IoT CCTV with MobileNet SSD

Kyeong Min Park and Gyu Sang Choi
Department of Information & Communication Engineering
Yeungnam University
Gyeongsan, Republic of Korea
lapis94@gmail.com, castchoi@ynu.ac.kr

Abstract— **It is very labor-intensive to review the CCTV footages searching for a specific event and/or entity, because stored data do not have any additional information other than series of concatenated images. In this paper, we automate this procedure using a pre-trained MobileNet Single Shot Detection (SSD) Convolutional Neural Network (CNN) and identify objects in streaming videos. We have developed the prototype of our proposed system with a Raspberry PI and a compatible camera. For simplicity, we focus on two kinds of objects only, i.e. humans and cars, in the prototype. Our system works well with streaming videos of 1 frame per second in real-time.**

I. INTRODUCTION

Recently, CCTVs have been installed in various places to prevent crimes and accidents. The CCTV installed in this way becomes an important data for identifying the causes and processes of various crimes and accidents. However, most CCTVs have a problem in that the shooting angle is fixed at the time of installation so that only one angle can be taken. In order to find CCTV records in case of an accident, a person looks directly at the video and uses the method of finding and extracting the corresponding video. This method is very labor intensive.

To solve this problem, 'IoT CCTV' system is developed so that the user can change the shooting angle by moving the stick in the desired direction. By connecting to the server via Wi-Fi, it is easy to install and manage compared to the existing system and allows the user to watch the video recorded in real time. It identifies the specific object by the image only without the use of additional sensors and achieves quick results when necessary.

In addition, the purpose of this study is to build a low-cost system using only Raspberry Pi, not a high-performance server.

II. RELATED WORKS

A. Related Work 1

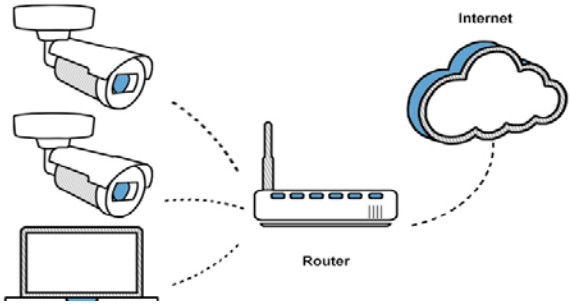


Figure 1. IP CCTV diagram

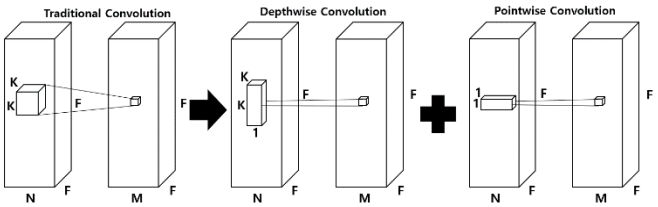
In case of the existing CCTV system, the camera and recorder are connected by cable and communicated with the server to store them. In case of additional installation of the camera, the cost of purchasing a new camera and additionally installing the cable connecting the recorder and the camera to the wall or ceiling There is a cost. In the system to be implemented in this task, the motor is attached to the camera to increase the shooting angle compared to the existing camera, which increases the range of one camera and can communicate with the server via Wi-Fi. Unlike the wireless AP environment, there is no additional installation cost.

B. Related Work 2

MoblieNet is a type of CNN architecture. VGGNet and ResNet aimed at improving performance have a large performance improvement range but have a large network size. In AlexNet, we used eight layers, but in ResNet we have 152 layers. To solve this problem, MobileNet has been proposed to achieve more than a certain level of performance with a small amount of computation. MobileNet simplifies and uses the convolution filter compared to the existing CNN.

Traditional Convolution filters have the computational cost of :

$$F^2 * N * M * K^2 \quad (1)$$



F^2 means input map size $F * F$. N means input channel.

Figure 2. Separable Convolution

K^2 means kernel size $K * K$. M means output channel. MobileNet will split this into two filters.

The depthwise filter has a channel of 1 with a calculation. Depthwise filter has the computational cost of :

$$F^2 * M * K^2 \quad (2)$$

The pointwise filters have a kernel size of 1 * 1 with a computation. Pointwise filter has the Computational cost of:

$$N * M * K^2 \quad (3)$$

Comparing the computational cost of the sum of the depthwise and pointwise filters with that of the traditional filter is:

$$\frac{F^2 * M * K^2 + N * M * F^2}{F^2 * N * M * K^2} = \frac{1}{N} + \frac{1}{K^2} \quad (4)$$

Its calculation is reduced to the usual 1/8 to 1/9 (when K = 3, M> = 32).

When this operation is taken, $1 * 1$ convolution is mainly used, which can be calculated using a general matrix multiply function. In addition to this method, a channel reduction method that reduces the size of the channel is used. This reduces the channel by multiplying it by a factor less than one. Using this approach, the computational cost can be greatly reduced compared to the conventional CNN model.

