International Journal of Hybrid Information Technology
Vol. 9, No.8 (2016), pp. 199-210
http://dx.doi.org/10.14257/ijhit.2016.9.8.18

An Implementation of Intelligent Surveillance Bot System based on Robust CNN Algorithm

Ming-Shou An and Dae-Seong Kang,

Dong-A University, Dept. of Electronics Engineering, 37 Nakdong-daero 550 beon-gil Saha-gu, Busan, Korea
dskang@dau.ac.kr

Abstract

As accidents and crimes have increased at an alarming rate lately, most people are faced with the necessity of security and surveillance in private places as well as in public places. Therefore, the demand for intelligent surveillance systems has been increasing, and various technologies for image recognition and analysis from cameras are being developed for use in automatic monitoring. Among these techniques, the most popular and advanced method is deep learning. It is a field of machine learning based on neural networks. In this paper, we propose the object classifier technique with the Convolutional Neural Networks (CNN) algorithm, which is most widely used in image processing. Additionally, we implement the image analysis bot system based on the above proposed algorithm.

Keywords*: Intelligent Surveillance System, Machine Learning, Neural Network, Deep Learning, Convolutional Neural Network*

1. Introduction

An intelligent video surveillance system can automatically recognize and classify the target [1], track a moving object in the video sequence [2], and mark the target to draw the trajectory [3]. It can also simultaneously analyze multiple targets in the same scene [4] and can have flexible settings based on the features of the prevention objects [5]. A good surveillance system can adapt to different environment changes, such as light, day and night changes, and weather changes, and can adapt to camera jittering [6]. An intelligent surveillance system is developed and produced by international advanced intelligent video analysis technology.

For the main application of intelligent surveillance systems, there are two major development directions. One is to license plate recognition and face recognition as the core representative intelligent recognition technology to be mainly used in fields such as policing, airport security, and customs. Other uses include perimeter guarding, statistics on the number of people, automatic tracking, wrong-way travel detection, and illegal parking detection as representative of behavior analysis technology. These techniques are called computer vision (CV) [7], artificial intelligence (AI) [8], machine learning (ML) [9,10], and deep learning (DL) [11,12]. For building an intelligent video surveillance system, there are some algorithms that need to be applied from the fields of the above techniques. In general, object detection, tracking, classification, and recognition algorithms are the most widely used applications [13]. For these, a large number of methods have been proposed and researched by many researchers.

In this paper, to build the efficient intelligent surveillance system, the CNN algorithm, one of the deep learning algorithms, was used for the image analysis bot (bot, the abbreviation of robot, is a software tool for processing data) system to detect and classify the object (person and vehicle) in a video sequence.

ISSN: 1738-9968 IJHIT

2. Convolutional Neural Networks

The CNN [14,15] is generally comprised of several layers of neural networks, and the layers are divided into three different basic layers as follows:

-Convolution layer: Extracts the convolution feature

-Pooling layer: Because of the structural characteristics of the image, there are many pixels to be analyzed. To reduce the number of features, the pooling layers conduct a subsampling process that computes the max, min, or average value of a particular feature over a region of the image.

-Feedforward layer: Finally, after several convolutional and pooling layers, the high-level reasoning in the neural network is conducted via fully connected layers. A fully connected layer takes all the neurons in the previous layer (be they fully connected, pooling, or convolutional) and connects them to every single neuron it has. For this, forward propagation and backward propagation algorithms can be used.

For all layers, forward propagation is applied for estimation, and backward propagation is used to learn the weights of neurons. The learning process is similar to that of other deep learning algorithms.

2.1. Affiliations

In our method, three convolutional layers were applied for feature extraction. Through the convolutional layers, the convolutional operation can make the original signal (image) enhance the features and reduce the noise. Layer C_1 in Figure 1, one of the convolutional layers, consists of four feature maps. Each unit (neuron) in each feature map is connected to the 5×5 neighborhood of the input image. For the 32×32-sized input image, the size of the feature map is 28×28 pixels, which can prevent the input connections from falling off the boundary. Each feature map unit should compute $(5 \times 5 = 25)$ trainable parameters and a trainable bias parameter. For four feature maps in layer C_1, there are $(5 \times 5 + 1) \times 4 = 104$ trainable parameters. These feature maps were subsampled as 14×14 pixels by the S2 layer. We will describe the subsampling process in the next section (3 Subsampling Layers).

