Effective Algorithm for Object Detection in the Video Stream for ARM Architectures

Kanstantsin Kurachka, Ihar Nestsiarenia
Sukhoi State Technical University of Gomel
Gomel, Belarus
Kurochka@gstu.by, Ihar.Nestsiarenia@gmail.com

***Abstract*—An algorithm is presented that allows detecting moving objects in a video stream at a fixed camera position. The algorithm is characterized by low resource consumption, what makes it possible to use it in ARM architectures [1] or for data pre-processing on client devices. The algorithm performs the following steps: image scaling, clipping background, and detection of objects. Prototype algorithm implemented in Python [2] using the OpenCV library. It was tested on a single-board computer Raspberry PI 3 [3], showed a performance of 20 FPS with an input stream frame size of 1920x1280 pixels.**

***Keywords*—detection, video processing, classification**

I. Introduction

Currently, neural networks are widely used for digital image processing, but they usually require significant computational resources. Therefore, in practice, it is advisable to combine the using of neural networks with algorithms that do not use training, which will in some cases increase the speed of solving problems and reduce resource consumption [6].

Neural networks show a high degree of classification of a large number of different objects, but the performance is 6-8 FPS [5], [7]. The solution to this problem is to use high-performance hardware with the GPU, it is not always possible and appropriate on grounds of the cost and energy efficiency. One of the modern practical tasks is the real time analysis of streaming video data. For such problems, the accuracy of the algorithm can be neglected. There is no need to ensure the detection of objects at each frame, it is necessary to achieve that at least one of a series of frames of the object is detected.

This kind of task includes detection of cars using surveillance cameras. There are two approaches to solving this problem, the first one is to process completely on the portable device, the second one is to send all video stream to the server for centralized analysis and recognition. The first approach depends on resources, it is not always possible to use high-performance devices on the end users of the system. In the second approach, it is necessary to provide a reliable and secure communication channel with high performance for connecting client devices to the data processing server, which in modern conditions is also costly and sometimes completely unrealizable for a number of subjective reasons. Providing a similar channel for multiple connections is very expensive, and the cost will increase with connecting new devices, which indicates the problems of scaling such a solution.

A possible solution could be the development of algorithms that reduce requirements of throughput and the performance of the computing server by performing preliminary partial processing of the video stream on the end devices. This paper proposes using a combination of transformations and filters to significantly reduce the amount of data transmitted to the processing server. On the resulting images, you can quickly search for objects using cascades [8].

On the client side, cheap single-board computers with ARM architecture [3] are offered as hardware for pre-processing. low power, e.g., Raspberry PI 3 [2].

II. General Algorithm

The camera receives the data that needs to be processed. To reduce video traffic on the client-server, the input video stream is sent to a single-board computer, where it is pre-processed according to a multi-level scheme. Levels are represented by the black box model and are interconnected only by input and output data. This approach allows the use of conveyor parallel processing, thereby ensuring a uniform load of the existing cores and the highest performance [4].

The diagram of processing is presented in Fig. 1 and includes the following steps:

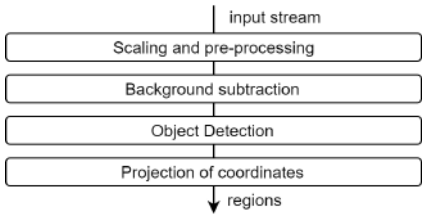

Figure 1. General algorithm diagram

- Scaling – provides camping downsampling as to detect the object is not necessary for high definition.
- Background subtraction – allows you to simplify the task of searching for objects since all objects that do not move for a long time will be removed from the image.
- Object detection in the image – the direct application of search algorithms and the formation of the result in the form of coordinates of rectangular regions.
- Coordinate projection – the coordinates of the regions are converted according to the original image size.

At the first stage, the image size is reduced and it is converted to shades of gray:

$$[M]^{norm} = G([M]) \qquad (1)$$

where M – is the original image, G – is the normalization operator of the input image.

III. Background subtraction

To improve the quality of image detection, it is possible to cut off static unchangeable parts (background), we take advantage of the fact that cameras are placed statically. This gives a significant advantage when searching for objects and their recognition. The previous frames can be used to select the background, the part which is not changing some period of time. This part of the video frame can be removed, what will reduce the amount of data to analyze. Thus, the background subtraction is performed based on the available set of images (2).

$$[M]^{FG} = \| [M]^{norm} - \cap_i [M]_i \| \tag{2}$$

It seems that for detecting moving objects of a given type, there are enough background subtractions algorithms [9], but this is not so, because it is necessary to notice not any moving objects, but only a certain type. For example, for the task of detecting moving cars, the classifier should not react to other road users (pedestrians, cyclists, etc.), and in general to foreign objects that are in the frame.

