.5}$$

A camera which has high MTBF value means that the camera remains in the *Useful* state for a longer period and therefore is more useful. Fig. 4.3 shows the MTBF values for the cameras in the four cities. It shows that MTBF values are continuous. In order to compare these values to availability, and refresh time values (discussed in later sections), they need to be normalized between 0 and 1. MAGICC uses max-min method to normalize MTBF values. Fig. 4.3(a) shows that more than 90% of the cameras have MTBF values less than 60 mins but the maximum value of MTBF found from the analysis is 1900 mins. Since max-min method of normalization is susceptible to the outliers. Therefore, MAGICC removes the outliers from the MTBF values by using Tukey method [29] before it normalizes the MTBF values. Tukey method uses first and quartiles from the data sample to find outliers. We remove these outliers so that they do not skew our result. We found that for MTBF values there are 270 cameras out of 2,500 cameras are outliers. Fig. 4.3(b) shows the distribution of MTBF values in our data sample after removing outliers. Equation (4.6) defines the reliability score derived from MTBF values of a camera.

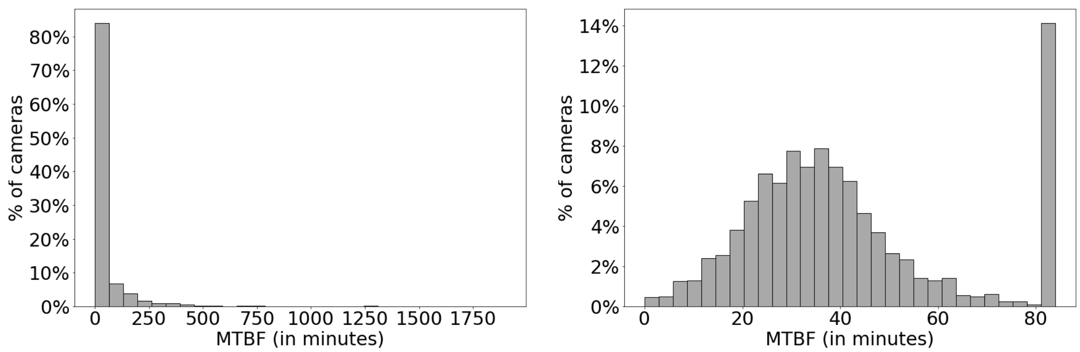
$$\text{Reliability score }, RE_i = \frac{MTBF_{ci} - \min(MTBF_c)}{\max(MTBF_c) - \min(MTBF_c)} \tag{4.6}$$

where,

$$MTBF_{ci} = \text{MTBF of the ith camera of a network}$$

$$\max(MTBF_c) = \text{max MTBF in a camera network}$$

$$\min(MTBF_c) = \text{min MTBF in a camera network}$$



(a) (b)

Figure 4.3. MTBF value distribution for cameras (a) With outliers (b) After removing outliers using Tukey method

4.2 Availability

Availability of a camera tells about the rate of recovery of a camera to a *Useful* state from any other state. Fig.4.1 shows the duration of time a camera spends in the *Not Useful* or *Down* state is used to calculate the availability score. It is different from reliability of a camera as a camera which has lower reliability can still be more useful if it quickly recovers from either *Not Useful* or *Down* states as shown in Camera A in Fig.4.2. Availability at a time t is defined by Huang et al. [30] as probability that a camera is in the *Useful* state at that time instant t. Mathematically, it is as follows:

$$\text{Availability, } A(t) = P(\text{Useful}(t))$$

As shown in the previous section that the states of a camera model have exponential probabilities. Therefore, as time progresses the probabilities of the states in the camera model become constant and their derivatives become zero. Fig. 3.4 shows that probability of the three states of a camera becomes constant with time. Equation (4.7) shows probability vector $P(t)$ of three states of a camera, where P_u, P_n, P_d represents probabilities of *Useful*, *Not-Useful*, and *Down* state respectively.

$$P(t) = [\ P_u(t) \quad P_n(t) \quad P_d(t)\ ] \tag{4.7}$$

We take the derivative of $P(t)$,

$$\frac{\partial P(t)}{\partial t} = P(t) \cdot M(t)$$

where,

$$M(t) = \begin{bmatrix} -(\lambda_n + \lambda_d) & \lambda_n & \lambda_d \\ \mu_n & -(\mu_n + \lambda_{nd}) & \lambda_{nd} \\ \mu_d & \mu_{nd} & -(\mu_d + \mu_{nd}) \end{bmatrix}$$

For steady state condition, $\frac{\partial P(t)}{\partial t} = 0$. Solving for steady state probability, P_u, for the availability of the camera in the *Useful* state.

$$P_u = \frac{\mu_n \mu_{nd} + \mu_n \mu_d + \mu_d \lambda_{nd}}{A + B + C} \tag{4.8}$$

$$A = (\lambda_n + \lambda_d)(\mu_{nd} + \lambda_{nd})$$

$$B = \mu_n(\mu_d + \mu_{nd} + \lambda_d)$$

$$C = \mu_d(\lambda_n + \lambda_{nd})$$

As P_u is a probability and its value is between 0 and 1. It is used as the availability score, A_i, for a camera.

$$A_i = P_u \tag{4.9}$$

4.3 Camera Refresh Time

This is the third important aspect which we use to evaluate the usability of a camera. This is the time duration after which a camera updates the data. As discussed earlier, we use networks of cameras hosted on various websites for the four cities. These websites do not provide direct access to these cameras. They provide access to these cameras by via snapshots from these cameras. These snapshots get updated

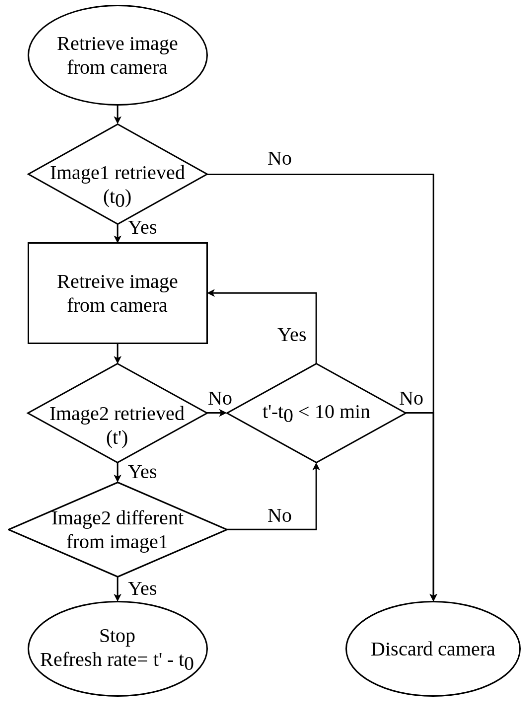


Figure 4.4. Flow Chart for calculation of Refresh Time from a camera

after certain duration known as refresh time (RT). Fig. 1.2(b) shows different refresh times of the cameras across various cities.