Layer C_3 in Figure 1, also one of the convolutional layers, consists of 14 feature maps. The 14 5×5 convolution kernels convolute the output of Layer S_2, followed by the addition of trainable biases. Then, each feature map has only 10×10 neurons (size of output image of feature maps through layer C_3). Here, the inputs of the previous eight feature maps in C_3 were fused with two neighborhood feature maps in S_2. Therefore, layer C_3 has 364 trainable parameters.

Layer C_5, the third convolutional layer, has 56 feature maps. Each unit in layer C_5 is connected to 14 5×5 neighborhoods of the layer S_4. Since the size of the feature maps of layer S_4 is 5×5 (same as the kernel size) through the subsampling processing, the size of the feature maps of layer C_5 is 1×1. It constitutes the full connection between layer C_5 and S_4. The reason why C_5 is still labeled as a convolution layer rather than the full associative layer is because if the input becomes larger than 32×32, while the other remains unchanged, then the size of the feature maps will be greater than 1×1. Layer C_5 has 112 trainable parameters.

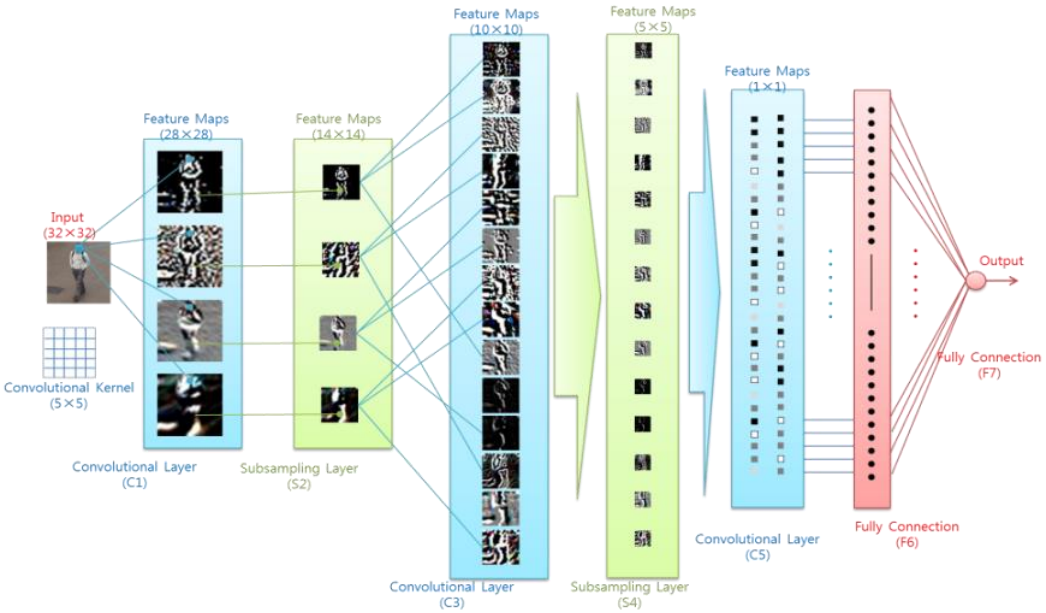


Figure 1. Structure of Proposed CNN Algorithm

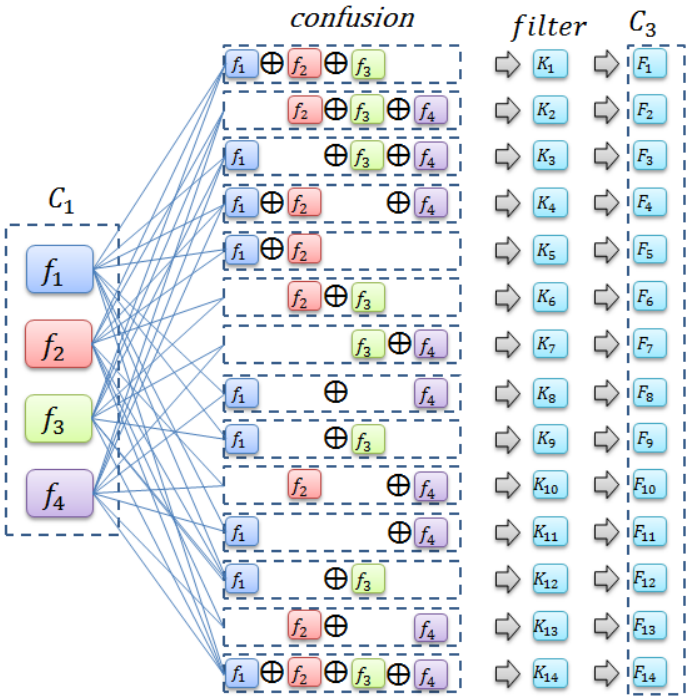


Figure 2. Selection Rule of Feature Map Connection between C_1 and C_3

In Figure 2, f_1, f_2, f_3, f_4 are the feature maps obtained in layer and K_1,..., K_{14} are the convolutional kernels. F_1,..., F_{14} are the feature maps obtained from the output of C_3.

In Figure 4, f_1, f_2, f_3,..., F_{14} are the feature maps obtained in layer C_3. K_1,..., K_{14} are the convolutional kernels. F_1,..., F_{14} are the feature maps obtained from the output of C_5. This illustrates the feature map confusion rule between C_3 and C_5.