Before using the subtraction of the background operation, an additional configuration of the scene is required to select and classify sections of the road: curb, pavement, lanes in both directions, etc.

In this case, at the preliminary stage, the boundaries of the analyzed areas of the video stream will be determined. In this case, it is possible to perform parallel processing of each of the areas found using cascades. The use of decomposition of the analyzed scene will reduce the amount of data processed, increase the quality and speed of the search.

Shadows and noise in the image can complicate the work of the background subtraction algorithm. For example, a shadow moving together with objects may significantly reduce the quality of detection or even lead to the impossibility of recognition and classification of an object. Therefore, it is necessary to perform image transformation in such a way that all the "surroundings" of the desired object are removed, even if this operation will result in distortion of the object itself.

Thus, in formula (2), it is required to specify not only the metric used but also the set $[M]_i$. All this is determined depending on the selected background subtraction algorithm. When choosing an algorithm, the main criteria, determined by the need to perform on low-performance devices, were its resource intensity, speed, the accuracy of correctly detected images after background clipping.

The following algorithms were analyzed:

- MOG – based on a Gaussian mixture model [10].
- MOG2 – based on MOG ideas, the main feature is that the algorithm chooses an approximate number of Gaussian distributions for each pixel [11], [12].
- GMG – algorithm combines a statistical evaluation of the original image and pixel-by-pixel Bayesian segmentation [13]. After evaluating the first frames, the algorithm finds the objects separating them from the background.
- KNN – based on the k-means algorithm [14].

As a result of computational experiments, at the preliminary stage, methods based on the k-means algorithm were excluded, since they were less effective according to the specified criteria.

MOG uses a number of Gaussian distributions, also called Gaussian components, to simulate the background of pixels. Each pixel has its own set of Gaussian components. Each component needs three parameters: the average (background intensity), weight (as an indicator of the significance of the component) and the standard deviation of intensity [15], [16].

MOG2 is an improved version of the MOG algorithm, and, due to the possibility of adaptively adjusting the number of distributions for each pixel, is more resistant to changes in lighting on the scene. This is a significant advantage, making it possible to use this algorithm on external observation cameras. For the task of detecting cars is an important condition. The algorithm should work steadily throughout the day and when the weather conditions change.

As a result, formula (2) was converted to the form:

$$[M]_t^{BG}(x^i) = p(x^i|BG) = \sum_{k=1}^{K} \pi_k^i N(\mu_k^i, \sigma I) \tag{3}$$

where K - the number of Gaussian components, each with a weight $\omega_{i,t}$ and intensity $\mu_{i,t}$ and standard deviation $\sigma_{i,t}$, N – normal distribution.

The following parameters for the MOG2 algorithm were calculated empirically: the number of Gaussians is 2, the filtering of shadows, and the complexity reduction threshold that determines the number of examples needed to determine whether an image component exists or not. Such values allow you to deal with various noises in the video, shadows, camera vibration. At the same time, which is very important, the possibility of classifying objects does not deteriorate. For analysis, 10 frames were used with the quality of shooting 20 FPS. This number of frames is enough to detect the movement of objects and allows you to select the background (Fig. 2).

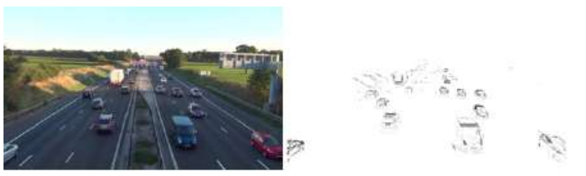

Figure 2. Background subtraction

IV. Object detection using cascades

Algorithms that use the idea of cascades show their effectiveness when searching for one type of objects in an image [17].

The basis of these algorithms are the Haar primitives [8], which represent the discretization of a given rectangular region into sets of heterogeneous rectangular subregions. In the original version of the algorithm, only primitives without turns were used, and to calculate the feature, the difference of the sum of the brightness of one region from the sum of the brightness of the other subregion. When developing the approach, we began to use oblique primitives, instead of calculating the difference in brightness, it was proposed to assign a separate weight for each subdomain and calculate weighted sums of features for different types of areas. Thus, a set of classifiers is compiled, each of which is assigned a weight during training; training is usually performed using the AdaBoost algorithm.

The advantage of the Viola-Jones algorithm is the ability to quickly calculate signs using the integral representation of the image. This means that to calculate the sum of the intensities of pixels in the selected area, you can use 4 values of the sum of intensities s_1, s_2, s_3, s_4 from which you can calculate the value in a given area 4. The construction of the integral representation of the image has complexity $O(n)$ where n is the number of pixels in the image. The amount of light in the given area 5.

$$D = s_1 + s_4 - (s_2 + s_3), \tag{4}$$

$$s_i = \iint_S I(x,y)dxdy \approx \sum_{x=0}^{x_{Si}} \sum_{y=0}^{y_{Si}} I(x,y), \tag{5}$$

where S is the sampling primitive, $I(x,y)$ – pixel intensity with coordinate (x,y) x_{Si} – boundary coordinate x of rectangle Si y_{Si} – boundary coordinate y of rectangle Si $i = \overline{1,4}$.

