Efficiency analysis of dynamic object detection in computer vision system

O A Pakhomova and O Ja Kravets
Department of automated and computing systems, Voronezh State Technical University, VSTU, 396066, Russia, Voronezh, Moscow avenue, 179
E–mail: csit@bk.ru

Abstract. The concept of computer vision and its key aspects are given. Problems of dynamic objects detection are considered. The latest approaches are described to improve the accuracy and speed of recognition. Their main advantages and disadvantages are considered. The structure of the computer vision system and the interaction of its components are shown. A method for embedding a motion detection module to increase the efficiency of the recognition subsystem is proposed. A motion vector search algorithm is described, the main idea of which is to divide a frame into blocks and search for similar parts in adjacent frames. Experimental results showing the effectiveness of the approach are demonstrated.

1. Introduction
Computer vision is a technology that allows finding, tracking, classifying and identifying objects of the real world, extracting and analyzing the information received. Machine and computer vision are related areas. Computer vision can be called a general set of tools, and machine vision shall be regarded as an area of application. A key aspect in the field of computer vision is the problem of automatic image recognition of dynamic objects.

Image recognition is the process of extracting common characterizing object features from the total mass of irrelevant data in order to determine their belonging to a certain class. In most cases, to uniquely identify a dynamic object, just one of its images is not enough, since the information about the characteristics may be incomplete. Accordingly, modern recognition systems solve the problem of detecting objects and their maintenance throughout the entire surveillance period. And with each new frame the accumulated recognition result is refined and the likelihood of unambiguous identification of the object is increased [1–2].

A list of the leading machine learning technologies that are actively used in conjunction with video analytics systems for detecting graphic objects:

1. TensorFlow is a library of machine learning and deep research of neural networks from Google, which is based on data flow graphs. It is enough to describe the architecture of the neural network and its parameters. Then this data is sent to the TensorFlow session, where all the calculations that are required to produce the results take place. Calculations are represented as a directed graph, in which the edges are the
paths of data movement. The data is represented as tensors – multidimensional data arrays with variable size. Tensors go from node to node along the edges of the graph [3].

2. TFlearn is a modular and open machine learning library designed to work with higher levels of the TensorFlow API and makes the development much easier [3].

3. Microsoft Cognitive Services are a set of cloud services that provide their customers with the opportunity to identify, describe, and process images, sound, and much more using the software interaction interface (API) [4].

4. OpenCV (Open Source Computer Vision Library) is a library of computer vision algorithms, image processing, and general-purpose numerical algorithms with open source. Implemented in C/C++, Python, Ruby, Matlab, and other languages [4].

5. The Accord.NET Image Processing and Machine Learning Framework are a set of machine learning libraries written in the C# programming language and allowing access to a set of highly efficient detection algorithms [4].

In general, modern machine learning algorithms are based on the neural network approach.

2. Problems of dynamic objects detection

Recently, using algorithms based on convolutional neural networks, it has been possible to achieve significant success in the field of static object detection [5].

The main difference of the algorithms is the viewing of images through a small window moving to the right and down [6–8]. There are some important features inside the region of interest, such as a vertical line, a circle, a square, etc. The calculation of such features is performed during the training of the convolution network. The location of important features is shown on the feature maps. Moreover, their combination in a specific area can indicate the presence of a more complex feature. For example, some feature maps can detect a bicycle by lines and circles (based on simple feature maps) [9].

Currently, the mechanism for constructing the convolutional neural network architecture includes the following steps: firstly, extracting feature maps set across the entire input image \( F = N_{feat}(I) \) \([3, 10–15]\), secondly, generating the result of neural network based on feature maps set \( Y = N_{det}(F) \) \([16–21]\). In other words, the result of neural network means determining the class of object, e.g. bicycle from a set of lines, circles, etc.

Direct application of the methodology based on the convolutional neural network for the detection of dynamic objects entails the following problems:

- Extracting feature maps set for each image is very time-consuming. So, it is unacceptable, especially for real-time systems.
- There is reduced recognition accuracy due to blurring, defocusing or the presence of other artifacts on the video.

