Personal and Ubiquitous Computing (2022) 26:385–394
https://doi.org/10.1007/s00779-019-01296-z

ORIGINAL ARTICLE

Research of multi-object detection and tracking using machine learning based on knowledge for video surveillance system

Hyochang Ahn [1] · Han-Jin Cho [1]

Received: 15 February 2019 / Accepted: 8 August 2019 / Published online: 28 August 2019

Abstract
Recently, as the risk of crime and accidents increases, interest in security and surveillance of individuals and the public is increasing rapidly, and video surveillance system technology is continuously developing. Reliable object detection in the system is the basis of all elements using image information and it is used in various applications using the information, so accurate object detection and tracking are needed. Therefore, we propose a system for analyzing images with a knowledge-based deep learning technology for multi-object recognition and tracking enhancement. Algorithms for recognizing objects using existing convolution neural network (CNN) classifiers have a problem that it is difficult to process in real time because the processing time is increased when there are a lot of objects to be classified in the image. Therefore, we propose an algorithm that combines optical flow while maintaining the recognition performance through a knowledge-based CNN. An optical flow-based tracker can forecast the position of objects in the next frame based on the position of objects in the current frame. A CNN-based detector can detect the position of objects through a knowledge-based mining method between the two images. CNN-based detectors also carry out mining method on current frame information. This detector can select more capacity features based on the background to more accurately forecast the location of the tracked targets and targets. The fusion of the tracker and detector compensates for accumulated errors that can occur in the tracker and for drift from the detector. The experimental results show that the proposed algorithm combining CNN and optical flow can detect and trace multiple objects in a video stream, and can carry out robust detection and tracing even in a complex environment.

Keywords Object detection · Object tracking · Knowledge · Machine learning · Surveillance

1 Introduction

Moving objects, motion tracking, and human detection are widely used in video conferencing systems and real-time surveillance systems [1–3]. Among them, real-time surveillance systems have the ability to automatically detect and track the presence of objects in an environment where not many moving objects appear and are mainly applied to computer vision systems that can replace human roles [2, 4]. This research has been used especially in the security surveillance field, weather observation system, intelligent traffic control system, and military field [4–6]. In addition, various studies are being conducted for high accuracy and fast processing for object recognition and tracking. Advances in information technology are increasing the need for surveillance to prevent theft and leak information [2, 7, 8]. Therefore, knowledge-based computing technologies, such as tracking and detection technologies of moving objects, have emerged as important technologies in the field of surveillance related to security.

Especially, the technology to determine the location of an object or region of interest that exists in surveillance images is an important issue in the field of research related to computer vision [2, 3]. Information about the location of an object is useful when it is necessary to infer high-level information and can help reduce the computation. Determining the position of an object from a given image can be done with two treatments [9–11]. The first is object detection and the second is object tracking. For the former, we first extract the features from the image and then learn the model related to the class of the object from this through the prior knowledge. Once the image

✉ Han-Jin Cho
hanjincho@kdu.ac.kr

Hyochang Ahn
youcu92@kdu.ac.kr

[1] Department of Energy IT Engineering, Far East University, Eumseong, Chungbuk, Republic of Korea

is given, the object can be detected using the learned model. Therefore, a machine learning model is mainly used for detection. On the other hand, in the latter case, only the pixel information of the region of interest to be searched is given rather than the object class to be searched, and the region having the highest similarity is searched for from the newly inputted image frame. That is, the object detection refers to a method of finding a previously known object on an input image, and the object tracking refers to a technique of finding an object using morphological similarity between adjacent frames in a moving image [12–14]. However, the experimental image or real scene in which the object to be tracked exists can have various scenarios, and the image quality and resolution also vary [11, 14]. Therefore, the abovementioned conventional tracking techniques cannot guarantee a high success rate in all of these situations. Recently, a method combining object detection and tracking has also been studied. This method is called tracking by detection. Various methods of machine learning are used to learn the detector to detect the object.

The factors that make it difficult to track objects in image recognition using existing object detection algorithms, include sudden movements of objects, changes in objects or scenes, shape changes of objects, occlusion due to the surrounding background, and changes in illumination. To cope with these factors, real-time object detection and tracking and learning algorithms that perform continuous learning of change are being studied. Therefore, we propose our method combining convolution neural network (CNN) and optical flow that multi-object detection and tracking using machine learning is based on the knowledge for a video surveillance system. The optical flow-based tracker can forecast the position of objects in the next frame based on the position of objects in the current frame. The CNN-based detector can detect the position of objects through the knowledge-based mining method between the two images [15, 16].

The rest of the section in this paper consists of the following. Section 2 introduces object detection and machine learning method, and Section 3 describes the proposed method. Section 4 shows the performance of the proposed method through experimental results. Finally, conclusions are given and future researches are explained in Section 5.

2 Related work

Important technologies in the field of image analysis include object detection, object classification, and object tracking [17, 18]. First, object detection defines the object of interest through the feature of the object's movement, shape, and shape in the input image and detects the object in the image [4, 5]. Object classification refers to classifying objects detected by methods, such as object detection into people, vehicles, and signs according to need, and classification methods according to need are very diverse [19]. Finally, object tracking means to perform motion path and path prediction of the object of interest selected by object detection and classification in a series of image frames. In this session, we briefly overview the technology of detecting and tracking multiple objects.