To determine class membership, a classification function is introduced in each stage. Each "weak" classifier produces two values, depending on whether the value of the attribute belonging to this classifier is greater or less than the specified threshold. At the end, the sum of the values of "weak" classifiers is compared with the cascade threshold and the decision is made whether the object is found or not by this cascade 6. In the process of learning a lot of classifiers are built and weights are selected for each of them. The resulting classifier is used to determine whether the site belongs to the desired object class 7.

$$\Psi(x) = sign[\sum_{i=1}^{k} \omega_i R_i(x)] \tag{6}$$

where is "weak" classifier, ω_i – weight of i-th classifier; k – the number of classifiers.

As a result of background subtraction, a filtered image is obtained, which is divided into regions r using the $E()$ operator:

$$r = E([M]^{FG}),$$

each found region, we introduce a classifier using 4, which returns the probability of finding an object in the region.

$$\Psi(o_i) = 1, if o_i \in r_i \tag{7}$$

where: o_i – the desired object.

The use of only signs of Haar makes it possible to find more than 80% of the objects. Change video angle, scale, etc. the quality of recognition with the use of cascades sharply deteriorates [18]. In practical terms, the number of signs increases nonlinearly, which significantly reduces the speed of the algorithm. In this case, you need to either limit the possible situations, for example, only for a certain camera position (distance, and rotation angle), or add additional video processing to reduce the set of features without presenting special video requirements. The proposed modification of the algorithm is undemanding to the initial configuration.

This algorithm is undemanding to computing resources and can be implemented on almost any modern ARM processor, which allows the proposed concept to be used to build complex recognition systems.

V. Data preparation and train classifier

High-quality data preparation is the most important step in the development of the classifier. The algorithm will work effectively only when there is a lot of data collected that maximally covers all possible variations of the objects being classified. In relation to the task of detecting cars, you need to select photos taken from different angles and from different angles. Also in the training set should get cars with different color shades.

To train the cascade on "raw" images, you need to prepare more than a thousand samples, with an approximately uniform distribution of different combinations of colors, models and angles.

Due to the non-linear dependence of computational complexity and quality of classification on the number of features, combining all the properties of objects in one classifier leads to an increase in resource intensity and a drop in the recognition rate of the desired object. Therefore, by applying background subtraction as a side effect, very simplified car images are obtained, which makes it possible to neglect the different color shades of the original images.

Thus, it is proposed to introduce an additional step in preparing the training data: the source video stream is processed by the background clipping algorithm. And to achieve maximum accuracy of the classifier, you need to use the same configurations for background clipping, which will be used in a working system. An example of an image before and after applying a filter is shown in Fig. 3.

To train a classifier, it is necessary to assemble a set of negative examples - background images, and objects that should not be detected.

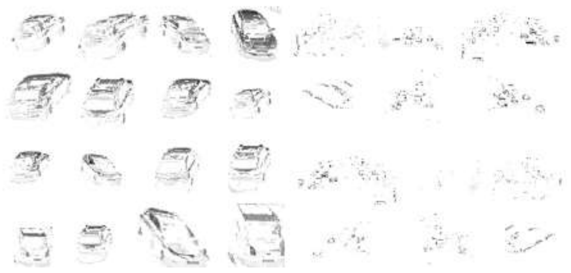
Figure 3. Examples of images for training, on the left are positive examples, on the right are negative

For classifier operation, it is recommended to use at least 500 positive examples of images, 2-4 times more negative ones. It is also recommended to use augmentation, with reflection horizontally, scaling, and turning in the limit of 15 degrees.

VI. Evaluation

The article presents an algorithm that is not demanding of computational resources, characterized in that it is a combination of image processing and detection algorithms, which allows video to be processed in real time. The first is the clipping of the background based on the previous frames, then - learning the cascade on the resulting images. After clipping the background image, a lot of noise is produced, but the vehicles have expressive boundaries, resulting in a simple set of features.

The combination of algorithms allows to achieve high precision detection of passenger cars (about 95%) and process up to 20 frames per second with a resolution of 1920x1280. An example of detection is shown in Fig. 4.