Fig. 4.4 shows how to calculate the camera's RT, the system downloads a reference image. The reference image indicates the initialization stage. The system removes the cameras which do not send any information. Every second it polls the remaining cameras to capture images from the website. It polls for 10 mins (empirical threshold) or when it receives an image which is different from the reference image, whichever is lower.

A higher value of refresh time means that the camera sends outdated information, which makes the camera to be less useful. Therefore, refresh time is inversely related to the usefulness of a camera. Fig. 4.5 shows refresh time of 40% cameras is less than 25 seconds. It also shows that the refresh times are continuous values. In order to compare it with reliability and availability scores, we normalize refresh time using the max-min method. Similar to MTBF values, the system removes outliers(if any)

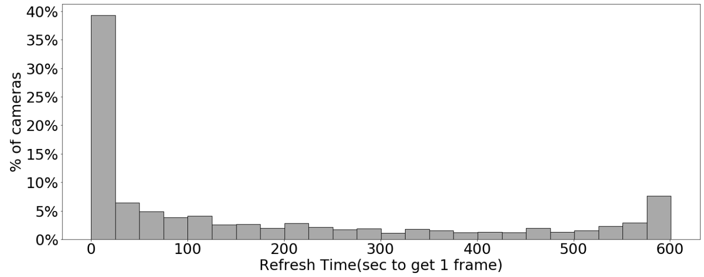


Figure 4.5. Distribution of refresh times in the four cities. About 40% of the cameras have refresh time less than 25sec.

from the set of refresh time values using the Tukey method before it normalizes the values. In the refresh time values we obtain from these 4 cities we do not observe any outliers but we have kept Tukey method in case a camera network has outliers.

$$\text{Refresh Time score, } RS_i = 1 - \frac{RT_i - RT_{min}}{RT_{max} - RT_{min}} \tag{4.10}$$

where,

$RT_i =$ refresh time of ith camera in network

$RT_{min} =$ min of the refresh times in network

$RT_{max} =$ max of the refresh times in network

4.4 Usability

In a large camera network, all the cameras are not equally useful. A camera is most useful if it sends the most updated quality information. Availability(A_i) and reliability(RE_i) scores quantify the quality of information and refresh time(RS_i) score quantifies duration after which data is updated from a camera. Therefore, usability score is a function of these three scores.

The usability function may change with the application requirements. For a real-time object tracking application the refresh time and reliability of the camera should be weighted more. Whereas, a warehouse inventory counting application may require to have better reliability and availability. In our use cases, traffic monitoring and emergency response applications, refresh time, reliability and availability are equally important. A low score in either the refresh time or availability should not reduce the usability of a camera for such applications. Therefore, to maintain the applications requirement and keep the scoring function simple, we define the Usability score as a linear combination of availability, reliability and refresh time scores as shown in Equation (4.11).

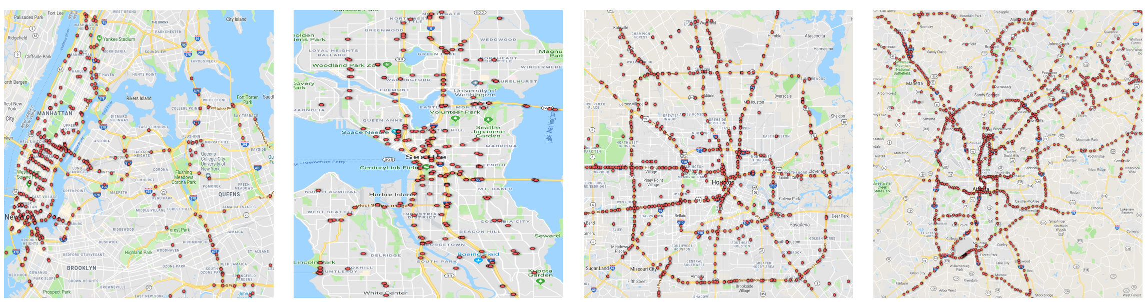
$$\text{Usability score, } U_i = w_1 \cdot RE_i + w_2 \cdot A_i + w_3 \cdot RS_i \tag{4.11}$$

where, w_is are weights for each score. Since we give equal importance to each score, we take w_is to be all equal to $\frac{1}{3}$. A Usability score of 1 means that the camera is most useful, whereas a score near 0 means that the camera does not perform well for the given application.

The reliability and refresh time scores in the usability score function are normalized across all four camera networks before they are used. As discussed earlier these scores are normalized by using the max-min method so that all the scores can be compared against each other.

5. EVALUATION

Not all the cameras in a network provide quality information. MAGICC finds the usability of each camera based on the intended application. We evaluate MAGICC on 2,500 cameras from four cities. Fig. 5.1 shows the spatial distribution of these cameras in these four cities. We do not have control over the location, position, or orientation of these cameras. Therefore, we do not control the quality of information we receive from them. Fig. 5.2 shows examples of the cameras in city1, city2, city3, and city4. It shows that the cameras in city1 are distributed in areas that are both high population and traffic dense; whereas, cameras in city2, city3, and city4 are located on the highways.



(a) city1 (b) city2 (c) city3 (d) city4

Figure 5.1. Spatial distribution of cameras in 4 cities. city2, and city3 have cameras on the highways but the city1 cameras are placed in densely populated area as well

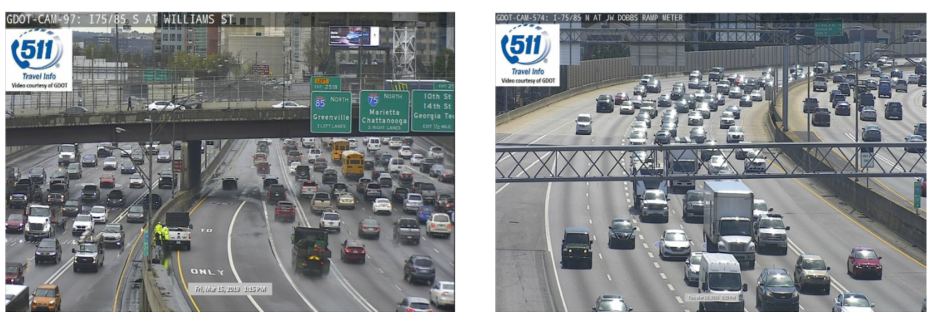
5.1 Experimental Setup

We use the camera networks from the 4 cities which are available to anyone with an internet connection. These cameras are hosted on websites that periodically update

(a)

(b)

(c)



(d)

Figure 5.2. Examples of the city1, city2, city3, and city4 cameras. City1 cameras are located in areas of dense vehicle and population densities. City2, city3, and city4 cameras are located on the highways.

the image collected from each camera. MAGICC collects these images at a sampling rate that is lower than the periodic update rates of the websites. Table 5.1 shows the total number of cameras and cameras from which we are able to obtain images for each city. The images from the responsive cameras are collected over a period of 7 days.