3. Approaches to problem-solving

3.1. Sparse Feature Propagation

One of the approaches to solution of dynamic object detection problem is Sparse Feature Propagation method [1] based on the concept of key-frames (figure 1). In general, processing a frame with a similar appearance among neighboring frames entails a similar computational complexity. So, calculating of the feature maps set \( F = N_{feat}(I) \) is necessary only for key-frames. The value of feature maps set on any non-key-frames extends from its previous key-frames.
The intra-frame pixel-wise motion is recorded in a two-dimensional motion field $M_{i \rightarrow k}$, where two-dimensional motion field is part of a frame (matrix of pixels), the approximation to which can be seen on nearby frames. The propagation from key frame $k$ to frame $i$ is shown on equation 1.

$$ F_{k \rightarrow i} = W(F_k \cdot M_{k \rightarrow i}). \quad (1) $$

$W$ – feature warping function. Then the detection network (see section 2) $Y = N \det(F_{k \rightarrow i})$ works on $F_{k \rightarrow i}$, as approximation to the real feature $F_k$, instead of computing the result based on $F_i$.

However, propagated features are only approximate and error prone. This can impair the accuracy of recognition.

![Image](image.png)

**Figure 1.** Approaches to dynamic object detection.

3.2. Dense Feature Aggregation

The Dense Feature Aggregation method [2] gives the possibility of improving the quality and accuracy of recognition due to the aggregation of common features from neighboring frames (figure 1). This approach allows preventing the decrease in accuracy caused by the presence of blurring or overlapping objects in some images.

Feature maps set $F$ is calculated for frames of any kind.

For $\forall i \in F$ in the range $[i-r, i+r]$ are warped in $i$ as shown on equation 2, forming feature maps set $\{F_{k \rightarrow i} | k \in [i-r, i+r]\}$ where $r \in [2; 12]$ frames. The aggregated feature maps $F^-_i$ on frame $i$:

$$ F^-_i(p) = \sum_{k \in [i-r, i+r]} W_{k \rightarrow i}(p) \ast F_{k \rightarrow i}(p), \forall p \quad (2) $$
Weight $W_{k \rightarrow i}$ is calculated as the similarity between propagated feature maps $F_{k \rightarrow i}$ and real feature maps $F_i$. The weight is normalized at every location $p$ (point of frame) over nearby frames $\sum_{k \in [i-r,i+r]} W_{k \rightarrow i}(p) = 1$.

Dense Feature Aggregation method significantly improves the accuracy of dynamic objects detection. However, it is very slow due to repetitive calculations of $F$.

3.3 Motion vector search module
Computer vision systems are designed to solve highly specialized tasks, such as motion detection and recognition of license plates on video sequences, calculation and classification of objects on the conveyor during passenger baggage screening, etc.

The main advantage of such systems for people is the lack of fatigue and the possibility of continuous monitoring [22–23].

The common structure of the computer vision system includes the following components (figure 2):

- a set of video cameras,
- an image capture and conversion module,
- a recognition module,
- a database and a knowledge-base,
- a decision support system.

Computer vision systems generally work in real time. This imposes temporary restrictions on the recognition subsystem. In other words, the number of frames processed per second affects the recognition accuracy and, therefore, the number of fixations of high-speed objects as well. Based on the existing problems with the detection of dynamic objects described in section 2, it is proposed to integrate the motion detection module into a common system to increase recognition speed and reduce the number of false alarms (figure 2).

Input data of the embedded module are video data from surveillance cameras. Output data of the embedded module include:

- a data structure consisting of the analyzed image, the coordinates of the detection zone and the identifier for linking with other objects.
- a data structure consisting of an object identifier and a message to stop tracking it.

A feature of the module under consideration is the transmission of informational messages about an object in a separate parallel data stream. The accumulated information about the object is entered into the recognition subsystem. Thus, it will be possible to distribute the computational load between the two modules of the system.

The detection module consists of the following components:

- a temporary storage of raw photos,
- a database and a knowledge-base for storing intermediate results,
- the subsystem of the search for motion vectors, performing frame-by-frame analysis, identifying key features and drawing up a map of their movement throughout the video sequence.