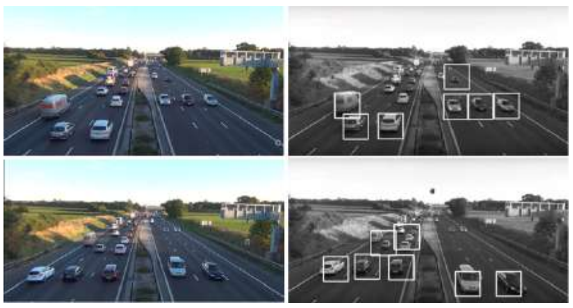
Figure 4. Detection Results

The proposed algorithm can be used to build traffic monitoring systems, as well as subsequent statistics collection, which will provide the most relevant information about the state of a city's roads at any time . The introduction of monitoring systems makes it possible to apply various mathematical models to optimize traffic, because automatic systems for collecting statistics from highways will allow you to accumulate a sufficient amount of information that can be used in conjunction with machine learning algorithms in order to improve throughput and safety of highways. Also, the collected information will be very valuable when making changes in traffic (for example, closing the street, or expanding it), the information collected will always be relevant, which means modeling changes in traffic will provide the greatest efficiency.

The resulting algorithm can be used to build statistics collection systems with preliminary data processing on the device. Low algorithm requirements make it possible to use low-power devices.

References

[1] Seal David, ARM architecture reference manual, Pearson Education, 2001

[2] Python. Available at: https://www.python.org/ (accessed 2018, Dec)

[3] Raspberry Pi 3 model b. Available at: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ (accessed 2018, Dec)

[4] Kurochka K. S., Safonau I. V. Medical Images Indexing and Retrieval in a Distributed Computing Environment //Journal of Automation and Information Sciences, 2010, vol. 42, No 5

[5] Kurachka K. S., Tsalka I. Vertebrae detection in X-ray images based on deep convolutional neural networks //Informatics, 2017 IEEE 14th International Scientific Conference on IEEE, 2017, pp. 194-196.

[6] Xun Wang, Jie Sun, and Haoyu Peng. Foreground object detecting algorithm based on mixtureof gaussian and kalman filter in video surveillance.JCP, 8(3):693–700, 2013.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-timeobject detection with region proposal networks. In Advances in neural information processingsystems, pages 91–99, 2015.

[8] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features.In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEEComputer Society Conference on, volume 1, pages I–I. IEEE, 2001.

[9] Temirlan Z. Kamilla A. Vehicle detecting, tracking and counting for traffic control systems.icp2015, 2015.

[10] Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixturemodel for real-time tracking with shadow detection. Video-based surveillance systems, 1:135–144,2002.

[11] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. InPattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on,volume 2, pages 28–31. IEEE, 2004.

[12] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per imagepixel for the task of background subtraction.Pattern recognition letters, 27(7):773–780, 2006.

[13] Andrew B Godbehere, Akihiro Matsukawa, and Ken Goldberg. Visual tracking of humanvisitors under variable-lighting conditions for a responsive audio art installation. In AmericanControl Conference (ACC), 2012, pages 4305–4312. IEEE, 2012.

[14] Darren Butler, Sridha Sridharan, and VM Jr Bove. Real-time adaptive background segmentation.In Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEEInternational Conference on, volume 3, pages III–349. IEEE, 2003.

[15] Hamed Tabkhi, Robert Bushey, and Gunar Schirner. Algorithm and architecture co-design ofmixture of gaussian (mog) background subtraction for embedded vision. In Signals, Systemsand Computers, 2013 Asilomar Conference on, pages 1815–1820. IEEE, 2013.

[16] Nils Stahl, Niklas Bergstr om, and Masatoshi Ishikawa. Exploiting high-speed sequences forbackground subtraction. In Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on,pages 106–110. IEEE, 2015.

[17] Paul Viola and Michael J Jones. Robust real-time face detection. International journal ofcomputer vision, 57(2):137–154, 2004.

[18] M Oliveira and V Santos. Automatic detection of cars in real roads using haar-like features.Department of Mechanical Engineering, University ofAveiro, 3810, 2008

ЭФФЕКТИВНЫЙ АЛГОРИТМ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ НА ВИДЕОПОТОКЕ АДАПТИРОВАННЫЙ ДЛЯ ARM АРХИТЕКТУРЫ

Курочка К.С., Нестереня И.Г.

Представлен алгоритм, позволяющий детектировать движущиеся объекты в видеопотоке при фиксированном положении камеры и отличающийся низкой ресурсоёмкостью, что позволяет его использовать в ARM архитектурах [1] или для предварительной обработки данных на конечных (клиентских) устройствах. Алгоритм реализует следующие этапы: масштабирования изображения, отсечения фона, и детектирования объектов. Тестирование алгоритма, реализованного на языке Python [2] и с использованием библиотеки OpenCV, на одноплатном компьютере Raspberry PI 3 [3] показало производительность – 20 FPS при размере кадра входного потока 1920x1280 точек.

Received 29.12.18