Table 5.1.
Distribution of number of cameras per city from which images can be collected.

cities	# cameras	# cameras (responsive)
city1	604	454
city2	601	367
city3	1,037	561
city4	1,430	1,056

Fig. 5.3 shows the experimental setup for evaluating MAGICC. It finds both the numbers of vehicles and persons detected over time for each camera in the list of 2,500 cameras from four cities. It then uses these values as input to the K-S test to obtain the parameters of the exponential distribution for each state transitions for the camera. MAGICC calculates reliability and availability from these parameters using Equation (4.6) and Equation (4.9) respectively. MAGICC also calculates the refresh time scores separately from the cameras. MAGICC uses Equation (4.11) to find usability score for each camera and it creates a rank list of cameras using this usability score.

5.2 Model Evaluation

We evaluate MAGICC by comparing it's scores with the scores we get by manually scoring each camera for the two applications: traffic monitoring and people counting.

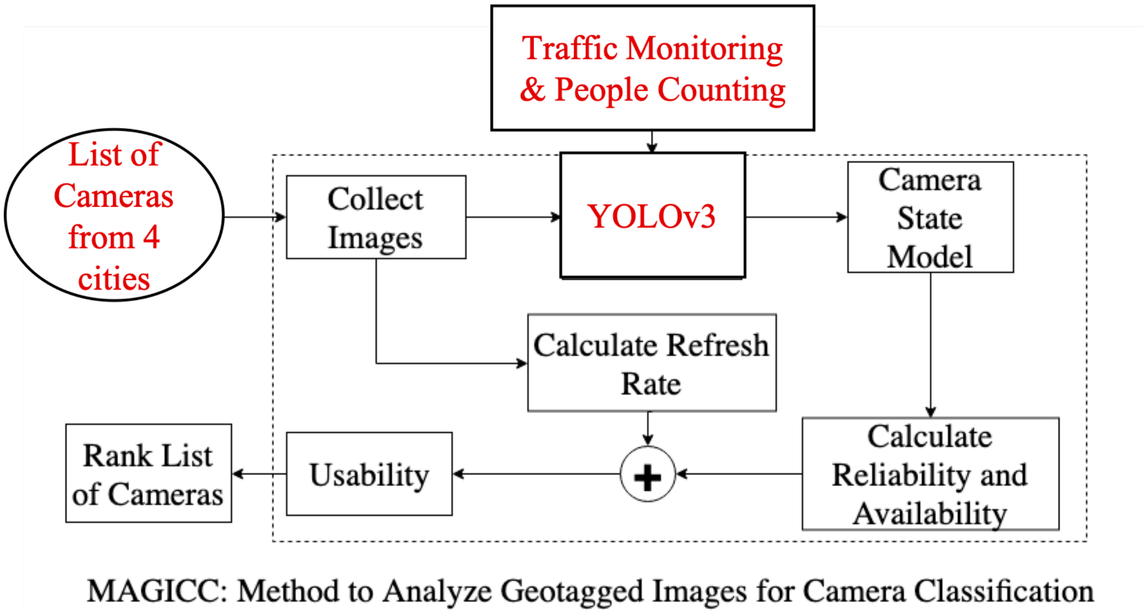


Figure 5.3. Experimental setup for evaluating MAGICC. We evaluate MAGICC based on traffic monitoring and people counting applications. YOLOv3 is used as an object detector for these applications.

MAGICC calculates usability score for each camera and outputs a ranked list of cameras for each application across four cities.

5.2.1 Traffic Monitoring

MAGICC identified usability scores for 2,024 out of the 2,500 sample cameras tested for traffic monitoring application. Remaining 476 cameras do provide information in more than 75% of the sampled images and thus are removed from usability score calculation and consider to have lowest usability. Out of the 2,024 cameras, Fig. 5.4(a) and (b) shows two examples that have high usability scores. These cameras are located on roads that have heavy traffic and will in turn detect many vehicles. Some of the cameras similar to the one shown in Fig. 5.4(c) are positioned such that they capture vehicles which are parked along the street. Since the system does not differentiate between a moving and a stationary vehicle, these cameras are ranked highly.

(a) usability = 0.98 (b) usability = 0.97 (c) usability = 0.99

Figure 5.4. Examples of the cameras with high usability score for traffic monitoring. (a) Camera is placed at one of the busiest highway (b) Camera is placed in a city (c) Camera is located at a busy street in the city

(a) usability = 0.20 (b) usability = 0.16 (c) usability = 0.19

Figure 5.5. Examples of the lower ranked cameras for traffic monitoring by MAGICC. (a) Camera is blocked by a pole (b) Flash from the vehicles blinds the camera at night (c) Camera is in an ideal location however, it does not update very often.

Many cameras that have much lower usability scores are often blocked or do not have direct views of streets. Fig. 5.5(a) shows a camera that is blocked by a pole. This camera will not be very good at identifying traffic and it has a low usability score, which is an expected behavior. Fig. 5.5(b) and (c) both show images that can be useful in identifying traffic patterns, however they have low usability scores. Fig. 5.5(b) shows a road at night when headlights from an oncoming car blind the camera. It is expected for this camera to have a low usability score as it is difficult to monitor traffic from this camera during night. Additionally, Fig. 5.5(c) shows an image with many cars. One would expect this camera to have a high usability score. Upon further analysis, this camera has a high refresh time of 598 seconds, hindering the camera's ability to send real-time traffic update. When sampling the website over multiple days, the same image was returned.

(a) frame_i (b) frame_{i+1} (c) frame_{i+2}

Figure 5.6. This camera has usability score of 0.13 in our system generated lowered ranked list for traffic monitoring. The camera shows lot of cars in frame_i but in the next two frames the number of cars goes to zero.

