Journal of Science and Engineering Research
Vol. 1, No. 1, 2022, pp. 47-62
eISSN: 2957-7187

Target Detection based on Edge Computing

Mohamed Osama

Southern Cross University, Bilinga, Australia
mohamed.osama@scu.edu.au

DOI: http://dx.doi.org/10.56828/jser.2022.1.1.4
Article history: *Received (October 7, 2021); Review Result (November 10, 2021); Accepted (January 10, 2022)*

Abstract: Edge computing can effectively reduce the cloud load pressure and improve the real-time response of the system by performing real-time analysis and operation on the front-end data collected at the edge of the network. At the same time, with the rapid development of deep learning technology, especially target detection algorithms, video surveillance is more efficient and intelligent, and the defects of artificial video surveillance methods are effectively alleviated. Therefore, this paper proposes an architecture scheme for a video surveillance system oriented to edge computing, using the target detection algorithm based on a deep convolution neural network to capture abnormal targets efficiently, and the feasibility of the scheme is verified by monitoring data of coal bed methane wells and stations. Aiming at the problems of large parameters of feature extraction network model and high memory consumption of current target detection algorithms, a lightweight deep convolution network MMGNet for edge computing is proposed. The network can effectively reduce the calculation consumption and storage requirements of the model through the deeply separable convolution mode, and further realize the model lightweight by combining Ghost Net. The proposed network maintains the form of "first expansion, then compression" in structure, and introduces the channel attention mechanism to improve the model accuracy. In addition, the proposed network also adds a multi-scale convolution kernel to extract feature information, simulating people's view of object characteristics from different perspectives, further improving the detection accuracy, and achieving 71.52% Top-1 accuracy on the CIFAR100 dataset. Finally, aiming at the abnormal target monitoring method in video surveillance, the target detection algorithm based on MMGNet is adopted to capture the abnormal target quickly and accurately.

Keywords: Edge calculation, NVIDIA TX2, Deep convolutional neural network, MMGNet, Target detection algorithm

1. Introduction

In the field of modern security protection, the industry generally uses artificial video surveillance mode to detect abnormal signals in specific areas and then combines with the cloud computing paradigm [1] to realize all-around and systematic security protection. This model depends on the subjective judgment of the monitoring personnel. If the monitoring personnel have low professional and technical ability, it is easy for the abnormal signal to be missed and falsely detected. Nowadays, in the era of the Internet of Everything, the number of video monitoring nodes is increasing exponentially, and the data of video streams collected and transmitted is surging, which will inevitably cause a serious burden on the data storage, transmission, and network bandwidth resources of traditional cloud computing paradigm. For the application scenarios with low latency requirements such as video surveillance, the traditional cloud computing paradigm will not be able to meet the existing production and living needs. Therefore, researchers put forward edge computing [2], which realizes data collection, analysis, and processing at the data source by performing computing at the edge of the network, which not only ensures the real-time processing requirements of the system but also greatly optimizes the utilization rate of network resources. At the same time, the rapid development of technologies such as chips and embedded devices provides a strong hardware foundation for edge computing. Therefore, the video surveillance system based on edge computing architecture has important theoretical research and practical application value [3]. The first problem in a video surveillance system is whether abnormal targets can be accurately and efficiently captured. With the rapid development of computer vision technology [4][5], applying an object detection algorithm to abnormal signal capture can effectively alleviate the interference of human factors and improve the quality of video surveillance. The deep convolutional neural network has become one of the important methods to solve the problems related to targeting detection of its efficient feature extraction ability. However, due to the redundancy of parameters of the deep convolutional neural network, the model file is generally large, and the computing power and storage requirements of hardware devices are high, which makes the target detection algorithm based on the convolutional network run on the edge embedded platform with great challenges. Therefore, how to design an efficient and lightweight deep convolution neural network for feature information extraction is a hot and difficult research topic nowadays. In this paper, coalbed methane wells and stations are selected as research and application scenarios. Coalbed Methane (CBM) is an important combustible clean energy, and as a working platform for exploiting CBM, CBM wells and stations are mostly distributed in remote suburbs and unattended. Therefore, to ensure the safe exploitation of coalbed methane, it is necessary to realize video monitoring of coalbed methane wells and stations. Traditional monitoring methods mostly use artificial video monitoring based on cloud computing, which requires a large number of people to monitor the video images and relies on the subjective judgment of the monitors, which easily leads to false detection and missed detection of abnormal targets due to human negligence. At the same time, there are thousands of cameras involved in the monitoring system of coalbed methane wells and stations, which generates a large amount of data flow, which will cause huge consumption of network transmission load and cloud center storage calculation, and easily lead to system delay. Therefore, aiming at such scenes as coalbed methane wells and stations, it has important research value and broad application prospect to construct an intelligent video monitoring system based on edge computing to realize real-time accurate capture of abnormal targets.