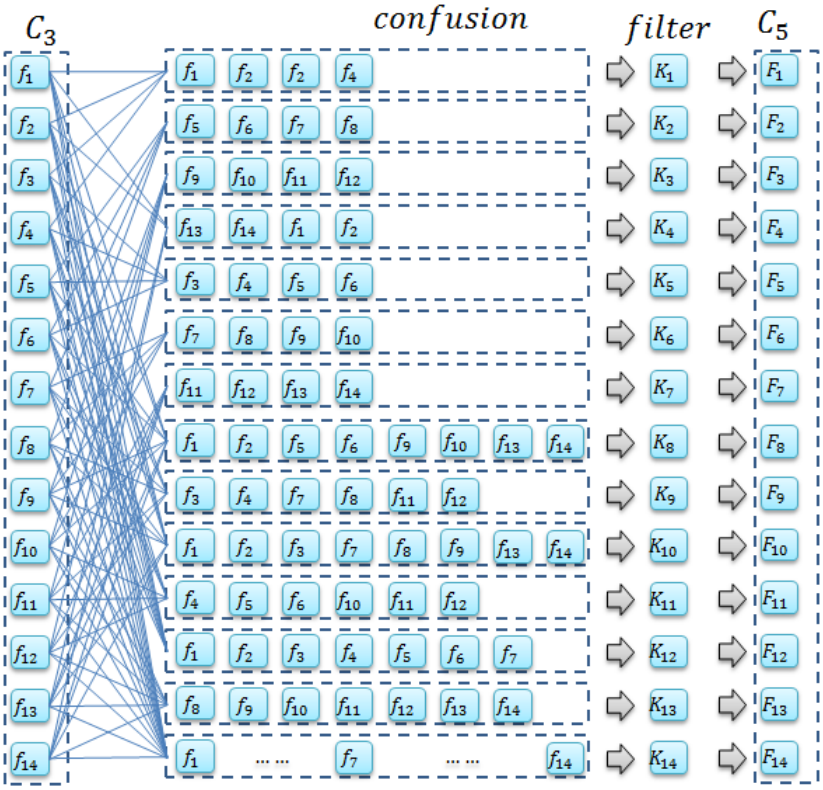


Figure 3. Selection Rule of Feature Map Connection between C_3 and C_5

2.2. Subsampling Layers

The first subsampling layer is layer S_2, which is composed of four 14×14 feature maps, one for each feature map in C_1. Each unit of these feature maps is connected to a 2×2 neighborhood of the corresponding feature maps in C_1. For subsampling, we applied the Low-Pass, High-Pass Segmentation and Reconstruction approach. The image can be seen as a matrix, for which is generally assumed that the size of the image matrix $X \times Y$ and X, Y are approximately non-negative integers. The frequency domain decomposition for the image is as follows:

$$A_{i-1}f(X,Y) = A_i f(X,Y) + D_i^1 f(X,Y) + D_i^2 f(X,Y) + D_i^3 f(X,Y), \quad (1)$$

where A is low-frequency components, D_i^1 can be seen as the horizontal component, D_i^2 is the vertical component, and D_i^3 is the diagonal component of the high-frequency component D.

The unit of the 2×2 area corresponding feature map in the previous layer C_1 is the receptive field. Then, this area added a trainable bias through the low-pass decomposition filter to reconstruct a subsampled pixel with a trainable parameter, and the results were calculated by the sigmoid function. The size of each feature map in layer S_2 is a quarter (1/2 row and 1/2 column) of that of the feature maps in C_1. It has eight trainable parameters.

Layer S_4 is also a subsampling layer with 14 5×5 feature maps. The receptive field of each 2×2 area unit corresponds to the feature map in the previous layer C_2, like that of S_1 and C_1. Therefore, layer S_2 has 28 trainable parameters.

2.3. Full Connection Layers