Fig. 5.6 shows another camera which has a low usability score but still has the potential to detect many vehicles. This camera gives burst of information regarding traffic. Although there are many vehicles detected in frame_i, the camera may not be good at determining traffic flow. The camera only shows a short portion of the road that is why it detects the vehicles only for a short duration of time. This makes this cameras has low availability and reliability.

Fig. 5.7 shows the distribution of cameras for reliability, availability, refresh time, and usability scores for traffic monitoring application. MAGICC uses Equation(4.6) to find reliability score for each camera. Fig. 5.7(a) shows more than 50% of the city1 cameras have reliability score close to 1, whereas city2, city3, and city4 appear to have a normal distribution of reliability scores.

Availability of a camera is calculated by MAGICC using Equation (4.9). Fig. 5.7(b) shows that cameras in city2, city3, and city4 have most of the cameras in low availability. City1 cameras on the other hand are distributed for the whole range.

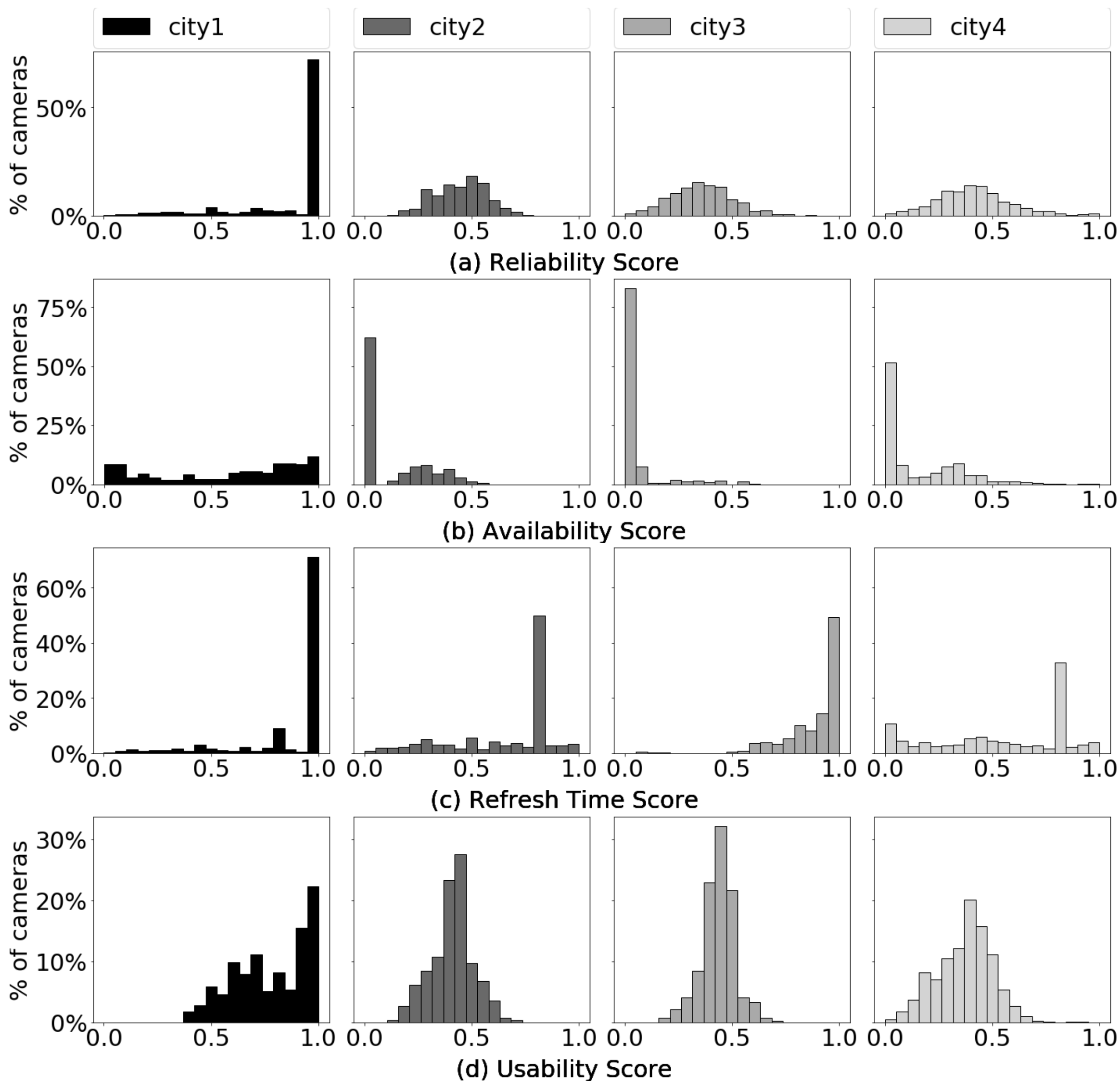


Figure 5.7. Distribution of reliability, availability, refresh time, and usability scores in 4 cities for traffic monitoring application. City1 cameras have high usability score because most of the cameras have high reliability, availability, and refresh time scores as compared to other cities.

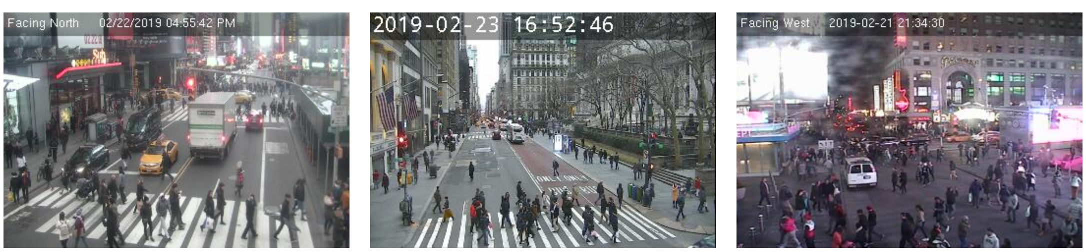
MAGICC calculates refresh time score of a camera by using Equation (??). Fig. 5.7(c) shows that most of the cameras have good refresh time scores.

A city may have many cameras that have low refresh times; that alone may not make them usable for an intended application, as we can see from the usability distribution shown in Fig. 5.7(d).

5.2.2 People Counting

The other application we tested our system on is detecting and counting people. Fig. 5.8 shows examples of three cameras with high usability for this application. These cameras focus on populated intersections or urban centers. Fig. 5.8(a) and (b) show people crossing a downtown intersection, while Fig. 5.8(c) shows people spread out in a tourist destination.

