Design and Implementation of Artificial Intelligence Image Detection System Based on Noise Monitoring

Qiong Ren, Juming Chang *
College of Artificial Intelligence, Jianghan University, Wuhan 430056, China

*Corresponding author email: qiongren@jhun.edu.cn

Abstract. Image quality detection technology can replace manual inspections to automatically detect video quality, accurately analyze, judge and alarm abnormalities in video images in the surveillance system to ensure the normal operation of the expanding network video surveillance system. The thesis designs an intelligent traffic image sensor system to realize fast motion detection and object recognition of the surveillance scene. This system introduces a background model based on region segmentation and a feature-based object recognition algorithm. Experimental results show that the system can perform automatic movement detection and object classification and recognition efficiently in real time.

Keywords: Image noise; artificial intelligence; image detection system; image processing.

1. Introduction
In recent years, with the rapid development of multimedia technology, network technology and handheld intelligent systems, as well as the continuous improvement of people's living standards, the widespread use of intelligent anti-theft monitoring systems has become a development trend. As an indispensable part of modern enterprises, monitoring systems have been widely used in many fields such as transportation, hospitals, banks, home furnishings, education, etc., which can effectively avoid potential safety hazards and improve work efficiency [1]. Although the traditional monitoring system can obtain the image information of the monitored area in real time, it cannot automatically identify suspicious targets from the image or video information, does not have the real-time alarm capability, and cannot realize unattended intelligent monitoring. This article introduces an embedded intelligent anti-theft monitoring system based on image processing technology. The system realizes real-time monitoring of scenes, automatically extracts suspicious stationary or moving targets, and intelligent alarms.

2. System Design

2.1. Hardware system design
The hardware system includes the camera input part, the embedded processor Au1200 part for network transaction interaction, and the BF533 part for collecting/processing video images. Fast data exchange between Au1200 and BF533 through DPRAM. The system hardware block diagram is shown as in Fig. 1.
2.2. Software system design

The system software is based on the embedded Linux operating system and is developed using QtEmbedded. Qt is a cross-platform C++ graphical user interface library. Its good packaging mechanism makes it highly modular and reusable, which is very convenient for user development. As shown in Figure 2, the software system is mainly composed of an image acquisition and storage module, a target extraction module and an intelligent alarm module [2]. The system software collects scene image data in real time through the image acquisition and storage module, and then sends the image data to the target extraction module for real-time processing. The target extraction module uses image processing algorithms to automatically detect the collected images. If a suspicious target is extracted from the image, it will immediately send an alarm signal in the form of a short message through the smart alarm module to notify the monitoring personnel to deal with it in time, and store the suspicious image data.

1) Image data acquisition

The ARM platform controls the USB camera to collect scene images at a rate of 1 frame/s, and then send them to the image processing and target detection module for processing. To control the camera, you need to write a camera driver, and the application layer program calls the camera driver to complete the operation of the camera. The driver of the USB camera is a character device driver. The application opens a device file with the major device number through the open("/dev/video0", O_RDWR) function to establish communication with the device driver, and then through ioctl() The function configures the opened device, the Qcam module completes the image capture, and closes the device using the close() function.

2) Image target detection algorithm

The target detection algorithm in the image mainly includes time domain difference method, background difference method, optical flow analysis technology and so on. The image processing and target detection module of this system uses a moving target detection algorithm based on background subtraction, which can effectively detect static or moving abnormal targets. Figure 3 shows the image processing flow for detecting stationary and moving targets using this algorithm. The overall
computational complexity of the algorithm is small, and the complexity is low, which is convenient for real-time processing of image data.

Figure 3. Image processing flow for target algorithm to detect stationary and moving targets

At present, there are many moving target detection algorithms commonly used at home and abroad, mainly including optical flow method, background difference method, inter-frame difference method and statistical method [3]. Due to the influence of various noises in the image, these algorithms have their own differences in effectiveness, real-time processing and anti-interference. The following is an introduction to the optical flow method. If the gray value of pixel \((x, y)\) at time \(t\) is \(f(x, y, t)\), after \(dt\) time, the pixel moves to \((x + dx, y + dy)\), assuming that the brightness value of the pixel during the displacement process satisfies

\[
f(x + dx, y + dy, t + dt) = f(x, y, t) \quad (1)
\]

According to the conservation of gray level, the optical flow equation can be obtained:

\[
\frac{\partial f}{\partial x} u + \frac{\partial f}{\partial y} v + \frac{\partial f}{\partial t} = 0 \quad \text{uf}_x + v f_y + f_t = 0 \quad (2)
\]