2.1 Object detection

Object detection requires an image processing technique to detect an object in an image captured by a camera. Object detection requires a process to distinguish foreground and background. However, since the computer cannot distinguish the foreground and the background from the image itself, the image processing technology capable of distinguishing the foreground and the background is required [20, 21]. Image processing technology analyzes and provides image information so that it can process and understand the characteristics of the image and the information required for the system.

2.1.1 Background subtraction

Background subtraction algorithm is the most widely used method for detecting objects moving within a still image, and the computation cost is relatively low compared to other methods [20]. Background subtraction technique uses the difference between the pre-modeled background and the current frame. At this time, if the difference value is larger than a specific threshold value, it is separated into the foreground (Fig. 1).

The pixel-based calculation is expressed as the following Eqs. (1) and (2):

$$\Delta X(x,y) = |X_t(x,y) - X_{t-1}(x,y)| \tag{1}$$

$$D(x,y) = \begin{cases} 1 & if\ \Delta X > T \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In Eq. (1), $\Delta X(x,y)$ represents the difference in brightness value at the coordinates (x,y) in the image. Also, $X_t(x,y)$ means the brightness value at the coordinates (x,y) in the image of the current frame. $X_{t-1}(x,y)$ denotes the brightness value at the coordinates (x,y) in the image of the previous frame. In Eq. (2), $D(x,y)$ denotes the binarized difference image, and T denotes the threshold value. If the value of Eq. (1) is larger than a specific threshold value T, it is expressed as 1. If it is smaller than or equal to 0, the binarized difference image $D(x,y)$ is generated. In the binarized difference image $D(x,y)$, the foreground is denoted by 1 and the background is represented by 0. As a result, the binarized difference image $D(x,y)$ is obtained in which the foreground is white and the background is black. This method can be applied to a continuous frame to distinguish the background from the foreground in the image, and it is possible to detect the object quickly because the operation cost is relatively low. However, there

Fig. 1 Background modeling result

is a problem that a different image may not be correctly obtained due to the movement of a camera, noise of an image, sudden change of illumination, shadow, and the like.

2.1.2 Gaussian mixture model

As mentioned above, there are some problems in separating the foreground and the background by applying the background difference technique to an actual general image [21, 22]. Much research has been done to separate the foreground and the background in a slightly better way. One of them, the Gaussian mixture model, is a method published by Stauffer and Grimson in 1999, which forms a background by separating foregrounds by modeling a mixture of k Gaussian probability distributions for each pixel of the image.

First, the expression of the Gaussian mixture model with each pixel value {1, ……, X_t} and k Gaussian models can be expressed as follows (Fig. 2):

$$P(X_t) = \sum_{i=1}^{k} \omega_{i,t} \cdot \eta\left(X_t, \mu_{i,t}, \sum_{i,t}\right) \tag{3}$$

In this case, k is the number of Gaussian models with a value of 3 to 5 and X_t is the value of the current pixel. $\mu_{i,\,t}$ is the weight of the ith Gaussian model in time t, $\omega_{i,\,t}$ is the average of the ith Gaussian model in time t, and $\eta_{i,\,t}$ represents the covariance matrix of the ith Gaussian model in time t. Also, μ represents a Gaussian probability density function defined by the following expression (4):

$$\eta(X_t, \mu, \sum) = \frac{1}{2\pi^{\frac{n}{2}}|\sum|^{\frac{n}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1}(X_t - \mu_t} \tag{4}$$

The k Gaussian distributions are ordered by ω/σ, and the first distribution, B is represented by the background model:

$$B = \text{argmin}_b\left(\sum_{k=1}^{b} \omega_k > T\right) \tag{5}$$

2.2 Machine learning

Machine learning is literally a machine learning. Assuming that $Y = F(X)$, the previous algorithms have implemented a function $F()$ for input X to generate the desired output Y. This has been used for a long time and has been used in most engineering applications today. However, machine learning can be defined as a process breaking this framework. It is a sort of black box solution that maps a large amount of input X to the desired output Y and finds $F()$ itself on the machine itself. There are many kinds of machine learning. The neural network (NN) which became famous in the advent of AlexNet in 2014, including well-known algorithms such as AdaBoost, Random Forest, and SVM (support vector machine).

2.2.1 Boosting

Boosting algorithm is a classification scheme that produces weak detectors from the learning data and makes them sequentially detectable, thereby creating a strong detector with high accuracy [23–25]. The sequential inspection method of cascade classifier method is used. The AdaBoost (adaptive boosting) algorithm is an algorithm developed to improve the performance of the next learning process by weighting the detector that yields more than 50 % of the results [25, 26].

The basic algorithm of boosting assigns classes as 0 and 1 to learning data, and assigns initial weight to all n equalizers

Fig. 2 Gaussian mixture modeling

for n detectors using feature information. Thereafter, it iterates T times and changes each weight so that the total error value is minimized. If the reliability of the detector is less than 50 %, it is automatically excluded because it is an inaccurate result than the random selection, and it is judged whether or not to detect by the weighted sum of weak detectors as in Eq. (6).