As discussed earlier, most of the cameras in city1 are located in areas that are both highly populated and traffic dense, shown in Fig. 5.1(a). Compared with cameras in the other three cities, cameras in city1 are able to detect more people. Fig. 5.9 (a) and (b) show the reliability and availability of the cameras in city1 for this application. Due to the difference in geographical location of the other cities, there is a less number of useful cameras.



(a) usability = 0.69 (b) usability = 0.86 (c) usability = 0.85

Figure 5.8. Examples of the cameras with high usability score for People counting application. (a) Camera is placed at one of the busiest intersection (b) Camera near intersection (c) Camera near a busy tourist spot

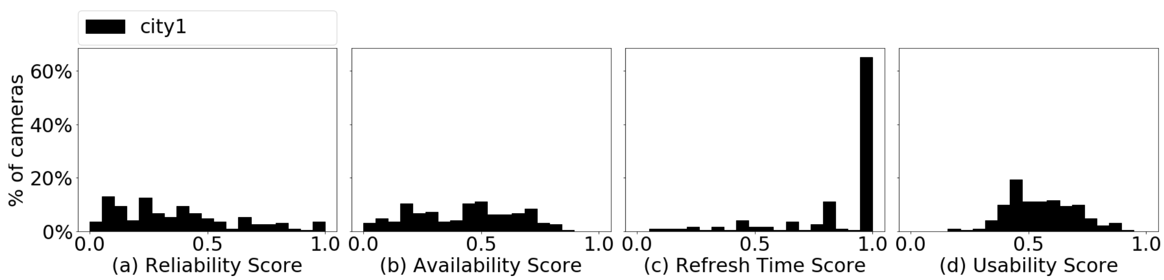


Figure 5.9. Distribution of various scores in city1 for people counting application. Cameras in cities 2-4 do not detect people; therefore, this figure does not show these cameras.

5.3 Day vs Night

The data from a camera changes over time. Therefore, its usefulness also varies with time. To find a relation between usefulness of a camera and time, we evaluate variation of usability of cameras within a day and across days. We observe that there is stark difference in the usability of the cameras during the day as compared to night. For all the four cities more than 80% of vehicles and people detected during the whole day are from the images from the day. Fig. 5.10 shows an example of one of a camera in a city3 camera network. This camera has a reliability score of 0.38, availability score as 0.04 and the refresh time score as 0.24 for detecting vehicles. Therefore it has a usability score of 0.22. However, Fig. 5.10(a) shows that this camera captures a lot of vehicles during the day and is quite useful. Therefore, we evaluate its performance between sunrise and sunset time of city3. It is found that during the daytime its reliability score increased to 0.90 and availability score shoots up to 0.84, and thereby the usability score becomes 0.74. Thus, making this camera most useful during the day.

(a) (b)

Figure 5.10. Example of a Camera with a usability score of 0.22 for the whole day, 0.74 for daytime traffic monitoring. Camera performed lower at night as there is less traffic on the road.

As observed during analysis of lower ranked cameras. Most common reasons for cameras become less usable during the night are (1) Less traffic at night(in general) (2) Cameras are setup to capture images during the daytime. These camera have low quality of images during the night. (3) Cameras are blinded by the headlights of the vehicles.

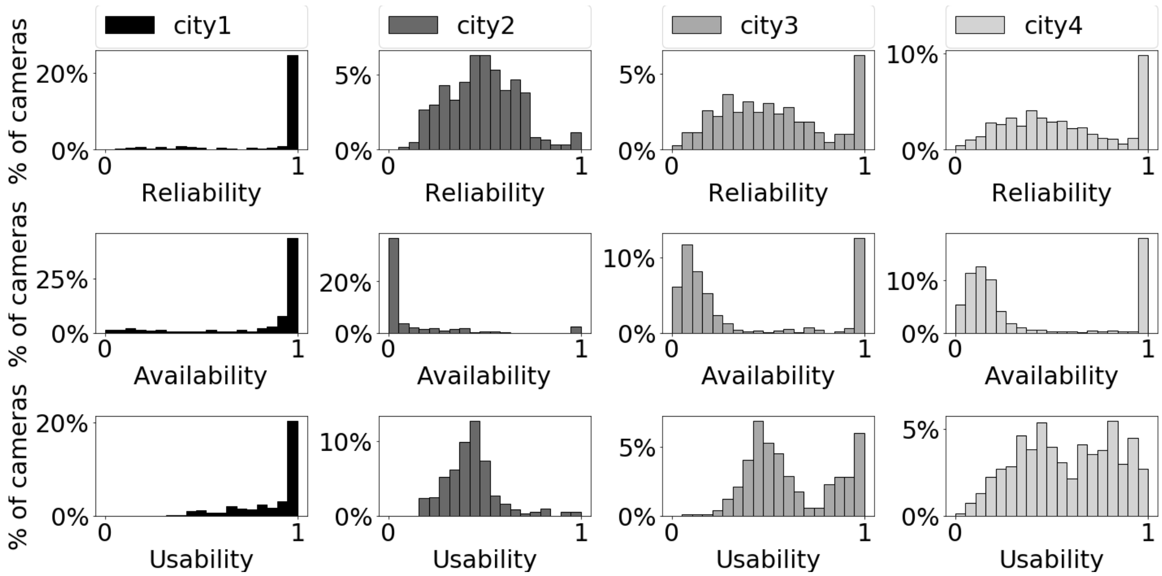


Figure 5.11. Reliability and availability of the cameras for the 4 cities during the day for traffic monitoring application. From Fig. 5.7 we observe that for city3 and city4 reliability and availability score distribution moves closer to value 1. Thereby improving the usability score of the cameras in city3 and city4

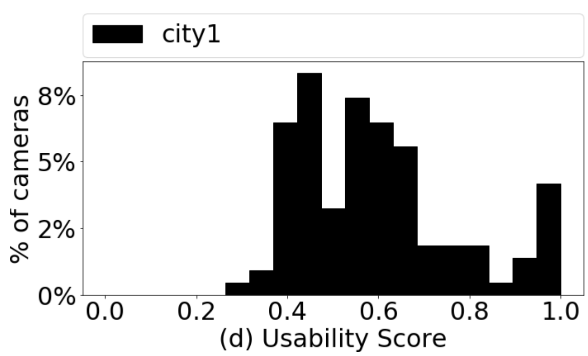


Figure 5.12. Usability score of the cameras for city1 during the day for people counting application. The cameras in city2, city3, and city4 are found to be not useful for people counting application.