2. Related Work

2.1. The concept of edge computing

To realize large-scale and intelligent management and application, the Internet of Things puts forward higher requirements for data collection and intelligent processing. At present, cloud computing has the advantages of large scale, stable and reliable service, and low cost, which can meet the development needs of the Internet of Things within a certain range. By using its large-scale computing cluster and high storage capacity, cloud computing can effectively promote the transmission and calculation of sensor data in the Internet of Things and provide intelligent services for users. With the increase of IoT devices, the big data generated by IoT edge devices can not be processed in time and effectively, which leads to the inability of IoT to provide timely services to users. The Internet of Things architecture with cloud computing as the core of data processing needs to transmit a large amount of structured and unstructured raw data collected by sensors from the local embedded devices of the Internet of Things to the cloud computing server, which leads to excessive pressure on the network with limited bandwidth and increases the transmission delay of data in the network. Untreated original data packets containing the user's privacy are uploaded directly to the cloud server, which will greatly increase the risk of user privacy leakage. Edge computing is a computing model or service as opposed to cloud computing. Cloud computing emphasizes centralized processing of data by relying on abundant computing resources of servers, while the purpose of edge computing is to locally process data generated by embedded devices of the Internet of Things in the Internet of Things. Edge computing refers to any computing and network resources between the data source and the path of the cloud computing center. The actual deployment of edge computing is characterized by natural distribution. This requires edge computing to support distributed computing and storage, realize dynamic scheduling and unified management of distributed resources, support distributed intelligence, and have distributed security capabilities. Edge computing products need to consider the integration and optimization of software and hardware to adapt to various conditions and constraints in the Internet of Things environment and support the digital diversity scene of the industry.

2.2. Application scenario of edge computing

(1) Smart home

With the development of Internet of Things technology, the smart home system has been further developed. At present, the smart home mainly controls the smart devices in the home by connecting to the cloud, and the interaction between devices in many home LANs is also realized by cloud computing. However, over-reliance on the cloud platform will also bring many problems. For example, once there is a network failure at home, it is difficult to control the equipment. In addition, controlling devices in the home through the cloud platform sometimes has a slow response speed, which will bring a strong sense of delay, and this bad experience will become more and more frequent with the increase of smart home categories. To reduce the network transmission load and improve the real-time service, the edge computing model is an ideal platform for building a smart home system. For smart homes, the security and privacy of access networks are also valued by people. Edge computing can establish an encrypted channel between the Internet of Things gateway and the data center, further improving the security and privacy of the system.

(2) Intelligent Transportation

At present, intelligent transportation is based on cloud computing to provide intelligent services. In the transportation system, sensors collect traffic data in real-time and upload them to the cloud, and traffic control systems such as traffic lights are coordinated in the cloud. With the increasing complexity of transportation systems, massive real-time data needs to be processed, so the original cloud computing method can not guarantee the timeliness of vehicle-road cooperation. With the development of unmanned driving technology, in the driving process, according to the traditional mode of uploading road condition data and vehicle equipment data to the cloud-first, then analyzing, processing, and returning to the equipment, the signal transmission will be delayed, and traffic accidents will easily occur in emergencies. The edge calculation can reduce the delay and accident probability by processing at the edge side.

(3) Cloud edge collaboration

Edge computing nodes can be responsible for data calculation and storage within their scope, to share the pressure of central cloud nodes. In practice, most of the data still needs to be saved for further processing even after processing and analysis. The data processed by the edge of the Internet of Things still need to be transmitted to the remote cloud server, and the powerful computing power of cloud computing is used to do big data analysis and mining. At the same time, these data also need to be backed up. When there is an accident in the process of edge calculation, the data stored in the cloud will not be lost. Comparing cloud computing with the brain of crop networking data analysis and processing, edge computing is equivalent to the spinal cord of the Internet of Things, which can process data independently under certain circumstances. Only when cloud computing and edge computing work together can the Internet of Things better provide intelligent services for users.

2.3. Edge computing framework

At present, the cloud computing research and related technologies can't efficiently handle the massive data generated by these edge devices and deal with new application scenarios. Therefore, given these problems, academic and industrial circles began to design new computing models and research, and many solutions and measures appeared, such as Cloudlet, mobile edge computing [3], fog computing [4], and micro data center [5] and so on. Facing the big data processing demand of the Internet of Things, the mode of sinking the computing tasks in the remote cloud server to the near sensors or embedded devices came into being, that is, the edge computing model. Edge computing adds the function of big data analysis and processing to the Internet of Things edge device, and migrates some or all computing tasks of the original cloud computing model to the Internet of Things edge device, which ensures the privacy and security of users, saves the time delay required for data transmission, and efficiently enables the Internet of Things to provide timely and intelligent application services for users. The reference framework of edge computing is the focus of many organizations. It embodies the abstract general framework of edge computing and provides the implementation paradigm of the edge computing framework. The open-source platform of edge computing for the Internet of Things is dedicated to solving the problems existing in the process of developing and deploying Internet of Things applications, such as the diversity of device access modes. These platforms are deployed in edge devices such as gateways, routers, and switches to provide support for Internet of Things edge computing applications. The representative platforms are Edge X Foundry released by Linux Foundation and Apache Edgent of Apache Software Foundation. Edge Foundry is a standardized interoperability framework for the development of industrial Internet of Things edge computing, which is deployed on edge devices such as routers and switches, provides plug-and-play functions for various sensors, devices, or other Internet of Things devices, and manages them, and then collects and analyzes their data, or exports them to edge computing applications or cloud computing centers for further processing.