Where \(u, v\) represents optical flow, and \(u = \frac{\partial x}{\partial t}, v = \frac{\partial y}{\partial t}\). According to the optical flow equation of formula (2), the optical flow \(u, v\) cannot be solved uniquely, and additional constraints are required. The most commonly used constraint is the hypothesis of optical flow smoothness, that is, it is assumed that the optical flow changes smoothly over the entire image. After the optical flow smoothing constraint is added, the calculation of optical flow is transformed into a minimization problem, namely:

\[
e_{\min} = e_s + \lambda e_s \quad (3)
\]

Among them, \(\lambda\) is the Lagrange multiplier, and its size is determined by the noise of the image. The greater the noise, the greater the value, and vice versa, the smaller. \(e_s\) represents the square value of the gradient amplitude of the optical flow, namely:
\[ e_s = \iint (||\nabla u||^2 + ||\nabla v||^2) \]
\[ = \iint (\frac{\partial u}{\partial x}^2 + (\frac{\partial u}{\partial y})^2 + (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2) \]  
\[ e_s = uf_x + vf_y + f_t \]  
(4)

Since the minimum value of \( e_{\text{min}} \) is taken, it can be derived:
\[ \frac{\partial e_{\text{min}}}{\partial u} = 0 \]
\[ \frac{\partial e_{\text{min}}}{\partial v} = 0 \]  
(6)

We take \( \lambda = \partial^2 \) so that the iterative equation for solving optical flow can be derived as:
\[ u^{k+1} = \bar{u}^k - f \frac{f_x \bar{u}^k + f_y \bar{v}^k + f_t}{\partial^2 + f_x^2 + f_y^2} \]
\[ v^{k+1} = \bar{v}^k - f \frac{f_x \bar{u}^k + f_y \bar{v}^k + f_t}{\partial^2 + f_x^2 + f_y^2} \]  
(7)

Among them, \( \bar{u}, \bar{v} \) is the neighborhood average of \( u, v \) at point \((x, y)\), and \( k \) is the number of iterations. Generally, the initial value of the iteration is set to zero. When the result of two adjacent iterations is less than the preset threshold \( \varepsilon \), the iteration ends [4]. The optical flow detection method is suitable for static or dynamic scenes and camera movement, and can better detect moving targets. However, its calculations are complicated, and it cannot meet the real-time requirements without special hardware, and the calculation of the optical flow field is susceptible to shadows, noises and various occlusion situations, resulting in inaccurate results. Therefore, the optical flow method is mostly suitable for situations where the target motion rate and image noise are both small.

3) Image denoising algorithm
a) Pre-selected pixel acceleration algorithm

Analyzing the complexity of its algorithm, we know that the time complexity of the non-local mean denoising algorithm is quite large. We assume that the size of the similarity window in the algorithm is \((2f+1)^2\), and we confine the search for the similarity window to a "search window" of size \((2s+1)^2\). Suppose the original image size is \(N^2\), then the total complexity of the algorithm is \(O(N^2 \times (2f+1)^2 \times (2s+1)^2)\). We take \(f = 3, s = 10\) in many experiments, so the total time complexity is \(O(49 \times 441 \times N^2)\). For an image, it takes about 30 seconds to run on an ordinary PC, if you use MATLAB programming, it may be slower. Aiming at the shortcoming of the huge time complexity of the non-local mean algorithm, we propose an improvement to the accelerated processing of preselected pixels.

In our actual calculations, we can just find those \( j \) points with the largest \( w(i, j) \), and there is no need to calculate the Euclidean distance between the searched pixel points and the neighborhood of the point \( i \) to be denoised. Therefore, we need to have some prior knowledge to eliminate those \( j \) points whose \( w(i, j) \) is expected to be calculated to be smaller, so as to achieve an acceleration effect. The design of this preprocessing is based on the following assumption: From the formula
we know that the value of \( w(i, j) \) is determined by the distance between the neighborhood of point \( i \) and the neighborhood of point \( B \), that is, the degree of similarity between them. In order to accelerate, we have to take a larger \( w(i, j) \) instead of calculating the smaller \( w(i, j) \) values. so, we can get the similar \( i \) point neighborhood and \( j \) point neighborhood in advance. To characterize the degree of similarity, we can take an approximation: first calculate the local mean and local standard deviation of the elements in the similar window around each point of the entire image and record it as a similarity index for that point, which is recorded as \( (I(N_i), \text{Var}(I(N_i))) \). In order to remove the influence of noise, we approximate the points where \( \mu_i \) and \( \mu_j \) are very close to the point that is most similar to the point \( i \) to be denoised, and thus just calculate the weight between them. This can be represented by the following formula:

\[
\begin{align*}
 w(i, j) &= \begin{cases}
 \frac{1}{Z(i)} e^{-\frac{||v(N_i)-v(N_j)||^2}{\sigma^2}}, & \text{if } \mu_i < \frac{I(N_i)}{I(N_j)} < \mu_2 \\
 0, & \text{otherwise}
\end{cases}
\end{align*}
\tag{8}
\]