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} a_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} a_t \\ 0 & otherwise \end{cases} \quad (6)$$

2.2.2 SVM (support vector machine)

SVM works by finding hyperplanes that can classify them among data with multiple classes [27, 28]. It distinguishes it from other classification algorithms in that it finds a hyperplane with the largest margin of data among the many hyperplanes that can classify the data. Margin refers to the minimum distance (vertical distance) between hyperplane and data with different classes, where the vector closest to the hyperplane is defined as a support vector set [28–30]. The vector of the hyperplane is gradually adjusted to the misclassified samples on the hyperplane generated by the selected support vector set. The hyperplane can be expressed in all types of linear and nonlinear shapes, and there is also a method of classifying an existing vector into a hyperplane by mapping an extended vector using a kernel method. Although it is an algorithm developed for binary classification, it is often used in object recognition research through multiclass SVM of various advanced methods considering multiclassification. In general, when SVM has a relatively small number of learning data, it shows superior performance compared to other learning algorithms, and it has a merit of less calculation cost. Because of its sensitivity to parameter and kernel selection, it is important to choose the appropriate method depending on the type of data in the application phase.

3 Proposed method

One of the deep running algorithms, CNN, is one of the classifiers that have been spotlighted for their excellent performance in image signal processing. CNN is an artificial neural network that is designed to mimic human visual processing and is suitable for data processing. It has good performance in both video and audio signal processing compared to other deep learning structures, performing and emerging as the next generation of core classification algorithms.

Therefore, it is better to use CNN to do object recognition than to use other machine learning classification algorithms. However, the algorithm for recognizing objects using existing CNN classifiers has a problem that the processing time is increased and the real-time processing is difficult when the objects are classified in the image. In this paper, we propose a new method of recognizing objects by combining optical flows while extracting knowledge about various object and object characteristics in the image and maintaining recognition performance using CNN.

3.1 CNN (convolution neural network)

Recently, various studies such as image recognition, voice classification and recognition, and object detection have been conducted through deep learning, which has been proven in the area of machine learning due to the influence of the high-performance hardware (GPU) and big data [31–33]. CNN is a deep learning model proposed to overcome the problems of overfitting, local optimum convergence, and vanishing gradient through the structure of existing artificial neural networks [31].

Currently, CNN's speed has been greatly improved due to the high performance of hardware and the development of big data. You can easily collect the data needed for learning supervised and unsupervised learning, such as ImageNet, Flickr, and INRIA Person dataset, which provide photo sharing services. Many problems have been solved and it shows excellent performance in object classification and object detection. Unlike traditional object detection and learning methods, CNN has an excellent ability to automatically generate features for input images and has the advantages of feature extraction and learning in one structure.

CNN has a structure in which the knowledge-based features are created and classified by self-learning within the network. In order to detect an object, a feature point representing the feature of the knowledge based on the object is extracted, and the corresponding object is confirmed through the classifier using the extracted feature points. Each plane is a feature map, a set of units that are constrained so that the weights are the same. The input plane on each layer receives the processed image. That is, each unit in a layer receives input from a set of units found in small neighbor units in the previous layer. We can extract primitive knowledge-based features such as directionality, endpoints, and edges with local acceptance fields and neurons. Figure 3 demonstrates the structure of CNN.

In Fig. 3, CNN consists of feature extraction and classification. Feature extraction extract features use convolution and subsampling.

The convolution layer performs a convolution operation on the feature map generated through each step. The convolution mask used in the convolution operation and the element by element multiplication of the feature map are used, as shown in Fig. 4. The learning samples are learned in the proper format with prior knowledge and the line masks are used differently for each layer. Convolution masks are used differently for each layer do. The weight used in the convolution

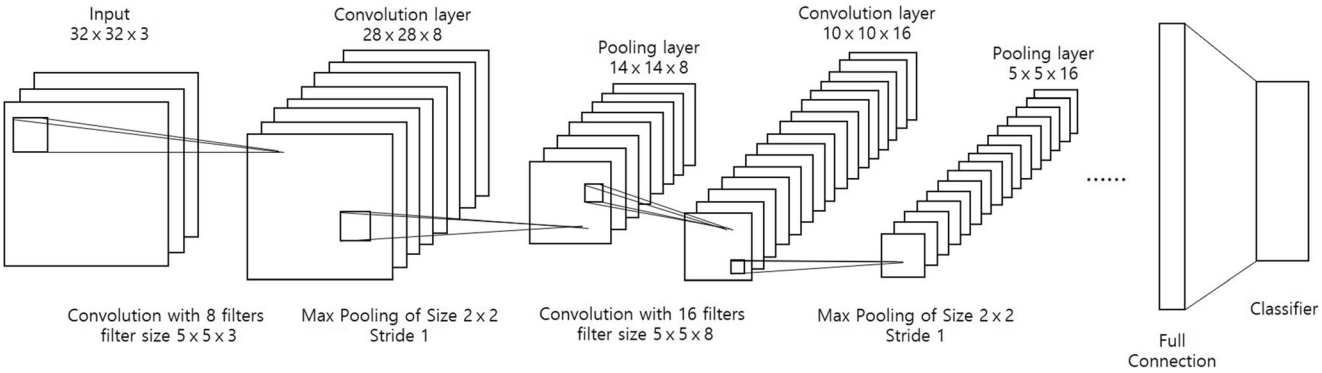

Fig. 3 CNN (convolution neural network) basic structure

operation is obtained during the learning process. Convolution is calculated and Eq. 7 shows how to calculate convolution.