The problem that EdgeFoundry aims at is the interoperability of IoT devices. At present, the Internet of Things with a large number of devices produces a large amount of data, and it is urgent to combine the application of edge computing. However, the diversity of hardware, software, and access methods of the Internet of Things brings difficulties to the data access function and affects the deployment of edge computing applications. The main purpose of

EdgeFoundry is to simplify and standardize the architecture of industrial Internet of Things edge computing and create an ecosystem around interoperable components. Apache NT is an open-source programming model and microkernel-style runtime, which can be embedded into edge devices to provide local real-time analysis of continuous data streams. Edenton solves the problem of how to efficiently analyze and process the data from Edgent devices. To speed up the development process of edge computing applications in data analysis and processing, Edgent provides a development model and a set of APIs to realize the whole data analysis and processing process. Container technology, Docker, as an open-source application container engine, is used by many edge computing platforms to provide flexible application deployment methods. Kubernetes is an open-source project that automatically deploys, extends, and manages containerized applications, and can be used in edge computing platforms to provide reliable and extensible container layouts.

With the rapid development of embedded devices, especially the emergence of GPU, TPU, and NPU, the computing power of edge devices has made rapid progress, which can quickly process and analyze the collected data and provide a hardware foundation for the development of edge computing. NVIDIA TX2, an excellent edge computing device, has been used in many fields [6]. TX2 has very powerful AI performance, equipped with 4GB PASCAL architecture GPU and 256 CUDA processing cores, and equipped with various interfaces for personalized function development. The specific parameters and specifications of the equipment are shown in Table 1. In this paper, TX2 is selected as the edge device of the video surveillance system architecture scheme.

Table 1: Technical specifications of NVIDIA TX2

Hardware name	Parameter specification
GPU	NVIDIA PascalTM, 256 CUDA cores
CPU	HMP Dual 2/2MB L2+Quad ARM@A57/2MB L2
internal storage	8 GB 128-bit LPDDR4
Data storage	32GB Emmc、SDIO、SATA
video clip	4K×2K 60Hz coding (HEVC),
	4K×2K 60Hz decoding (12 bits)
Monitor	Two DSI interfaces, two DP1.2 interfaces /HDMI 2.0 interfaces /e DP
	1.4 interface
PCIE	Gen | 1× 4+1× 1 or 1×2+2×1
USB	USB 3.0+USB2.0
connect	1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth
other	CAN、URAT、SPI、I2C、I2S、GPIO

2.4. Deep convolution neural network

A Deep Neural Network (DNN) is an improved development based on the traditional artificial neural network. It refers to the machine learning process of training a neural network model based on the training set sample data to obtain the network structure with optimized parameters and multiple hidden layers of neurons [7][8]. The reason why deep learning is called "depth" is relative to the "shallow learning" method of traditional data mining classification algorithms. The deep neural network has a multi-layer structure. Compared with the traditional classification algorithm, multiple hidden layers of the deep neural network can automatically learn the features of data, reducing the workload of artificial feature engineering. A deep neural network even abstracts high-dimensional data features to improve the accuracy of data classification. Among many deep neural network models, Convolutional Neural Network (CNN) is the most widely studied and applied. A convolutional feed forward neural network is a typical feed forward neural network, and its main function is that it can be used for data classification and prediction [9]. A convolutional neural network uses weight sharing network structure to make it more similar to the biological neural network, and the capacity of the model can be adjusted by changing the breadth and depth of the network. In convolutional neural networks, convolution replaces the general matrix multiplication in standard neural networks, so convolutional neural networks can effectively reduce the learning complexity of network models. In addition, the image can be directly imported into the network as the original input, thus avoiding the feature extraction process in the standard machine learning algorithm. The convolutional neural network is superior to the standard fully connected neural network in processing image data of the Internet of Things [10].