The full connection layer F_6 and F_7 contain the neural unit, the same as a classic network. These layers generate a classifier. In layer F_6, 14 neurons are fully connected to all the units with only one corresponding feature map in layer C_5. The output neuron in F_7 is fully connected to all neurons in layer F_6. This weighted sum is then passed through a hyperbolic tangent function to produce the state of the unit, in between -1.0 and 1.0. The output of the neuron is used to classify the input image as a nonperson (or vehicle) if its value is negative or as a person if its value is positive.

To learn the parameters of the proposed network, the weight update algorithm used the backpropagation algorithm. In the pooling layer (subsampling layer), the low-pass band value from a batch of size p by q was found. The equation is expressed as follows:

$$H_{l+1}(X,Y) = Af_{a-p\le a\le a+p, b-q\le b\le b+q}(H_l(X+a, Y+b)) \quad (2)$$

where (X, Y) is the coordinate of the pooling layer feature maps and H_i is the hidden variables in the lth layer. For this equation, we should only compute $\partial H_{l+1}/H_l$ because it has no parameters. If the value of the pixels is the value that is used for a low-pass band reconstruction, the value is passed; if not, a value of 0 is assigned.

In convolutional layers, the coordinate (X, Y) of the ith convolution filter in the $\mathrm{l}+1$th layer can be expressed as follows:

$$H_{l+1}(i,X,Y) = \sum_{i=1}^{m}\sum_{a=1}^{p}\sum_{b=1}^{q} H_l(j, X+a, Y+b) * \omega(i,n;a,b) \quad (3)$$

where the number of kernels of the lower convolutional layer is m and the number of kernels of the upper convolutional layer is n. If you calculate the differential coefficient, the gradient of the parameters can be expressed as follows:

$$\frac{E}{\partial\omega} = \sum_{X}\sum_{Y}\frac{E}{H_{l+1}}(X,Y)H_l(X,Y) \quad (4)$$

This means the gradient of the parameters can be calculated from the pixel value in the previous layer multiplied by the gradient value and aggregated. $\mathrm{E}/\partial H_l$ obtains the results of the sum of the convolutional gradient in the previous layer by weight ω. Table 1 shows the training algorithm of the CNN.