$$y_{ij} = \sum_{p=0}^{K-1} \sum_{q=0}^{K-1} x_{i+p,j+q} * \omega_{pq} \tag{7}$$

In Eq. 7, i and j represent rows and columns and K is the size of the convolution mask. w represents a weight, x represents an input value, and y represents an output value. The subsampling extracts the maximum or minimum value in the mask using a mask on the learning image and calculates a mean value. Figure 5 shows subsampling.

After convolution and subsampling are repeated, the features are extracted and connected to a fully connected network, which has the equivalent structure as the multilayer perceptron, to generate and output a vector having the equivalent dimension as the number of outputs. In the extracting features, the sigmoid function of the active function is used. As the input value of the function increases or decreases, the problem of the gradient disappear occurs. However, the problem can be solved by using ReLU (rectified linear unit). Equation 8 is a formula for ReLU. When x is less than 0 for input value x, 0 is output as the output value. If x is greater than or equal to 0, x is used as the output value.

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \tag{8}$$

Finally, a full connection is made and the output value is finally classified using softmax.

$$\text{softmax}_N(x) = \frac{e^{x_n}}{\sum_{k=1}^{N} e^{x_k}} \quad , n \in \{1, 2, 3, \ldots\ldots, N\} \tag{9}$$

3.2 Optical flow

Optical flow is a motion vector calculated based on the intensity variation of two adjacent images $f(y, x, t-1)$ and $f(y, x, t)$ in a continuous image. The KLT (Kanade Lucas Tomashi) feature tracking algorithm uses optical gradient