The convolution layer of a typical convolutional neural network consists of several Feature maps. Because each neuron in the convolution layer that outputs the feature map is locally connected with its input data, and the input value of the neuron is obtained by summing the corresponding connection weight and local input weight plus the offset value, this process is equivalent to the convolution process, so the neural network with this structure is called convolutional neural network [11][12][13]. The core of the convolutional neural network is the convolutional layer, which consists of several feature maps. Convolution has the characteristics of "weight sharing", which can reduce the number of parameters, reduce the computational cost and prevent over-fitting caused by too many parameters. Compared with the convolution layer, the pooling layer is much simpler. The so-called pooling is to compress each submatrix of the input tensor. If 2*2 is to be pooled, then every 2 *2 element of the submatrix will be turned into an element; if 3*3 is pooled, then every 3*3 element of the submatrix will be turned into an element so that the dimension of the input matrix will become smaller and the complexity of matrix operation will be reduced. If you want to turn every element of the input submatrix into an element, you need a pooling standard. Common pooling standards include maximum pooling and average pooling. Maximum pooling refers to taking the maximum value of corresponding elements as the element value after pooling. Similarly, average pooling means that the average value of corresponding elements is taken as the element value pool layer after pooling, which not only speeds up the operation speed of a convolutional neural network but also reduces the risk of over-fitting of convolutional neural network. Convolutional neural networks may have one or more fully connected layers. Each node of the whole connection layer is connected with all nodes of the previous layer. The output value of the last full connection layer is passed to an output layer, which can be classified by Softmax logic regression. With the development of deep learning, the scale of people using neural networks is increasing. When the model is too large to be stored in the local memory of the working node, it is necessary to split the model. The hierarchical structure of the neural network brings some convenience to its splitting. We can divide the model across layers, and we can divide the model according to the hierarchical structure of the neural network without crossing layers. Cross-layer partition divides the parameters of each layer into several parts equally, and each node stores one part, which is equivalent to storing the whole neural network in each node. According to the layer division, the edge parameters, activation function values, and error propagation values of every two layers of the neural network are stored in a working node.

With the rapid development of deep learning, target detection based on a convolutional neural network model is becoming a mainstream detection method. Common convolutional neural network models applied to target detection include VGG series networks (including VGG-16, VGG-19), Res Net series networks (Res Net18, Res Net34, Res Net50), and so on [14-15].

(1)VGG series network VGG network originated from the Oxford Visual Geometry Group of Oxford University, so it is called VGG for short. At first, to solve the 1000 class image classification problem in Image Net, the VGG network can also be applied to pedestrian re-recognition. Generally, there are two types of VGG networks, VGG-16 and VGG-19, which contain 16 and 19 hidden layers respectively. The VGG network is improved from Alex Net, and VGG replaces the larger convolution kernel in Alex Net with a smaller convolution kernel. In addition to the small convolution kernel mentioned above, the VGG network also uses a small pool kernel. Compared with Alex Net, the layers of the VGG network are deeper and the feature map is wider, which makes the network have a larger receptive field and at the same time reduces the amount of computation to a certain extent. However, there are three fully connected layers in the VGG network, which leads to great computational cost, and the use of more parameters also leads to greater memory occupation.

(2)Res Net series network Res Net network is a deep learning network model proposed by He Kaiming on CVPR in 2016, to solve the problem that deep convolutional neural networks are hard to train. In a VGG network, the convolution network can reach 19 layers, but with the increase of network layers, the network will be degraded: when a small number of networks are trained, the more network layers are trained, the smaller the loss is trained, and then gradually saturated, but when the number of network layers increases to a certain extent, the loss is increased instead of decreased. To solve the above problems, Res Net added a residual unit based on the VGG19 network structure. The residual unit contains two convolution layers, and the input in the VGG19 network will get the output after passing through these two convolution layers in turn. However, after adding the residual unit, the residual unit will add the input and the output through these two convolution layers to get the output. This process is called skip connection. Skip connection makes up for the missing information that may appear in the original network feature extraction to a certain extent. Common Res Net networks are Res Net18, Res Net34, and Res Net50. Res Net18 and Res Net34 networks adopt a two-layer residual structure, while the three-layer residual structure is adopted by Res Net50, and the convolution kernels are 1*1, 3*3, and 1*1 respectively.

2.6. Video surveillance system architecture scheme for edge computing

In the field of traditional security protection, the industry mostly adopts artificial video monitoring methods based on the cloud computing paradigm. In this method, the video data collected at the front end is transmitted to the cloud, the data processing results are obtained by cloud computing, and then transmitted to the client by the network, and the monitor can capture the abnormal targets through the client. At present, the scale of front-end cameras deployed in the monitoring system may be thousands, the network load pressure caused by video data transmission surges sharply, and the demand for cloud data storage and computing power increases, which leads to the system response delay, which is not conducive to the application scenarios with high real-time requirements such as video monitoring. At the same time, the artificial video monitoring mode is limited by the professional quality of the monitoring personnel, and it is easy for human negligence to lead to missed detection and false detection of abnormal signals so that the processing decision of the detected abnormal targets cannot be made immediately. The system architecture scheme designed in this paper is to use edge computing to offload some computing tasks from the cloud to the edge of the network and realize real-time collection and analysis of field data at the data source through edge devices, effectively reducing the data transmission delay. The edge computing device automatically executes the control strategy through the analysis results, gets rid of the influence of human factors on the monitoring quality, achieves the decision response triggered by events, and improves the real-time response of the system. The overall system architecture scheme is shown in Figure 1 below.