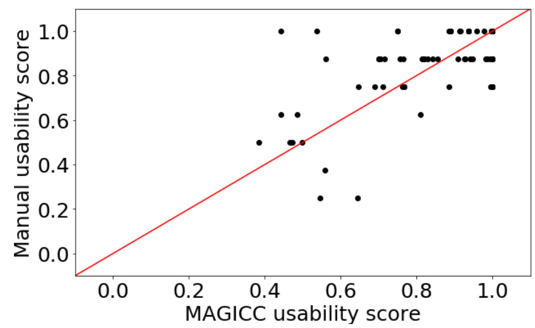
Fig. 5.11 and Fig. 5.12 shows the distribution of the reliability, availability, and usability scores during the daytime for the cameras in four cities for traffic monitoring and people counting application respectively. These figures compared with Fig. 5.7 and Fig. 5.9 shows that there is shift in the availability and reliability scores towards better values. This shift in scores also leads to shift in usability towards better values of the cameras in the four cities.

5.4 Model Validation

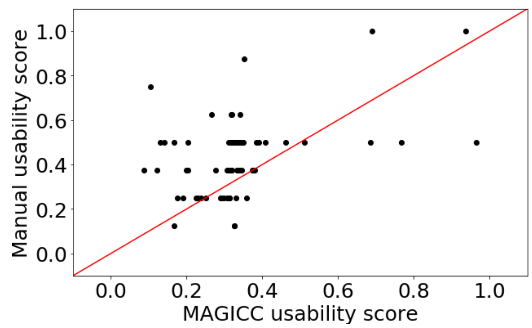
We validate the MAGICC by comparing the usability scores obtained from it with the manual scores for the cameras for traffic monitoring and people counting applications. For manual scoring, we ask a candidate to rate a camera between 0-9 for the usability of the camera for the two applications. A score of 0 means that the camera cannot be used for the intended application; whereas, a score of 9 means that the camera is very useful for the application.

Fig. 5.13 and Fig. 5.14 shows the comparison between manual usability scores and MAGICC usability scores for traffic monitoring and people counting application respectively. The plot is for a 20% sampled set of cameras for all the cities. The red line in the graph indicated the ideal case, i.e., manual score matches exactly the score given by MAGICC. The values above the red line indicates that manual score is higher than the MAGICC given score. And the values below the red line indicates that manual score is lower than MAGICC usability score. Fig. 5.13 shows that the manual scores comes very close to the MAGICC usability scores for the four cities. Some of the manual usability scores specifically for city3 are very low as some cameras were down during the manual scoring and the candidate gave it a very low score.

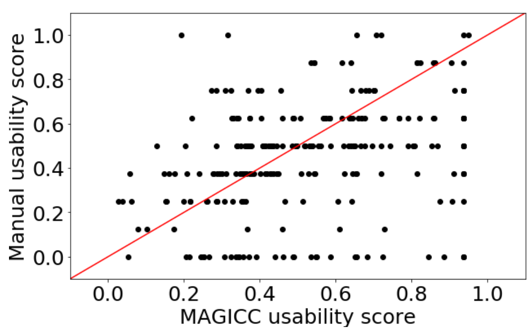
Similarly, manual scoring for people counting application in Fig. 5.14 shows that the data points comes very close to the ideal line. This shows that the manual scores for people counting application also comes very close to the MAGICC usability scores.



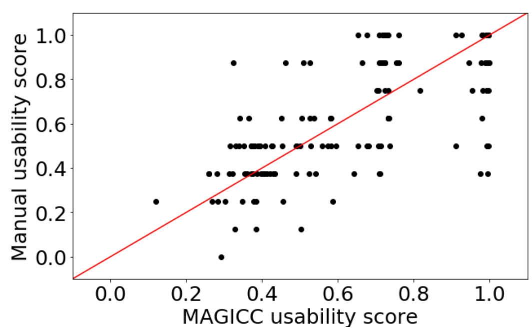
(a) city1



(b) city2



(c) city3



(d) city4

Figure 5.13. Manual scores comparison with the usability scores from MAGICC for the four cities for traffic monitoring application. The red line is the ideal line at which manual score equals the usability score by MAGICC. There were lot of cameras which were down in city3 during manual scoring.

The results discussed above show that the usability score from MAGICC resembles the user's expectation of the useful cameras. It that MAGICC is able to correctly identify the most useful camera for an intended application.

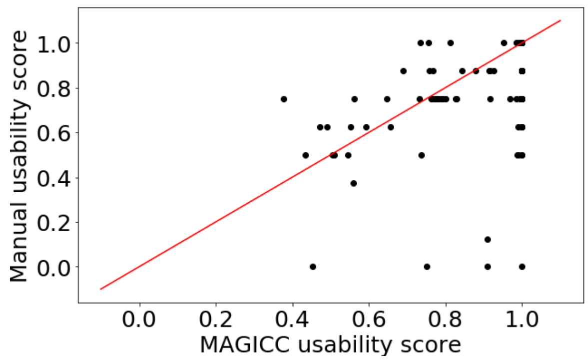


Figure 5.14. Manual scores comparison with the usability scores from MAGICC for the four cities for people counting application. The red line is the ideal line at which manual score exactly equals the usability score by MAGICC.

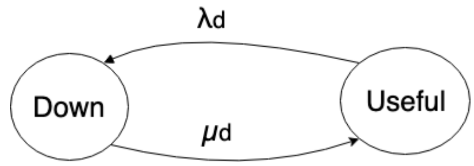
6. DISCUSSION

6.1 Various State Models

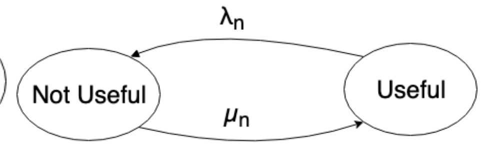
From the data that we collect from sampling the cameras, we find that not all the cameras in a network have full state transition model shown in Fig. 3.3. Fig. 6.1 shows various state models of a camera found in our analysis. Table 6.1 presents the distribution of the cameras in these state models across cities. A camera in a state model other than in Fig. 3.3 does not imply that the state model is incorrect. It indicates that in our data collection there are very few or no instances of some state transitions leading to a different state model for the camera. Thus, rates cannot be estimated for such states. This means that these transitions rarely happens and therefore have very low rates.

Table 6.1.
Distribution of Cameras in Various State Models. In our data samples, all the cameras do not always exist in the state model shown in Fig. 3.3. This is because some of the rates of state transitions are very low. These rates could not be determined from the samples in our dataset.