III. Main Idea of This Work

Two Raspberry Pis, which consist of a server and a client, implement two-way communication using Wi-Fi, and a client that controls the motor and transmits the captured image to the server to capture the image in the desired direction from the server. The server implements an operation of detecting an object using the CaffeModel trained with OpenCV and COCO Dataset.

A. Network

Implement bidirectional full-duplex socket communication using Wi-Fi and Thread. When the joystick is operated on the client, the x, y and axis coordinates are received and the input is sent to the server, and the server moves the motor according to the input coordinates.

B. Embedded I/O

When the user moves the joystick to operate the camera in the desired direction, the user receives inputs for the coordinates corresponding to the x-axis and y-axis, respectively. Output accordingly.

In order to receive the input, the performance goal is to set the joystick delay time to 1 second or less and the motor operation delay time to 1 second or less.

C. Object Detection

Object recognition is implemented using MobileNet implemented with OpenCV and Caffe. The network used a trained COCO dataset. The image captured by the client is processed into OpenCV function and put into the network by using a certain size. Objects detected in the network are boxed and tagged in the output image and output with the image. The network trained with COCO dataset can detect people, cars, bicycles, bottles, sofas, buses, etc., but in this implementation, only people and cars are detected.

IV. Result

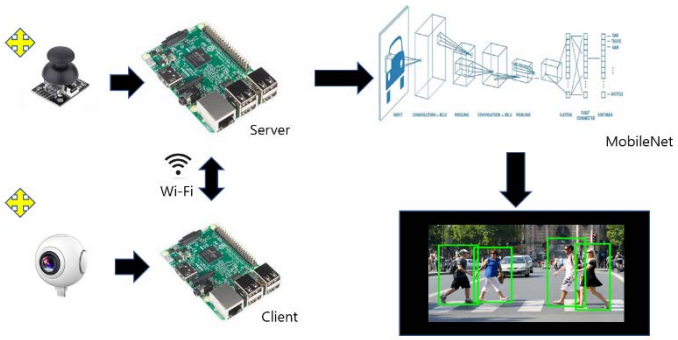


Figure 3. Diagram

The server and the client communicate over Wi-Fi. The implementation languages are Python, both server and client, and the libraries used are ZMQ, OpenCV, imutils, and threading. The ZMQ library was used to send OpenCV images to the server, and OpenCV was used for imaging and CNN processing. The server inputs the image received from the client into neural network by using OpenCV's DNN module in frame unit. The size of the input image was defined as 300 * 300 and the scalefactor value was defined as 0.007843. The network size is 300 * 300 and the mean subtraction is 127.5.

The output of this implementation is:

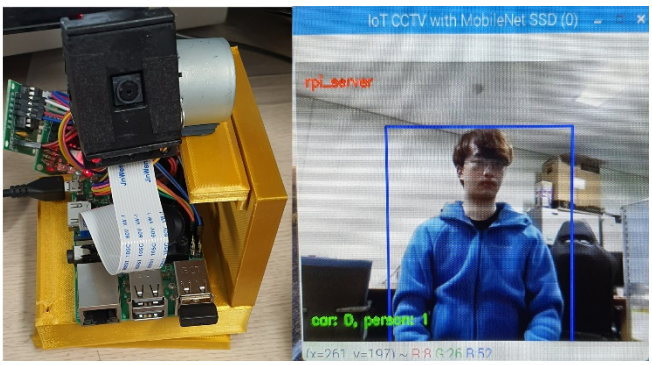


Figure 4. Client & Server Figure

V. CONCLUSION

In this study, a system was implemented to capture and transmit video using only raspberry pies without using high-performance servers and to determine objects in images. This system is stand-alone with no external network present. The installation of the system is also simple and inexpensive. However, the limitations of mobileAP's processing power have not shown perfect performance. It is thought that this can be resolved by software upgrades as well as by hardware upgrades. Currently, the image received is processed by frame, but the image data is stored continuously, and the overall performance improvement can be achieved by entering only the frame of a certain period into the object detector. In addition, I think it can improve performance by further reducing the size of the model or improving parallel processing.

ACKNOWLEDGMENT

This research was supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program. No.10063130 and MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2016-0-00313) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

REFERENCES

[1] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto and Hartwig Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, Google Inc.,2017

[2] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, “ImageNet classification with deep convolutional neural networks”, December 2012, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 pp. 1097-1105

[3] Rosebrock, A. (2019). Raspberry Pi: Deep learning object detection with OpenCV - PyImageSearch. [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/ [Accessed 15 Oct. 2019].

[4] Rosebrock, A. (2019). Raspberry Pi: Deep learning object detection with OpenCV - PyImageSearch. [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/ [Accessed 15 Oct. 2019].

[5] A. Rosebrock, "Raspberry Pi: Deep learning object detection with OpenCV - PyImageSearch", PyImageSearch, 2019. [Online]. Available: https://www.pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/. [Accessed: 15- Oct- 2019].