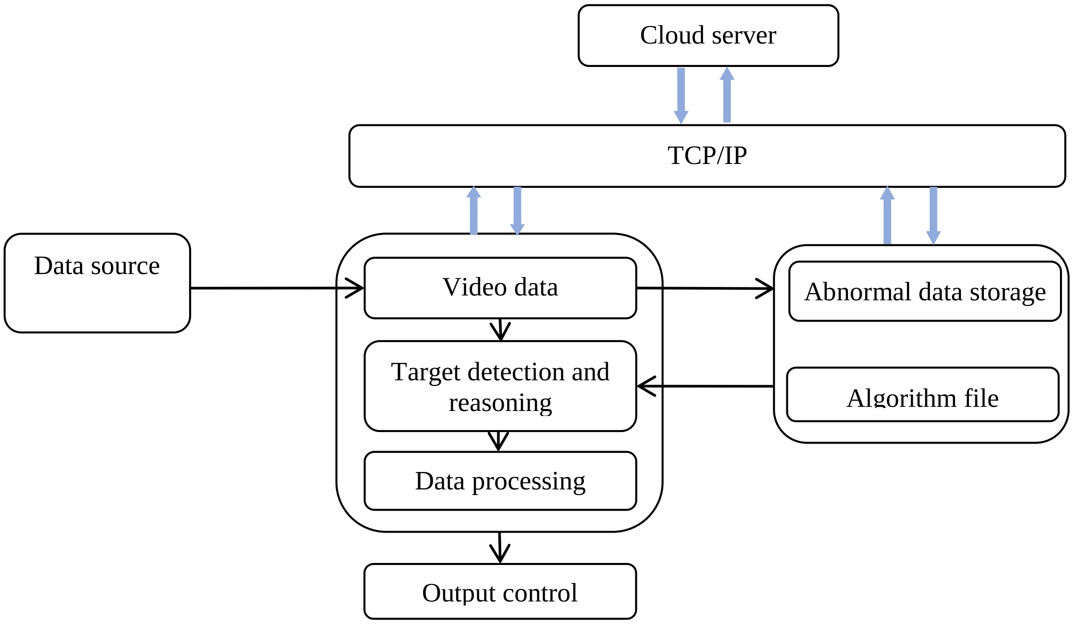

Figure 1: Video surveillance system architecture for edge computing

In this paper, the design of monitoring system architecture scheme hardware consists of an edge computing device, edge server, and cloud. The edge computing device adopts the NVIDIA TX2 development kit; the edge of the server is a PC; the Intelligent nebula server is a cloud computing center. And the parameters and specifications of the PC cloud server are shown in Table 2.

Table 2: Parameter specifications of cloud server and edge server

name	function	GPU	CPU	internal storage	hard disc
Intelligent nebula GPU	Cloud server	GTX 1080Ti 11GB	Intel Xeon E5	32GB	200GB
PC	Edge server	GTX 1050Ti 4GB	Intel Core i5	8GB	1TB

The operation process of the design scheme is as follows:

The edge computing device TX2 turns on the onboard camera and collects front-end data in real-time through the Open CV algorithm program, runs the target detection algorithm based on a deep convolution network on each frame of collected image data to realize abnormal target detection, and when the corresponding abnormal signal is identified, operates the GPIO interface to output high level as the input signal for subsequent control decisions. TX2 The collected picture data can be selectively uploaded to the cache of the PC, which is convenient for operators to find and verify historical data.

The PC receives the algorithm model parameters sent by the cloud and regularly pushes them to TX2 to update the edge computing equipment. At the same time, the PC also stores the abnormal target data collected by TX2 for backup and uploads the data to the cloud for model training at the specified time.

The intelligent cloud server receives the abnormal data from the PC for large-scale model training, and the model parameter file needs to be sent to the PC after the training. The data transmission protocols between the cloud and PC and TX2 all adopt SFTP.

3. Convolutional Neural Network for Edge Computing

3.1. MMGNet deep convolution network

Compared with the cloud, the storage space, running memory, and computing power of edge computing devices are smaller. The target detection algorithm based on a deep convolution network is difficult to run on edge devices in real-time because of the large parameters of the feature extraction network and high computational resources. Therefore, this paper proposes a deep convolution network MMGNET (Multi-scale convolution network based on Mobile Net and Ghost Net) suitable for edge devices for target detection. This network is a deep convolution neural network based on the MobileNet series and GhostNet network, which uses a multi-scale kernel to extract feature information. The standard convolution module Mbotleneck of MMGNet is shown in Figure 2.