Table 1. Algorithm: CNN

Initialize random weights

$E \leftarrow \infty$

while $E > threshold$ do

for $t \in$ training Set do

prediction ← forward propagation (t)

actual ← t. label

$$E \leftarrow \frac{1}{2}\sum_{k \in K}(O_k - t_k)^2$$

for each layer do

for each kernel

calculate convolution filtering

calculate subsampling $(H_{l+1}(i, I, O))$

end for

compute δ_l at each layer

$$\omega_l \leftarrow \omega_l \leftarrow \eta \delta_l O_{l-1}$$

$$\theta_l \leftarrow \theta_l \leftarrow \eta \delta_l$$

end for

end for

end while

3. Experimental Results

3.1. Pedestrian Detection

For pedestrians, we used the dataset of INRIA [16] (which has 1,218 negative images and 614 positive images) and CVC02 [17] (which has 7,650 negative images and 1,016 positive images), as shown in Table 2, to train our network.

Table 2. Comparison Results of the Detection Accuracy For Each Algorithm(%)

Dataset	Negative images	Positive images
INRIA	1218	614
CVC02	7,650	1,016

The average pedestrian detection time for one frame with 640×480 resolution is around 67ms using our computer. Table 2 shows the comparison results of the detection accuracy between the proposed detection algorithm using the CNN and other algorithms. To evaluate the accuracy of the detection algorithms, we defined the accuracy equation as

$$\gamma_{accuracy} = \frac{n_d}{n_A} \times 100\% \quad (5)$$

where n_d is the number of detected objects (pedestrians or vehicles) and n_A is the number of all objects in the dataset. In the experiment, we compare the results of the algorithms working in three video sequences. For the first video sequence, "PETS 2006" [18], HOG+SVM shows a higher quality performance than the others. In addition, the performance of the CNN is better than those of the other algorithms in the other video sequences, "PETS 2009" [19] and " TownCenter" [20].

Table 3 shows the results of the processing time between the CNN and others. In the case of the PETS 2006 dataset, it takes 0.061 seconds to process one frame. The frame rate of PETS 2006 is 24 (frames/sec) (i.e., the system processes 23 frames per second).

Table 3. Comparison Results of the Detection Accuracy for each Algorithm (%)

Dataset	ANN	Haar+ Adaboost	HOG+ SVM	CNN
PETS 2006	96.23	97.50	98.00	97.05
PETS 2009	92.38	92.60	92.00	94.22
TownCenter	87.25	87.55	90.50	92.65
Average	93.56	93.55	94.35	95.35

Table 4. Comparison Results of the Processing Time for Each Algorithm (sec/frame)

Dataset	ANN	Haar+ Adaboost	HOG+ SVM	CNN
PETS 2006	0.062	0.088	0.064	0.051
PETS 2009	0.059	0.105	0.055	0.049
TownCenter	0.061	0.084	0.057	0.050
Average	0.061	0.100	0.057	0.050

As the processing time is $0.061 \times 23 = 1.403$ (sec), the system has not calculated 23 frames per second. Through the experimental results, the tracking system using the CNN plays the video slowly for this reason. In contrast, as the proposed method reduces the processing time by about 0.011 seconds per frame, it presents faster, $0.049 \times 23 = 1.127$, than the results from some existing methods with CNN. The proposed algorithm is more suitable for a real-time tracking system than ANN.

3.2. Application of Intelligent Surveillance System

1st Event – Intrusion Detection

For this event, we set a region as a security area. If an invading person or car was found in this area, the system would detect the intruder and issue a warning. Figure 4 shows a flowchart of the 1st event of the intelligent surveillance system simulation. Figure 6 shows the results of intrusion detection for the PETS 2006, PETS 2009, and TownCenter

datasets. For the simulation, for each dataset, we set a square region as a security area. If the detected pedestrian broke into the set area, it would cover the region of the pedestrian with the color red, as shown in each figure.