b) Adaptive selection of optimal denoising parameters

The parameter \( h \) in the weight function \( w(i, j) = \frac{1}{Z(i)} e^{-\frac{||v(N_i)-v(N_j)||^2}{\sigma^2}} \) controls the smoothness of the noise. When \( h \) is relatively large, \( w(i, j) \) is relatively large, so the final weighted average effect will make the denoising part smoother. And if \( h \) is relatively small, the attenuation effect of the power function is more significant, so in the respective search area, the degree of detail retention is relatively high, so the detail information of the image itself will be maintained. Therefore, how to adaptively select an optimal parameter \( h \) according to the characteristics of each area to be denoised is critical to the denoising effect of the final non-local mean algorithm. In the works of Buade they took the value of \( h \) between \( 10\sigma \) and \( 15\sigma \), but each image corresponds to a different optimal \( h \) value. Let’s examine the images in detail to establish an adaptive Parameter selection algorithm to find the optimal parameters.

First of all, we know the value of \( h \) and the size of the neighborhood window \( |N_i| \). The original noise variance \( \sigma^2 \) of the image satisfies a certain relationship. We can assume that the relationship satisfies \( h^2 = f(\sigma^2, |N_i|, \beta) \), where \( \beta \) is a constant. Intuitively speaking, as our search similarity window \( |N_i| \) increases, the corresponding \( \| v(N_i) - v(N_j) \| \) will inevitably increase. Therefore, the value of \( h \) should be increased accordingly. Similarly, when \( h \) is larger and \( w(i, j) \) is larger, the smoothing effect is better. And if \( h \) is relatively small, the degree of detail retention is relatively high, so the detail information of the image itself will be maintained. So if the noise variance \( \sigma^2 \) of the original image is relatively large, we need a relatively large \( h \) to smooth the noise; if the noise variance \( \sigma^2 \) of the original image is relatively small, we need a relatively small \( h \) to preserve the details. So intuitively, \( h \) and \( |N_i| \) and \( \sigma^2 \) satisfy the same increase and decrease relationship, and we construct its functional relationship as \( h^2 = 2\beta\sigma^2 |N_i| \) . It can be regarded as a manual control parameter. We will examine the relationship between its value and the denoising effect in detail in the section of experimental analysis. For any image that knows the noise variance \( \sigma^2 \), we only need to substitute the above function to obtain the denoising parameter \( h \). But because generally speaking, for
an input image, we don’t know its noise variance in many cases. so, we make an estimate to examine the original noise variance \( \sigma^2 \) calculation of the image.

We assume that the noise of the image is Gaussian additive white noise. We use an approximation algorithm to approximate the noise variance \( \sigma^2 \) of the image by estimating the pseudo residual of each pixel. The pseudo residual of each pixel \( i \) can be obtained by the following formula:

\[
\varepsilon_i = \frac{4}{5} \left( I(i) - \frac{1}{4} \sum_{j \in P_i} I(j) \right),
\]

where \( P_i \) is the four-neighborhood of pixel \( i \). The constant \( \frac{4}{5} \) is to ensure \( E[\varepsilon_i^2] = \sigma^2 \) in an isotropic uniform region. Therefore, the noise \( \sigma \) can be estimated by the following formula:

\[
\sigma^2 = \frac{1}{|I|} \sum_{i} \varepsilon_i^2,
\]

where \( |I| \) is the size of the entire image.

2.3. Network transmission scheme

Since the video data requires high real-time performance, the video data transmission of this system selects the RTP protocol running on top of the UDP protocol to complete. The real-time transmission protocol RTP is a transmission protocol specially designed for the real-time transmission of multimedia data. It has the supporting RTCP transmission control protocol to provide control and QoS services. After the server receives the video request sent by the client, it encodes the video frame in X264, and sends the X264 code stream into the RTP library, which multiplexes it into RTP packets, and each packet is assigned a continuous number. These RTP packets will be filled into a socket and packaged into UDP packets, embedded in IP packets and sent to the client in a multicast manner [5]. After sending RTP data packets, RTCP data packets are periodically sent between the server and the client to provide QoS services to meet the transmission requirements of the video stream.

When the client receives the RTP packet, it determines whether there is a packet loss during the transmission process according to its consecutive number. If there is a loss, an interpolation method is adopted to approximate the lost intermediate value. After receiving the RTP data packet, remove the RTP packet header to restore the video data, and then perform storage, decoding, motion detection identification, real-time playback and playback.

3. Experiment and analysis

3.1. Background extraction

Here, \( N=100 \) video image frames are taken, and after averaging processing, the restored road background is shown in Figure 4, which basically filters out pedestrians and moving vehicles. Then, the original image frame sequence and the background image are processed separately to obtain the contour image frame of the moving target, as shown in Figure 5.