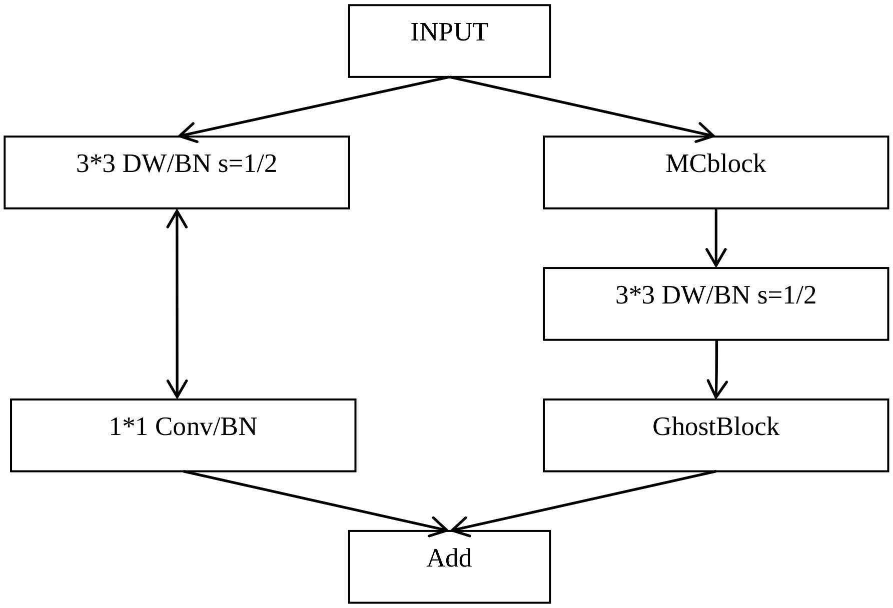

Figure 2: Structural drawing of Mbotleneck module

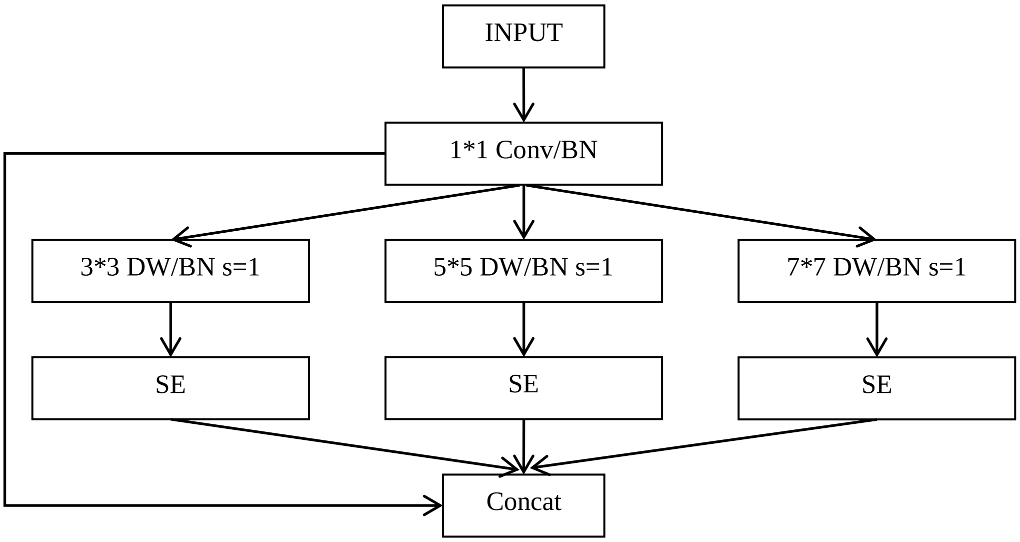

Figure 3: Structural drawing of MCBlock

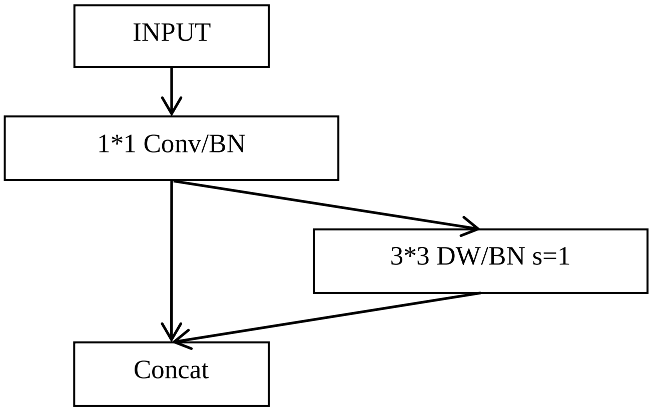

Figure 4: Structural drawing of ghost block

DW/BN means that the data is first deeply convolved with a step size of 1, and then batch normalization; S represents the convolution step, when s=1, the size of the input feature graph is unchanged, and when s=2, the size is reduced by half. Standard convolution modules are stacked to form a deep convolution network. When the input data is 112*112*3 pictures, the design network structure is shown in Table 3.