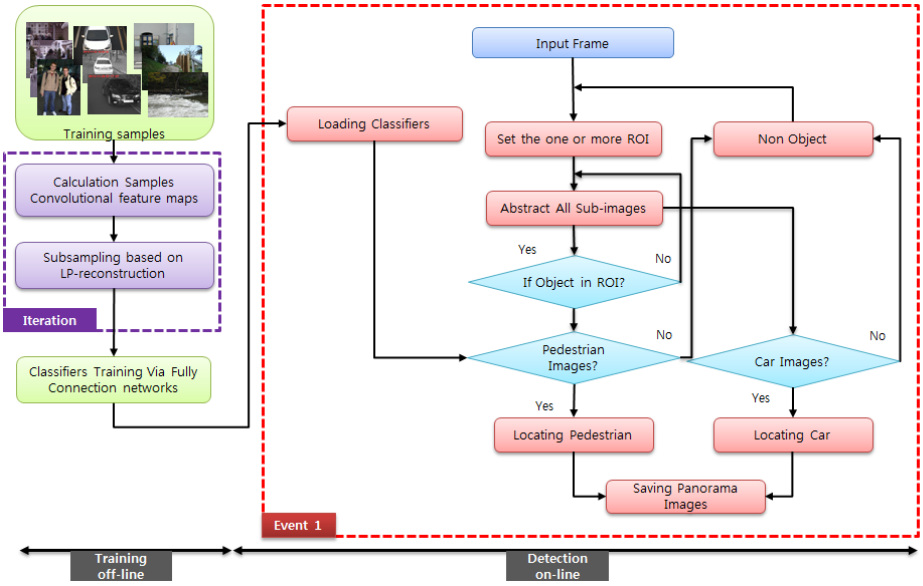


Figure 4. Flowchart of 1st Event

2nd event – pedestrian detection and count

For this event, we set a line as the detection base. Then, if a person or car crossed the line, the system would detect the event. Figure 5 shows the flowchart of the 2nd event of the intelligent surveillance system simulation. The detection results are shown in Figure 7.

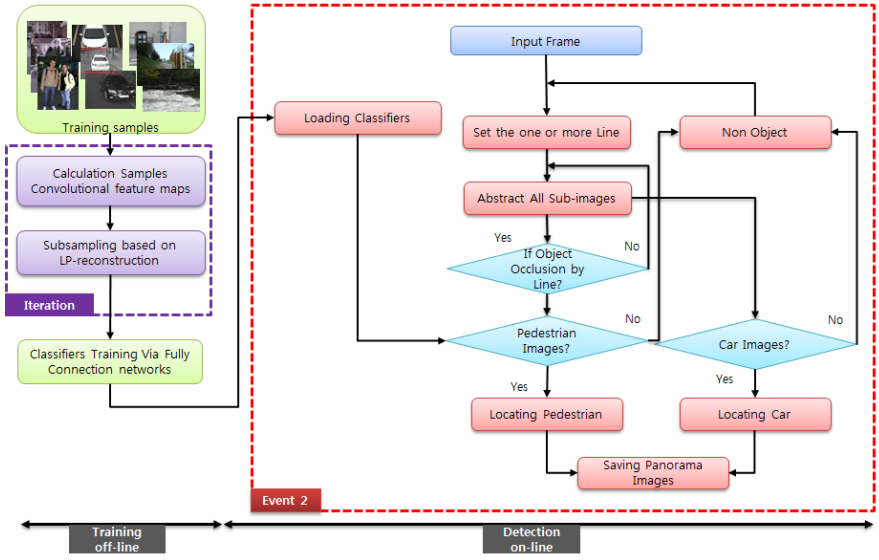
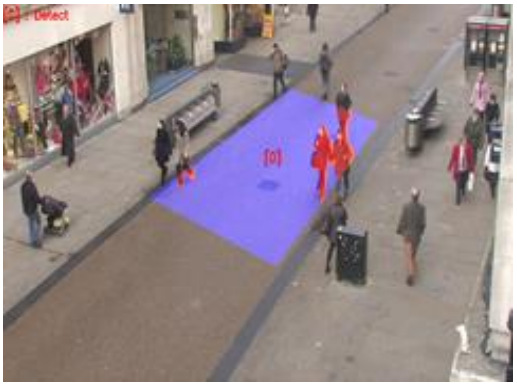
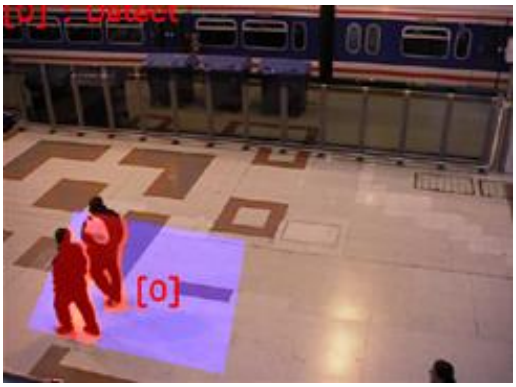