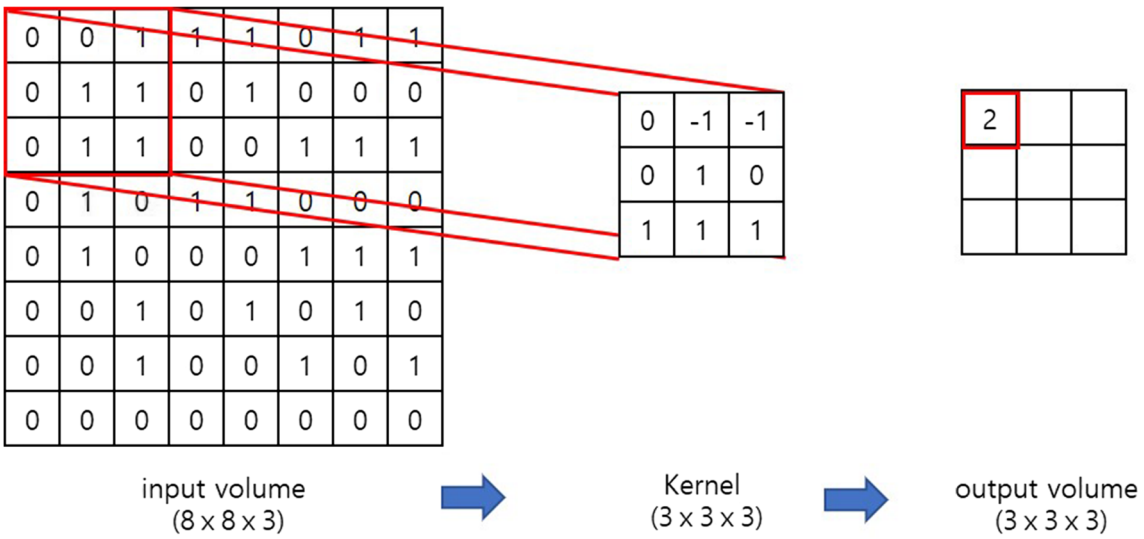

Fig. 4 Convolution operation

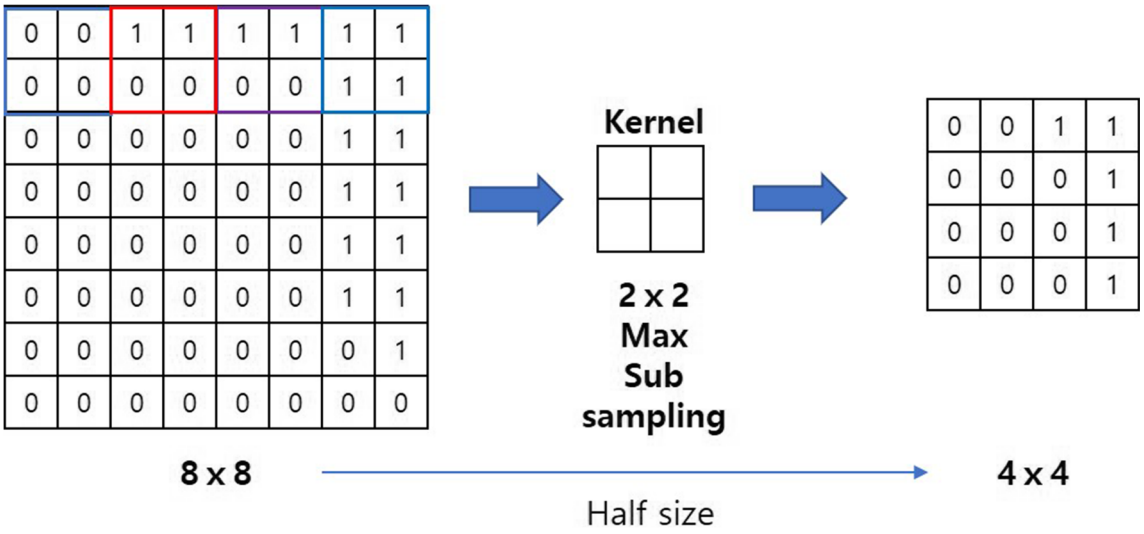

Fig. 5 Maximum value of subsampling example

detection technique to select and track feature points that can be tracked using the slope characteristic of the image intensity of the $N \times N$ image block based on the center point of the characteristic region within one frame [34, 35]. The optical detection method is a method of detecting motion by finding a set of characteristic optical vectors in two consecutive frames as one of the area-based detection methods. Optical flow means the moving distance and the directionality of the pixel with the same brightness pattern in the continuous frame. After acquiring optical flow for the whole image, the optical flow vector of the moving pixels is collected and detected as the moving object. KLT uses these optical flows to track features as follows. KLT assumes that the features of the object to be traced in