In the table, Input indicates the dimension of the input feature map, Exp indicates the number of channel expansion, and Out_channels indicates the number of channels of the output feature map; SE=1 means to join the channel attention module; Stride represents convolution operation step size, if the value is 1, it means that the data resolution is unchanged through the network layer, if the value is 2, it means that the data resolution is halved; Conv2D_n represents convolution operation with convolution kernel size n; Avpool means that the input data is globally averaged and pooled; Flatten/Classifier means that the output result of convolution network is expanded into the one-dimensional vector, the probability value of each label is calculated by Softmax function, and the maximum probability value index is taken as the predicted label value. Convolution is used near the input end of the network to reduce the resolution by 1/2, which effectively reduces the network computation. After reducing the size of the feature map, we use several Mbotleneck with the constant resolution to extract features, and then use MBOTLECK with downsampling to reduce the data resolution.

Table 3: Lightweight network structure designed when the input data dimension is 112*112*3 input layer

Name	LayerName	Exp	Out_channels	SE	Stride
112*112*3	Conv2D_5	-	16	-	2
56*56*16	Mbotleneck	48	24	-	1
56*56*24	Mbotleneck	64	24	-	1
56*56*24	Mbotleneck	64	24	-	1
56*56*24	Mbotleneck	144	64	one	2
28*28*64	Mbotleneck	200	64	-	1
28*28*64	Mbotleneck	200	64	-	1
28*28*64	Mbotleneck	224	96	one	2
14*14*96	Mbotleneck	256	96	-	1
14*14*96	Mbotleneck	256	96	one	1
14*14*96	Mbotleneck	256	96	-	1
14*14*96	Mbotleneck	256	120	one	1
14*14*120	Mbotleneck	256	120	one	1
14*14*120	Mbotleneck	288	160	one	2
7*7*160	Mbotleneck	320	160	-	1
7*7*160	Mbotleneck	320	160	one	1
7*7*160	Mbotleneck	320	160	-	1
7*7*160	Conv2D_1	-	720	-	1
7*7*720	Avg Pool	-	720	-	1
1*1*720	Conv2D_1	-	100	-	1
1*1*100	Flatten/Classifier	-	-	-	-

The expansion number of channels in the network is 2-3 times the number of channels in the input data. When the resolution of the input data is high but the number of channels is low, select a larger expansion multiple to enrich the feature information; When the data resolution is low but the number of channels is high, choose low expansion times to avoid the increase of parameter quantity and memory consumption caused by high expansion times of channels. At the end of the network, global pooling and point convolution are used to output characteristic data with a resolution of 1, and the number of channels is equal to the total number of classification categories. After Flatten, a one-dimensional vector is obtained, and then the final category is predicted by the Softmax function. Compared with the full connection layer, it can effectively reduce the increase of parameter quantity and memory consumption, as shown in the following Table 4.

Table 4: Comparison between the end structure of global pooling/coupon product and the full connection layers in terms of parameter quantity and memory consumption

Structure name	Params (MB)	Memory consumption(MB)
Global pooling/point convolution	10.0	84.9
Full connection layer	94.4	295.6

3.2. **Experimental results and analysis**

In this experiment, the intelligent nebula server is used, and the related parameters and specifications are shown in Table 5 below:

Table 5: Hardware parameter specification of the experimental environment

CPU	internal storage	GPU	hard disc
Interxion E5	32GBNVIDIA	GTX1080ti11GB	200GB

The software programming environment is Windows Python 3.7, CUDA10.0, the deep learning framework is Keras and Tensor Flow_GPU_1.15, and other frameworks such as Open CV and Matplotlib are also used for image data processing and visualization of experimental results. Because the deep convolution network contains a large number of parameters, the training process needs large-scale data sets. Through data enhancement technology, the information richness of single picture data can be enhanced, the data volume of the training data set can be increased at low cost, the generalization ability of algorithm can be effectively increased, and the over-fitting phenomenon of the network in the training process can be suppressed. In the experiment, the CIFAR100 data set is used to train and test the design network by randomly rotating the training data set, cropping the area, adding Gaussian noise, and increasing the contrast, brightness, and saturation of pictures. This data set involves 100 object classes, each class covers 600 pictures, totaling 60,000 pictures. The ratio of the training set to the test set of CIFAR100 is 5:1. The test on the CIFAR100 test set shows that the Top-1 accuracy rate is 71.52%. Comparing MMGNet with other classical deep convolution neural networks, the parameters, memory consumption, and Top-1 accuracy are shown in Table 6.