Figure 5. Flowchart of 2nd Event

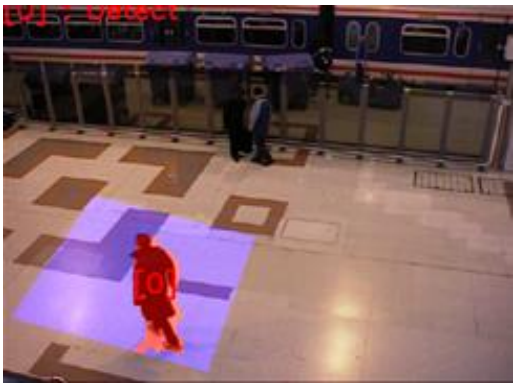


(a) Detection of persons for TownCenter dataset – 1

(b) Detection for persons for TownCenter dataset – 2

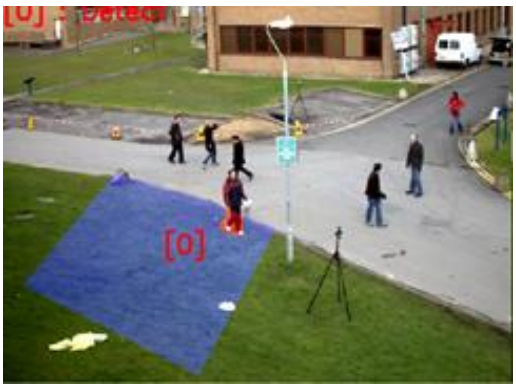


(c) Detection of persons for PETS2006 dataset – 1



(d) Detection for persons for PETS2006 dataset – 2

(e) Detection of persons for PETS2009 dataset – 1



(f) Detection for persons for PETS2009 dataset – 2

Figure 6. Detection Results for 1st Event

(a) Detection of persons for TownCenter dataset – 1

(b) Detection for persons for TownCenter dataset – 2

(c) Detection of persons for PETS2006 dataset – 1

(d) Detection for persons for PETS2006 dataset – 2

(e) Detection of persons for PETS2009 dataset – 1

(f) Detection for persons for PETS2009 dataset – 2

Figure 7. Detection Results for 2nd Event

4. Conclusions

As the existing deep learning algorithm for object detection and classification, the convolutional neural network (CNN) has been mainly used. However, it has shortcomings in real-time processing due to the computational complexity when processed together the other event algorithm.

In this thesis, we present an object detection technique that is composed of the modified CNN algorithm.

In CNNs, each filter is replicated across the entire visual field. These replicated units share the same parameterization (weight vector and bias) and form a feature map. In the pooling layer, to maintain the original information of the feature maps, we applied an effective method with low-pass reconstruction.

In the experiments, four datasets are used for training and four datasets are used for testing. Through the experiments, the proposed method shows similar results in object detection performance to those of the previous method with the CNN. The proposed algorithm shows a reduced processing time of about 0.011 seconds for each frame in comparison with the CNN, and the proposed method is proven to be more suitable for the real-time surveillance system. Finally, we simulated some intelligent surveillance system experiments with events.