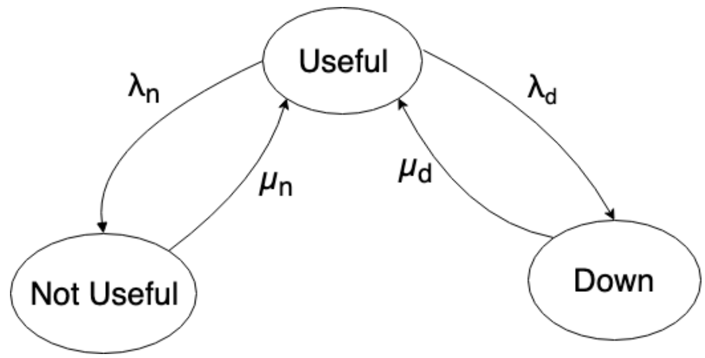
Camera States	# Cameras	
	Vehicle	Person
Only *Useful* State	9	0
State in Fig. 3.3	515	34
State in Fig. 6.1(a)	5	0
State in Fig. 6.1(b)	274	166
State in Fig. 6.1(c)	1,200	40
Only *Down* State	435	2,198



(a) No *Not Useful* State



(b) No *Down* State



(c) No transition between *Not Useful* and *Down* State

Figure 6.1. Various other State Models for camera found in our analysis of four camera networks in four cities.

6.2 Effect of Object Detector

For this paper, we do not evaluate the effect of using different object detector. We understand that a camera can be considered to be less useful because the object detector does not work efficiently for that camera. Fig. 6.2 shows an example of YOLOv3 for detecting vehicles for a camera in city1 network. Fig. 6.2 also shows that YOLOv3 does not give desired results for this camera as it misses detecting vehicles or detects vehicle with low accuracy. In this paper, we do not work on to improve the efficiency of YOLOv3. Improving YOLOv3 might help to change the ranks of some of the cameras but it will not displace the camera model.

6.3 City wise Evaluation

In this paper, we use cameras that are hosted on various websites. This set of cameras does not have all the cameras for the cities we use in our analysis. These

(a) Misses the red car in front (b) Wrongly predicts a car (c) Detects very few cars at night

Figure 6.2. Example of YOLOv3 missing or wrongly detecting car on one of the city1 Camera. This method requires computer vision for automatic analysis; improving computer vision is a topic beyond the scope of this paper.

are the cameras that are made available to public. It is not an exhaustive list of public cameras nor does it have cameras that have restricted access. Therefore we do not claim that one city cameras are better than other city. We find the degree of usefulness of a camera based on the given set of cameras and the application.

6.4 Applicability

There are varied application of the system shown in this paper. Some of the applications where it will be most useful are:

1. It will help authorities to better plan the location for new cameras and evaluate usefulness of already placed cameras.

2. It will help verify if a desired computer vision application will work for a given set of cameras.

3. The system will be helpful to an emergency response team. It will help to proactively find most useful cameras which emergency team can use for managing an emergency in future.

4. Tung et al. [31] discuss that current computer vision models struggle to provide consistent and accurate results on real time data from cameras. This study will

help identify cameras such as in Fig. 6.2 which can be used to create a real-time image database. This database can be used to train and improve neural network performance which will improve the efficiency of computer vision applications. In order for these networks to be useful, the data from these cameras must be analyzed.

7. CONCLUSIONS

We build an automated system to identify the usefulness of a camera for a given computer vision application. The system creates a state transition model for each camera and then calculates a usability score. We show the correctness of the camera model and the system by evaluating it on 2,500 cameras from the four cities for monitoring traffic and people counting applications. We evaluate the MAGICC's by comparing its usability score with the manual usability scores of the cameras. Evaluation results show that the usability of a camera changes with time and intended application. Evaluation results confirm that MAGICC is able to find useful cameras for an intended application. The results further show that most of the cameras in all the four cities are useful for traffic monitoring application. However, the cameras in city1 are also useful for people counting application.

REFERENCES

REFERENCES

[1] "Houston transtar," https://traffic.houstontranstar.org/cctv/transtar/, accessed: 2019-04-05.

[2] M. Satyanarayanan, "Mobile computing: the next decade," *Mobile Computing and Communications Review*, vol. 15, pp. 2–10, 01 2011.

[3] A. Mohan, K. Gauen, Y. Lu, W. W. Li, and X. Chen, "Internet of video things in 2030: A world with many cameras," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[4] M. A. Mahmood, W. K. Seah, and I. Welch, "Reliability in wireless sensor networks: A survey and challenges ahead," *Computer Networks*, vol. 79, pp. 166 – 187, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614004708

[5] I. Silva, L. A. Guedes, P. Portugal, and F. Vasques, "Article reliability and availability evaluation of wireless sensor networks for industrial applications," 2012.

[6] T. Jesus, P. Portugal, F. Vasques, and D. Costa, "Automated methodology for dependability evaluation of wireless visual sensor networks," *Sensors*, vol. 18, p. 2629, 08 2018.

[7] D. Bruneo, A. Puliafito, and M. Scarpa, "Dependability evaluation of wireless sensor networks: Redundancy and topological aspects," in *SENSORS, 2010 IEEE*, Nov 2010, pp. 1827–1831.

[8] D. G. Costa, I. Silva, L. A. Guedes, P. Portugal, and F. Vasques, "Availability assessment of wireless visual sensor networks for target coverage," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Sep. 2014, pp. 1–8.

[9] A. Munir, J. Antoon, and A. Gordon-Ross, "Modeling and analysis of fault detection and fault tolerance in wireless sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 1, pp. 3:1–3:43, Jan. 2015. [Online]. Available: http://doi.acm.org.ezproxy.lib.purdue.edu/10.1145/2680538

[10] R. Lajara, J. J. Pérez-Solano, and J. Pelegrí-Sebastiá, "Predicting the batteries' state of health in wireless sensor networks applications," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 11, pp. 8936–8945, Nov 2018.

[11] C.-J. Liang, K. Chen, N. Bodhi Priyantha, J. Liu, and F. Zhao, "Rushnet: Practical traffic prioritization for saturated wireless sensor networks," *SenSys 2014 - Proceedings of the 12th ACM Conference on Embedded Networked Sensor Systems*, pp. 105–118, 11 2014.

[12] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu, "Zisense: Towards interference resilient duty cycling in wireless sensor networks," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '14. New York, NY, USA: ACM, 2014, pp. 119–133. [Online]. Available: http://doi.acm.org/10.1145/2668332.2668334

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788.