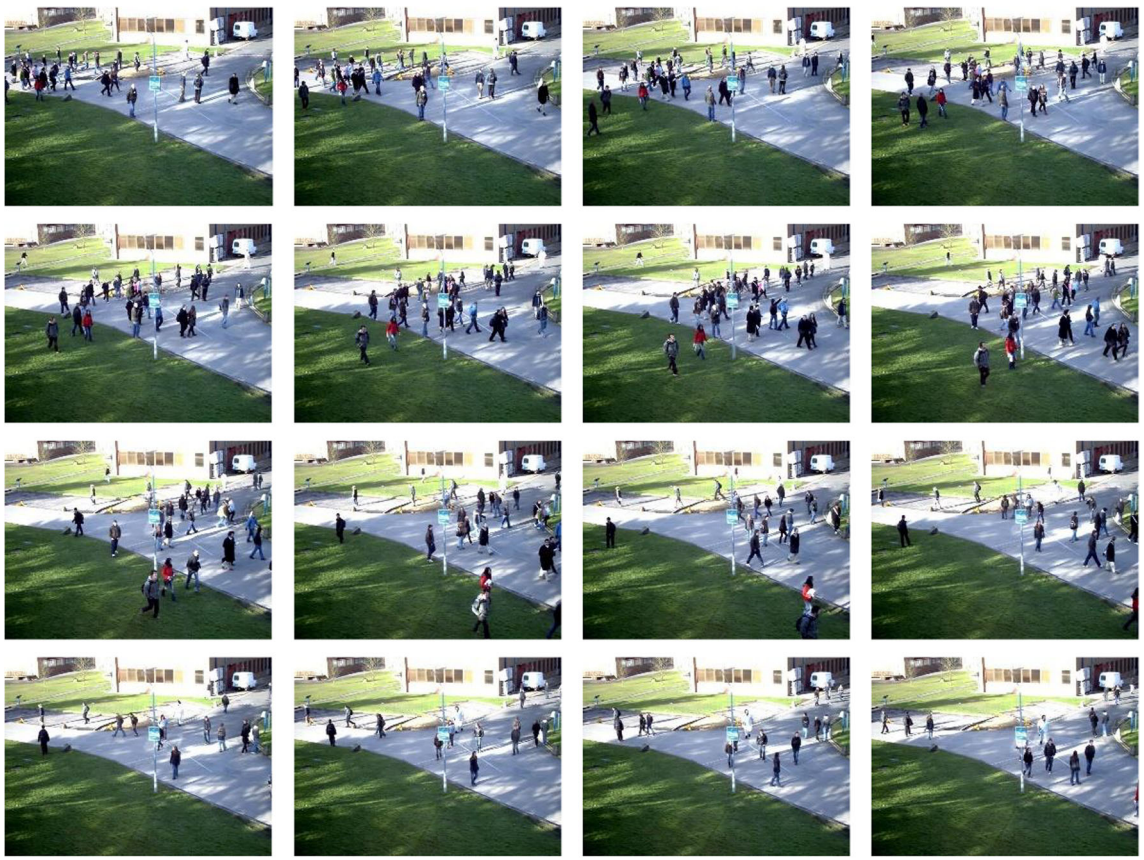

Fig. 6 Data video no. 1

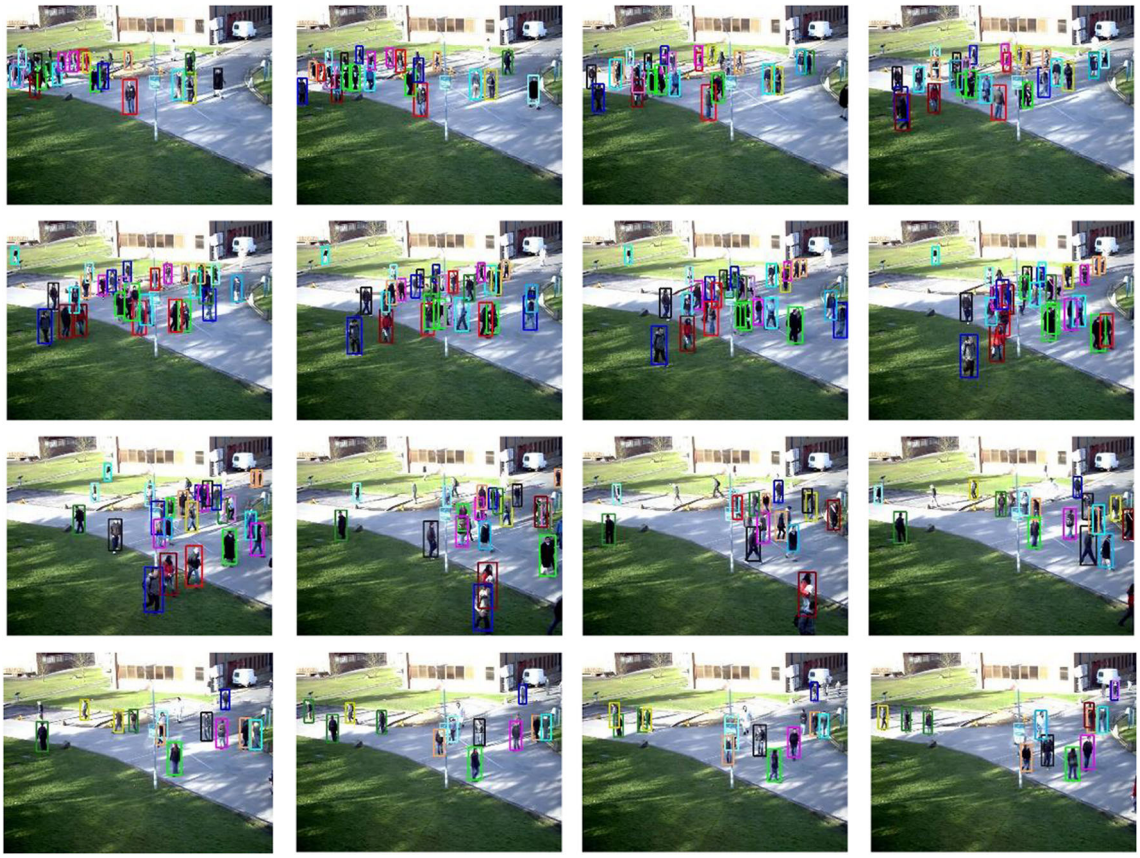

Fig. 7 Results for a proposed algorithm of data video no. 1

the continuous image are minimized. So, we have a space position and a time as a variable in the first input. In Eq. 10, we have location information for x, y, and time variable t as a basis as follows:

$$f(y,x,t+r) = I\Big(x{-}\delta(x,y,r), y{-}\eta(x,y,t,r) \tag{10}$$

In the above equation, x, y, t have displacement delta and mu when an arbitrary time r has passed. Here, delta and mu must determine the amount of movement. First, when mu is the current image x and d is the motion vector, delta has the Eq. 11 as follows:

$$\delta = D_x + d \tag{11}$$

where D is the role of determining the feature points with the gradient matrix and has the form 2.18 as follows:

$$D = \begin{vmatrix} d_{\mathrm{xx}} & d_{\mathrm{xy}} \\ d_{\mathrm{yx}} & d_{\mathrm{yy}} \end{vmatrix} \tag{12}$$

Then, the sum of the current position x and the motion vector d is set to J, and the current position is I, as shown in Eq. 13.

$$J(\mathrm{Ax}+d) = I(x) \tag{13}$$

Also, the error of the features of the KLT tracker is denoted by e, and the feature point due to the error of the generated features is removed, and the motion vector is averaged as follows:

$$e = \iint_W [J(\mathrm{Ax}+d){-}I(x)]^{20} w(x)\mathrm{dx} \tag{14}$$

where W denotes the size of the object to detect the feature point, w denotes the weight of the motion vector as a function using the normal distribution, and is used here to filter the noise. The problem arising from the difference between the images J and I can be linearized to Eq. 14 as follows:

$$T = \iint_W \begin{vmatrix} U & V \\ {V_{\mathrm{yx}}}^T & Z \end{vmatrix} \mathrm{wdx} \tag{15}$$