![Figure 4. Extracting the background image](image-url)
Figure 5. Contour image frame of moving target

3.2. Image segmentation

From Figure 5, you can see the contour of the moving target, but the segmentation effect is too fine to obtain an effective overall contour identification, and includes the contours of some non-vehicle targets. Then, an image segmentation algorithm based on gray-scale mathematical morphology is used to sum and extract the contour details of the same target object into an overall contour. Figure 6 shows the processed image [6]. Compared with Figure 5, it can be seen that the segmentation process combined with expansion and erosion can effectively obtain the approximate overall contour boundary of the moving vehicle, and filter out the noise interference caused by a part of the moving object with a smaller area.

Figure 6. Image segmentation effect based on gray-scale mathematical morphology

In the detection, the optimal result of the Kalman filter output of the current frame is derived iteratively from the coordinate information of the previous frame, and the coordinates of the same moving target in the previous and subsequent frames have continuity [7]. When the estimated value and the measured value are far from the optimal result, it means that the noise interference caused by the non-effective moving target occurs during the image segmentation, and the index image area corresponding to the non-effective moving target should not be segmented. Therefore, according to the degree of convergence on the graph, extract the index image area with a smaller difference between the estimated value \( x \) and the measured value \( z \) and the optimal output result \( s \) and keep it, and restore the boundary of the target object. As shown in Figure 7. It can be seen from the experimental results that when the noise interference at the jumping position appears in the video image, it can be effectively shielded and filtered.
In a scene with multiple artificial light sources interference, when a moving target appears, the amount of calculation of the compression algorithm and the motion detection algorithm will increase, and the system can still stabilize the frame rate at 25f/s while maintaining the image quality. It can be seen from Figure 7 that the identified detection area has a high degree of matching with the actual moving object. By combining the Kalman dynamic background update and image difference motion detection algorithm, the detection can be enhanced without increasing the false alarm rate. Sensitivity and accuracy.

4. Conclusion

This article proposes a fast and accurate video quality detection scheme for the real-time monitoring requirements of the network video surveillance system. Starting from the image space domain, this paper analyzes the shape and distribution characteristics of noise abnormal pixels in the neighborhood, combined with OpenCV image processing technology, proposes a spatial domain-based image noise detection algorithm, and realizes the detection of image noise, snowflakes and Stripe abnormality detection. When detecting abnormal image noise, according to the characteristics of different types of noise, the normal edge information of the image is fully considered to avoid the false detection of edge pixels and make the detection more accurate. Through experimental tests, the image noise spatial domain detection idea proposed in this paper has good results in most cases when detecting image noise interference. In order to further improve the detection accuracy, it is still necessary to continuously optimize these three detection algorithms, and combine the detection of signal loss, blur, color cast, jitter, distortion and other abnormalities to achieve complete detection of video quality.

Acknowledgments

This paper is sponsored by Scientific Research Program Guiding Research Project of Hubei Provincial Department of Education, (Project No. 2020GA031).

References

[1] Duan, P. Kang, X. Li, S. & Ghamisi, P. Noise-robust hyperspectral image classification via multiscale total variation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, pp. 1948-1962, June 2019.
[2] Benz, D. C. Benetos, G. Rampidis, G. Von Felten, E. Bakula, A. Sustar, A. ... & Buechel, R. R. Validation of deep-learning image reconstruction for coronary computed tomography angiography: impact on noise, image quality and diagnostic accuracy. Journal of cardiovascular computed tomography, vol. 14, pp.444-451 May 2020.
[3] Ralević, N. M. Karaklić, D. & Pištinjat, N. Fuzzy metric and its applications in removing the image noise. Soft Computing, vol. 23, pp. 12049-12061, Twenty-two.
[4] Fan, X. Dai, M. Liu, C. Wu, F. Yan, X. Feng, Y. ... & Su, B. Effect of image noise on the classification of skin lesions using deep convolutional neural networks. Tsinghua Science and
[5] Zheng, Y. B. Huang, T. Z. Zhao, X. L. Jiang, T. X. Ma, T. H. & Ji, T. Y. Mixed noise removal in hyperspectral image via low-fibered-rank regularization. IEEE Transactions on Geoscience and Remote Sensing, vol. 58, pp. 734-749, January 2019.

[6] Schofield, R. King, L. Tayal, U. Castellano, I. Stirrup, J. Pontana, F. ... & Nicol, E. Image reconstruction: Part 1–understanding filtered back projection, noise and image acquisition. Journal of cardiovascular computed tomography, vol. 14, pp. 219-225, March 2020.

[7] Karthik, B. Kumar, T. K. Vijayaragavan, S. P. & Sriram, M. Removal of high density salt and pepper noise in color image through modified cascaded filter. Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 3901-3908, March 2021.