[14] D. Costa, I. Silva, L. Guedes, F. Vasques, and P. Portugal, "Availability issues in wireless visual sensor networks," *Sensors*, vol. 14, no. 2, p. 2795–2821, Feb 2014. [Online]. Available: http://dx.doi.org/10.3390/s140202795

[15] T. Chakraborty, A. U. Nambi, R. Chandra, R. Sharma, M. Swaminathan, Z. Kapetanovic, and J. Appavoo, "Fall-curve: A novel primitive for iot fault detection and isolation," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '18. New York, NY, USA: ACM, 2018, pp. 95–107. [Online]. Available: http://doi.acm.org/10.1145/3274783.3274853

[16] W. Elghazel, J. Bahi, C. Guyeux, M. Hakem, K. Medjaher, and N. Zerhouni, "Dependability of wireless sensor networks for industrial prognostics and health management," *Computers in Industry*, vol. 68, pp. 1 – 15, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166361514001821

[17] D. Bruneo, S. Distefano, F. Longo, A. Puliafito, and M. Scarpa, "Evaluating wireless sensor node longevity through markovian techniques," *Computer Networks*, vol. 56, no. 2, pp. 521 – 532, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611003690

[18] A. Chatterjee and K. Das, *State Estimation and Anomaly Detection in Wireless Sensor Networks*, 06 2018, pp. 317–334.

[19] J. C. SanMiguel and A. Cavallaro, "Energy consumption models for smart camera networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2661–2674, Dec 2017.

[20] Y. Koh, A. Mohan, G. Wang, H. Xu, A. Malik, Y. Lu, and D. S. Ebert, "Improve safety using public network cameras," in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, May 2016, pp. 1–5.

[21] T. Semertzidis, K. Dimitropoulos, A. Koutsia, and N. Grammalidis, "Video sensor network for real-time traffic monitoring and surveillance," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 103–112, June 2010.

[22] E. Cayirci and T. Coplu, "Sendrom: Sensor networks for disaster relief operations management," *Wireless Networks*, vol. 13, no. 3, pp. 409–423, Jun 2007.

[23] B. Song, C. Ding, A. T. Kamal, J. A. Farrell, and A. K. Roy-chowdhury, "Distributed camera networks," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 20–31, May 2011.

[24] U. M. Erdem and S. Sclaroff, "Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 156 – 169, 2006, special issue on Omnidirectional Vision and Camera Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314206000671

[25] A. van den Hengel, R. Hill, B. Ward, A. Cichowski, H. Detmold, C. Madden, A. Dick, and J. Bastian, "Automatic camera placement for large scale surveillance networks," in *2009 Workshop on Applications of Computer Vision (WACV)*, Dec 2009, pp. 1–6.

[26] J. Kritter, M. Brévilliers, J. Lepagnot, and L. Idoumghar, "On the optimal placement of cameras for surveillance and the underlying set cover problem," *Applied Soft Computing*, vol. 74, pp. 133 – 153, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494618305829

[27] *Kolmogorov–Smirnov Test*. New York, NY: Springer New York, 2008, pp. 283–287. [Online]. Available: https://doi.org/10.1007/978-0-387-32833-1_214

[28] M. Rausand and A. Høyland, *System Reliability Theory: Models, Statistical Methods, and Applications*, ser. Wiley Series in Probability an.

[29] J. W. J. W. Tukey, *Exploratory data analysis*, ser. Addison-Wesley series in behavioral science. Reading, Mass.: Addison-Wesley Pub. Co., 1977.

[30] J. Huang, C. Lin, X. Kong, B. Wei, and X. Shen, "Modeling and analysis of dependability attributes for services computing systems," *IEEE Transactions on Services Computing*, vol. 7, no. 4, pp. 599–613, Oct 2014.

[31] C. Tung, M. R. Kelleher, R. J. Schlueter, B. Xu, Y. Lu, G. K. Thiruvathukal, Y. Chen, and Y. Lu, "Large-scale object detection of images from network cameras in variable ambient lighting conditions," *CoRR*, vol. abs/1812.11901, 2018. [Online]. Available: http://arxiv.org/abs/1812.11901

APPENDIX

Table 1.
$R2$ values for exponential curve fit on state probabilities

Prob	1	2	3	4	5	6	7
Useful	0.76	0.33	0.42	0.82	0.81	0.71	0.87
Not Useful	0.00	0.27	0.31	0.80	0.78	0.63	0.82
Down	0.92	0.63	0.45	0.72	0.75	0.77	0.84

Table 2.

p-Values of the Kolmogorov-Smirnov test for various testing distributions on state transition from *Useful* to *Not Useful* states. Exponential, Bradford, Uniform, Weibull$_{min}$ pass the p-value tests.

cam #	Lognorm	**Exp**	Exp pow	Norm	Gamma	Rayleigh	Alpha	Beta	**Bradford**	**Uniform**	Dweibull	**Weibull$_{min}$**
1	0.22	**0.21**	0.82	0.00	0.11	0.03	0.04	0.94	**0.91**	**0.94**	0.00	**1.00**
2	0.64	**0.46**	0.29	0.23	0.00	0.53	0.04	0.09	**0.87**	**0.87**	0.12	**0.10**
3	1.00	**0.21**	0.88	0.03	0.03	0.14	0.00	0.91	**0.87**	**0.89**	0.00	**1.00**
4	0.00	**0.73**	1.00	0.00	1.00	0.06	0.05	0.00	**0.90**	**0.90**	0.00	**0.56**
5	0.33	**0.42**	0.06	0.06	0.09	0.22	0.25	0.20	**0.86**	**0.86**	0.03	**0.73**
6	1.00	**0.17**	0.00	0.00	0.12	0.01	0.00	0.11	**0.94**	**0.94**	0.00	**0.13**
7	0.98	**0.63**	1.00	0.00	0.00	0.05	0.00	0.95	**0.93**	**0.93**	0.00	**1.00**
9	0.65	**0.66**	0.77	0.41	0.17	0.79	0.57	0.33	**0.84**	**0.84**	0.32	**0.00**
10	0.00	**0.67**	0.94	0.01	0.00	0.13	0.03	1.00	**0.94**	**0.94**	0.01	**1.00**

Table 3.
Error in best curve fit from the distributions on the data sample

distribution	1	2	3	4	5	6	7	8
Expo	**514.18**	201.35	**245.15**	**453.06**	253.14	434.09	**306.75**	**223.55**
Uniform	1345.41	359.71	447.26	1583.51	315.95	1023.60	864.49	454.82
Bradford	727.24	**190.44**	328.65	867.14	**251.17**	906.21	707.14	346.60
Weibull_{min}	3190.01	356.41	26680.66	469.00	1569.55	**314.08**	20815.29	348.21