U, V_t, and Z in Eq. 15 are expressed by Eqs. 16, 17, and 18, respectively.

$$U = \begin{vmatrix} x^2 g^2 x & x^2 \mathrm{gxgy} & \mathrm{xyg}_x^2 & \mathrm{xyg}_x g_y \\ x^2 \mathrm{gxgy} & x^2 g^2 y & \mathrm{xyg}_x g_y & \mathrm{xyg}_x \\ \mathrm{xyg}_x & \mathrm{xyg}_x g_y & y^2 g_x^2 & y^2 g_x g_y \\ x^2 \mathrm{yg}_x g_y & \mathrm{xyg}_y^2 & x^2 g_x g_y & y^2 g_x^2 \end{vmatrix} \tag{16}$$

Fig. 8 Data video no. 2

$$V^T = \begin{bmatrix} xg_x^2 & \mathrm{x}g_x g_y & yg_x^2 & \mathrm{y}g_x g_y \\ \mathrm{x}g_x g_y & \mathrm{x}g_y^2 & \mathrm{t}g_x g_y & \mathrm{t}g_y^2 \end{bmatrix} \quad (17) \qquad Z = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_x^2 \end{bmatrix} \quad (18)$$

Fig. 9 Results for a proposed algorithm of data video no. 2

4 Experimental result

The proposed algorithm was tested using Python on Windows 10. In addition, datasets were detected and tested using OpenCV library. In the experiment, we evaluate the importance of maximum margin learning for CNN and compare objects with tracking techniques in different ways. In this paper, we experimented on the proposed CNN tracker based on knowledge for various real-time moving images.

The experiments were designed to evaluate the robustness and performance of several components of the algorithm under different circumstances. The image data used in the experiments consisted of images of various environments. In this paper, performance evaluation is performed based on two items of object tracking accuracy and processing time. The criteria for accuracy and processing time are as follows. Accuracy indicates whether the result of object tracking contains actual objects. The processing time represents the total time from the start to the end of the object tracking algorithm. These two items were measured by CAMShift and optical flow and compared with the proposed method. The evaluation of the detection rate and accuracy used in this paper measured tracking success and tracking failure. The criteria for tracking success and tracking failure are as follows.

As a general evaluation criterion, the tracking success was determined by considering the size of the object rather than the accuracy of evaluating only the position of the object. The result of the proposed tracking success is that the overlap ratio threshold of the ground truth and the tracking result is 0.5 in order to evaluate the success of the tracking. In addition, the tracking success rate was evaluated as the difference between the center coordinates of the ground truth and the center coordinates of the tracking results as 25. Figures 6 and 7 show the results of detecting and tracking multiple objects using the image data and the proposed method.

Experimental results of multiple object tracking of the proposed algorithm show the number of tracking success frames, the number of tracking fail frames, the total number of frames and the accuracy, and the total processing time (Figs. 8 and 9). Table 1 shows the comparison between the proposed method and the conventional method.

Experimental results represent that the proposed method has higher accuracy than conventional algorithms. The results represent that the rate of tracking success increases from 125 frames to 177 frames on average. On the other hand, the execution time is somewhat slowed from 4.72 to 4.97 s. However, there is no momentous difference in performance. As a result, the performance of the proposed method is somewhat delayed, but the accuracy of the proposed method is greatly improved.

Table 1 The comparison of objects tracking results

	CAMShift	Optical flow	Propose method
Tracking success frame	124	125	177
Tracking failed frame	67	66	14
Total frame	191	191	191
Accuracy (%)	64.92	65.44	92.67
Processing time (s)	4.64	4.80	4.97

5 Conclusions

Recently, video surveillance and security monitoring system technology has been rapidly developed to monitor various situations and respond quickly, and related researches are actively being carried out. We propose a system for analyzing images with a knowledge-based machine learning technology for a multi-object recognition and tracking enhancement. Algorithms for recognizing objects using existing CNN classifiers have a problem that it is difficult to process in real time because the processing time is increased when there are many objects to be classified in the image. Therefore, we propose an algorithm that combines optical flow while maintaining the recognition performance through existing the CNN. The optical flow-based tracker is used to forecast the position based on the position of the next frame. The CNN-based detector can detect the position of objects through the knowledge-based mining method between the two images. CNN-based detectors also carry out mining method on current frame information. The detector can select more capacity features based on the background to more accurately forecast the location of the tracked targets and targets. The fusion of the tracker and detector compensates for accumulated errors that can occur in the tracker and for drift from the detector. Future research needs to study the surveillance system that can quickly detect multiple objects and predict motion using the proposed method.