By feature is meant a part of the object of observation (a block of pixels) that appeared on the key-frame and is repeated on subsequent intermediate frames. The feature maps show which the block of interest was first seen on the frame and its further movement on subsequent frames.

Figure 2 shows the interaction of the main components of the video analytics system. The algorithm for finding the motion vectors of dynamic objects is presented in figure 3–4.
Figure 2. Distributed system structure.

Figure 3. Detection motion algorithm.
Figure 4. Continuation of detection motion algorithm.

Formula 3 for determining the similarity of blocks that have the same position at different points in time:

\[ \text{result} = \sum_{k=0}^{n-1} \sum_{l=0}^{m-1} (I_t[i+k, j+l] - I_{t-1}[i+k, j+l])^2 \quad (3) \]

1* – the designation of this formula in picture 3. \( I_t \) – the array of pixel brightness at the moment \( t \); \( n, m \) – width and height of pixel block. \( i, j \) – block coordinates;

Formula 4 is for searching for a block by neighboring frames, which is similar to the one under consideration.

\[ \text{result} = \sum_{k=0}^{n-1} \sum_{l=0}^{m-1} (I_t[i+k, j+l] - I_{t-1}[i+(q*a) + k, j + (w*b) + l])^2 \quad (4) \]

2* – the designation of this formula in picture 4. \( q, w \) – a step right, left, up or down on a previous frame to search for a similar block.

The progress of the algorithm consists of the following steps:

1) Entering an array of images, each of which is an array of brightness.
2) Entering the brightness threshold (If the brightness of two pixels does not exceed a specified number, then they are considered equal).
3) Entering pixel block size (e.g. 16x16).
4) Calculation of the number of blocks by the width and height of the image.
5) Copying of all the blocks of the first frame into the resulting array of motion vectors.
6) Iteration over all images. If all images are processed, then exit.
7) Iteration over image width with a step size equal to the block width. If processing is completed, go to step 15.
8) Taking a thread for parallel calculating (Operations in steps 9–14 are performed in parallel).
9) Iteration over image height with a step size equal to the block height. If processing is completed, go to step 7.
10) Comparing blocks of the current and the previous frame by the formula 1*.
11) If the blocks are equal, then go to step 7.
12) Searching for a similar block in the previous frame in some neighborhoods of the current position of the block using the formula 2*.
13) If a similar block is found, then add its coordinates to the resulting array and go to step 15.
14) Adding a new block to the resulting array.
15) Synchronization of threads, waiting for the completion of calculations.
16) Definition of zones of displacement of blocks and sending each in a separate parallel stream and go to step 6.

4. Experiments
ImageNet VID dataset [4] is a prevalent large-scale benchmark for video object detection. It is used for this experiment. Model training and evaluation are performed on the 3,862 video snippets from the training set and the 555 snippets from the validation set. There are about 25 frames per second for the video snippets. The images are resized to a shorter side of 600 pixels for the image recognition network, and a shorter side of 300 pixels for the flow network. 30 object classes are used for experiment. Stochastic Gradient Descent training is performed (samples are selected randomly).

The metric to measure the accuracy of object detectors (mAP) using the considered methods is calculated and displayed on the graph. It precisely measures the “false positive rate” or the ratio of true object detections to the total number of objects.

Calculating mAP using equations 5:

$$mAP = \frac{1}{|classes|} \sum_{c \in classes} \frac{TP(c)}{TP(c) + FP(c)},\tag{5}$$

where True Positive $TP(c)$ - a proposal was made for class $c$ and there actually was an object of class $c$; False Positive $FP(c)$ - a proposal was made for class $c$, but there is no object of class $c$.

Overall comparison results are shown in figure 5.

![Figure 5. Speed-accuracy trade-off curves for the detection methods.](image-url)
As for Sparse Feature Propagation [1], by varying key frame duration from 1 to 10, it can achieve a five–fold acceleration speedup with moderate accuracy loss (within 1%). As for Dense Feature Aggregation [2], by varying the temporal window to be aggregated from ±1 to ±10 frames, it improves the mAP score by 2.9% but the runtime speed is about three times slower than the per–frame baseline. The method is not able to reach an acceptable processing speed (about 20 fps). As for the motion detection module, the mAP score is higher than Sparse Feature Propagation with a sufficiently high number of frames per second.