A future CNN algorithm should focus on how many objects are distinguished and how to improve performance. To improve performance, we need to answer various questions. The first is how many depth of the network used. The second is how many second convolutional layers are needed. The third is how many units are needed in each layer. Thus, we need to find optimal conditions to improve performance.

Acknowledgments

This study was supported by the Dong-A University research fund.

References

[1] B. Javidi, "Image Recognition and Classification: Algorithms, Systems, and Applications", CRC Press, **(2002)**.
[2] M. S. An and D. S. Kang, "Motion estimation with histogram distribution for visual surveillance", In: Wireless and Optical Communications Conference (WOCC), 2010 19th Annual. IEEE, **(2010)**, pp. 1-4.
[3] M. S. An, S. W. Ha and D. S. Kang, "Object Tracking Method based on Particle Filter with Color and Texture Information", Journal of Korean institute of information technology, vol. 8, no. 11, **(2010)**, pp. 225-230.
[4] M. S. An and D. S. Kang, "Multi-object Tracking Method based on Stereo Vision with the Multiple Features Information Fusion", Journal of Korean institute of information technology, vol. 9, no. 7, **(2011)**, pp. 201-206.
[5] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints" Int. J. Comput. Vision, vol. 60, no. 2, **(2004)**, pp. 91-110.
[6] E. A. L. Gianicoloa, A. Brunia and M. Serinellib, "Environmental Health Surveillance Systems", PIAS-CNR, **(2008)**, pp. 129-144.
[7] R. Szeliski, "Computer Vision: Algorithms and Applications", Springer, **(2010).**
[8] Artificial intelligence - Wikipedia: https://en.wikipedia.org/wiki/Arti- ficial_intelligence.
[9] C. M. Bishop, "Pattern recognition and machine learning", Springer, **(2006)**.
[10] De. Sa and J. P. Marques, "Pattern recognition: concepts, methods and applications", Springer Science & Business Media, **(2012)**.
[11] G. E. Hinton, S. Osindero and Y. W. Teh, "A fast learning algorithm for deep belief nets", Neural computation, vol. 18, no. 7, **(2006)**, pp. 1527-1554.
[12] G. E. Hinton, "Deep Belief Networks", Scholarpedia, vol.4, no. 5, **(2009)**, pp. 5947.
[13] G. Bradski and A. Kaehler, "Learning OpenCV", Sebastopol, CA; O'Reilly, **(2008)**, pp. 316-369.
[14] X.G. Chen, "Pedestrian Detection with Deep Convolutional Neural Network", Computer Vision-ACCV 2014 Workshops. Springer International Publishing, **(2014)**.
[15] P. Glauner, "Deep Convolutional Neural Networks for Smile Recognition", (MSc Thesis) Imperial College London, Department of Computing, **(2015)**.
[16] INRIA Person Dataset: http://pascal.inrialpes.fr/data/human/.
[17] Adavanced Driver Assistance System Datasets: http://www.cvc.uab.es /adas/site/?q=node/7.
[18] Pets 2006 Benchmark data: http://www.cvg.rdg.ac.uk/PETS2006 /data.html.
[19] PETS 2009 Benchmark Data: http://www.cvg.reading.ac.uk/PETS2009 /a.html.
[20] AVG-TownCentre: https://motchallenge.net/vis/AVG-TownCentre/gt/.

Authors

Ming-Shou An, He received a B.S. degree from Yanbian University, China, in 2007, M.S. degree and Ph.D. degree in electrical engineering from Dong-A University, Busan, Korea, in 2009 and 2016. He is currently a senior researcher in Media Device Lab in Dong-A University. His research interests are signal processing and pattern recognition.

Dae-Seong Kang, He received a B.S. degree from Kyungpook National University, Daegu, Korea, in 1984, M.S. degree and D.Sc. degree in electrical engineering from Texas A&M University, in 1991 and 1994, respectively. He is currently full professor of the Department of Electronic Engineering, Dona-A University, Busan, Korea. His research interests are image processing and compression.