References

1. Collins RT, Lipton AJ, Kanade T, Fujiyoshi H, Duggins D, Tsin Y, Tolliver D, Enomoto N, Hasegawa O, Burt P, Wixson L (2000) A system for video surveillance and monitoring. The Robotics Institute, Carnegie Mellon University, Pittsburgh, pp 1–68
2. Ahn H, Lee Y (2016) Performance analysis of object recognition and tracking for the use of surveillance system. J Ambient Intell Humaniz Comput 7(5):673–679
3. Valera M, Velastin SA (2005) Intelligent distributed surveillance systems: a review, IEE Proc Vision Image Signal Process, vol 152. No. 2. IET, pp 192–204
4. Wu Y, Jongwoo L, Ming-Hsuan Y (2015) Object tracking benchmark. IEEE Trans Pattern Anal Mach Intell 37(9):1834–1848
5. Yi W, Lim JW, Yang MH (2013) Online object tracking: a benchmark, In: CVPR, pp 2411–2418

6. Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. ACM Comput Surv (CSUR) 38(4):13
7. Kang S, Chung K, Lee J (2014) Development of head detection and tracking systems for visual surveillance. Pers Ubiquit Comput 18(3):515–522
8. Alostaz A, Hamed B (2016) Optimized automated tracking of a moving object with a robotic eye system. Control Intell Syst 44(1)
9. Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 25(5):564–577
10. Allen JG, Xu RY, Jin JS (2004) Object tracking using camshift algorithm and multiple quantized feature spaces. In: Proceedings of the Pan-Sydney area workshop on Visual information processing. Australian Computer Society, Inc., Kent Town, pp 3–7
11. Ahn H, Shin I (2018) Study on a robust object tracking algorithm based on improved SURF method with CamShift. Journal of the Korea Society of Computer and Information 23(1):41–48
12. Grabner H, Matas J, Gool LJV, Cattin PC (2010) Tracking the invisible: learning where the object might be, CVPR 2010, pp 1285–1292
13. Babenko B, Yang M, Belongie S (2011) Robust object tracking with online multiple instance learning. IEEE Trans Pattern Anal Mach Intell 33(8):1619–1632
14. Babenko B, Yang M, Belongie S (2009) Visual tracking with online multiple instance learning, In ComputerVision and pattern recognition, 2009. CVPR 2009. IEEE conference on, pp 983–990
15. Kalal Z, Mikolajczyk K, Matas J (2012) Tracking-learning-detection. IEEE Trans Pattern Anal Mach Intell 34(7):1409–1422
16. Chen Y, Yang X, Zhong B, Pan S, Chen D, Zhang H (2016) CNNTracker: online discriminative object tracking via deep convolutional neural network. Appl Soft Comput 38:1088–1098
17. Zhang K, Zhang L, Yang M (2012) Real-time compressive tracking. In: European conference on computer vision. Springer, Berlin, pp 864–877
18. Takala V, Pietikäinen M (2007) Multi-object tracking using color, texture and motion, CVPR 2007, pp 1–7
19. Jepson AD, Fleet DJ, El-Maraghi TF (2003) Robust online appearance models for visual tracking. IEEE Trans Pattern Anal Mach Intell 25(10):1296–1311
20. Elgammal A, Duraiswami R, Harwood D, Davis LS (2002) Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc IEEE 90(7):1151–1163
21. Stauffer C, Grimson WEL (1999) Adaptive background mixture models for real-time tracking, In computer vision and pattern recognition, IEEE Computer Society Conference on, vol 2, pp 2246
22. Gómez-Romero J, Serrano MA, Patricio MA, García J, Molina JM (2012) Context-based scene recognition from visual data in smart homes: an information fusion approach. Pers Ubiquit Comput 16(7):835–857
23. Grabner H, Grabner M, Bischof H (2006) Real-time tracking via on-line boosting. In: Bmvc, vol 1, No. 5, pp 6
24. Grabner H, Bischof H (2006) On-line boosting and vision, CVPR (1), pp 260–267
25. Viola P, Jones MJ (2001) Rapid object detection using a boosted cascade of simple features, CVPR 2001, vol 1, pp 1–1
26. Han G, Shen J, Liu L, Qian A, Shu L (2016) TGM-COT: energy-efficient continuous object tracking scheme with two-layer grid model in wireless sensor networks. Pers Ubiquit Comput 20(3): 349–359
27. Z S, Yu X, Sui Y, Zhao S, Zhang L (2015) Object tracking with multi-view support vector machines. IEEE Trans Multimedia 17(3): 265–278
28. Avidan S (2004) Support vector tracking. IEEE Trans Pattern Anal Mach Intell 26(8):1064–1072
29. Bai Y, Tang M (2012) Robust tracking via weakly supervised ranking SVM, CVPR 2012, pp 1854–1861
30. Cortes C, Vapnik VN (1995) Support-vector networks. Mach Learn 20(3):273–297
31. Luo L (2018) Network text sentiment analysis method combining LDA text representation and GRU-CNN. Pers Ubiquit Comput (1–8)
32. Steinkrau D, Simard PY, Buck I (2005) Using GPUs for machine learning algorithms. ICDAR 2005:1115–1119
33. Chellapilla K, Kumar SP, Simard P (2006) High performance convolutional neural networks for document processing, Tenth international workshop on Frontiers in handwriting recognition
34. Senst T, Eiselein V, Sikora T (2012) Robust local optical flow for feature tracking. IEEE Trans Circuits Syst Video Technol 22(9): 1377
35. Barron JL, Fleet DJ, Beauchemin SS (1994) Performance of optical flow techniques. Int J Comput Vis 12(1):43–77