Conclusion

In recent works aimed at solving the problems of dynamic object detection, the methods of Sparse Feature Propagation and Dense Feature Aggregation have been discussed. However, problems of unacceptable computational costs and loss of recognition accuracy have been identified. To eliminate them, it was proposed to embed the motion detection module into the general machine vision system, which receives images from video cameras at the input, and the accumulated information about the objects of observation is transmitted at the output in parallel streams. The detection module delegates the responsibility for searching and detecting motion, hiding unnecessary information and providing only the necessary areas for further recognition.

The efficiency calculations presented in section 4 have showed the feasibility of introducing the proposed approach into real computer vision systems.

References

[1] Zhu X, Xiong Y, Dai J, Yuan L and Wei Y 2017 Deep feature flow for video recognition CVPR
[2] Zhu X, Wang Y, Dai J, Yuan L and Wei Y 2017 Flow-guided feature aggregation for video object detection ICCV pp 1–8
[3] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large–scale image recognition ICLR pp 3, 4
[4] Ren S, He K, Girshick R and Sun J 2015 Towards real–time object detection with region proposal networks Advances in Neural Information Processing Systems 28 pp 1–8
[5] Redmon J, Divvala S, Girshick R, and Farhadi A 2016 Unified, real-time object detection CVPR pp 1–5
[6] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C – Y and Berg A C 2016 Single shot multibox detector ECCV pp 2, 5
[7] Redmon J and Farhadi A 2016 Yolo9000: better, faster, stronger Preprint 1612.08242 p 6
[8] Goyal P, Lin T – Y, Girshick R, He K and Dollar P 2017 Focal loss for dense object Preprint 1708.02002 pp 2, 5
[9] Zeiler M D and Fergus R 2014 Visualizing and Understanding Convolutional Networks ECCV pp 818–833
[10] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions CVPR pp 1, 2
[11] Szegedy C, Ioffe S, Vanhoucke V and Alemi A 2016 Inceptionv4, inception – resnet and the impact of residual connections on learning Preprint 1602.07261 p 2
[12] Huang G, Liu Z, Weinberger K Q and van der Maaten L 2017 Densely connected convolutional networks CVPR pp 2, 5
[13] Xception F 2017 Deep learning with depthwise separable convolutions CVPR p 5
[14] Howard A G, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M and Adam H 2017 Mobilenets: Efficient convolutional neural networks for mobile vision applications Preprint 1704.04861 pp 2, 5
[15] Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y and Guadarrama S 2017 Accuracy trade-offs for modern convolutional object detectors *CVPR* pp 1–5

[16] Girshick R, Donahue J, Darrell T and Malik J 2014 Rich feature hierarchies for accurate object detection and semantic segmentation *CVPR* pp 3–5

[17] He K, Zhang X, Ren S, and Sun J 2014 Spatial pyramid pooling in deep convolutional networks for visual recognition *ECCV* pp 5

[18] Girshick R 2015 Fast r-cnn *ICCV* pp 3–5

[19] Dai J, Li Y, He K and Sun J 2016 Object detection via region-based fully convolutional networks *Advances in Neural Information Processing Systems* 28 p 3

[20] Dollar P, Lin T–Y, Girshick R, He K, Hariharan B and Belongie S 2017 Feature pyramid networks for object detection *CVPR* p 4

[21] Dimidova L, Egin M and Tishkin R 2018 Novel modifications of the multi-objective genetic algorithm for SVM classifier *International Journal on Information Technologies & Security* 2 91–100

[22] Russakovsky O, Deng J, Su H, Krause J, Satheesh S 2015 Imagenet large scale visual recognition challenge *IJCV* pp 6

[23] Ryzhikov I, Brester I, Semenkin E, Krause J and Satheesh S 2017 Restart operator for multi-objective genetic algorithms: implementation, choice of control parameters and ways of improvement *International Journal on Information Technologies & Security* 4 25–36