Table 6: Comparison of parameters, running memory, and top-1 Accuracy between MMGNet and partial convolution neural network

ModelName	Params(MB)	Mc(MB)	Top-1Ac(%)
RCNN-160[16]	-	-	68.25
MIM[17]	-	-	70.8
VGG19	228	243.72	72.3
ResNet50[18]	97.8	133.64	77.4
InceptionV3[19]	22.74	103.83	77.2
MobileNet[20]	4.05	32.23	sixty-eight
MobileNetV2[21]	2.36	34.8	69.1
MMGNet	10.0	84.9	71.52

According to the experimental results in Table 6, compared with RCNN-160 and MIM traditional deep convolution network, the MMGNet designed in this paper

The 71.52% prediction accuracy achieved in CIFAR100 is higher than that of the two comparison networks. Compared with VGG19, Res Net50, and Incident V3 deep convolution networks, MMGNet has a slightly lower prediction accuracy parameter but has a significant reduction in parameter quantity and memory consumption, in which the parameter quantity of the network compared with the VGG model is reduced by 95% and the memory consumption of running is reduced by 65.1%, which proves that the network has achieved certain lightweight processing. Compared with two lightweight networks, Mobile Net and Mobile Net V2, the network designed in this paper have increased the number of parameters and memory consumption, but the prediction accuracy is higher than the two algorithms. To sum up, MMGNet deep convolution neural network can achieve lightweight and good prediction accuracy.

4. Design of target detection algorithm based on MMGNet

(1) Target detection algorithm

Coal-bed gas collection sites are gas-well stations, which are mostly distributed in remote mountainous areas and unattended. Therefore, to ensure the safe exploitation of coal-bed gas, video monitoring of coal-bed gas stations is needed. Cloud computing-based manual monitoring method is often used for the safety protection of coalbed methane wells and stations. This scheme relies on the subjective judgment of the monitoring personnel, and it is easy to miss the detection of abnormal targets due to human negligence. The experimental data comes from the video monitoring data set of a coalbed methane well station, and the video data information is shown in Table 7.

Table 7: video data set of coalbed methane well station

Video name	Resolution	Frame rate (frame /s)	Duration (s)
Vedio1	640*480	24	54
Vedio2	1280*720	30	241
Vedio3	1280*720	30	307
Vedio4	1280*720	30	386

Before the experiment, the target detection data set should be constructed. Firstly, the video data set is extracted to read the picture data, and then the target detection and labeling tool label Img manually labels it. Finally, the detected target frame is saved in Pascal VOC data set format, and the XML file corresponding to the picture is automatically generated by label Img, which includes data information such as the category and location of the target frame. The labeled target detection data set is shown in Table 8.

Table 8: Target detection data set of coalbed methane well station

category	Quantity (sheet)
Person	637
Car	335

(1) Experimental result analysis

The results of the experimental analysis are aimed at the prediction results of vehicles and pedestrians. Two types of APS are calculated separately and averaged to get m AP, and then the algorithm is loaded into the NVIDIA TX2 platform to get the actual detection time of a single picture. The experimental results are shown in Table 9.

Table 9: Experimental results of coalbed methane well station test set

Model Name	AP(%)		mAP(%)	Time(s)
YOLOV4-tiny(new)	Car	Person	92.15	0.102
	95.54	88.76		

From the test results in Table 9, it can be seen that the YOLOV4-tiny algorithm based on MMGNet has a good recall rate and accuracy rate for both vehicles and pedestrians. The average accuracy of the two types is 92.15%, which proves that the algorithm has a strong prediction accuracy, and the detection time of a single picture is 0.102 s when running on TX2, which effectively proves that the designed algorithm meets the actual needs.

The YOLOV4-tiny target detection algorithm based on MMGNet is transplanted to NVIDIA TX2, an edge computing device. The detection of video data collected in the field of coalbed methane wells shows that the m AP reaches 92.15%, and the detection time of single picture data is 0.102s, which proves that the target detection algorithm has high detection accuracy and low running speed, and meets the needs of the monitoring system architecture scheme in this paper.

5. Conclusion

Traditional artificial video monitoring methods based on the cloud computing paradigm are prone to generate a large number of redundant video data when monitoring abnormal targets with low probability and the monitoring quality will be reduced due to data transmission delay and human factors. The edge computing paradigm effectively reduces the system response delay, relieves the cloud load pressure, and reduces the resource consumption of data in storage and transmission by collecting, analyzing, and processing data at the edge of the network in real-time. Therefore, it is of great research significance and application value to explore the architecture scheme of a video surveillance system based on edge computing. At the same time, using deep learning technology to detect abnormal targets can effectively eliminate the degradation of monitoring quality caused by human factors, reduce the consumption of human resources and improve the real-time response of system decision-making. This paper focuses on the research of video surveillance system architecture based on edge computing and uses deep learning technologies such as deep convolution neural network and target detection algorithm to realize real-time detection of abnormal targets, which has achieved good verification results in the application scenario of coalbed methane wells and stations. A video surveillance system architecture scheme oriented to edge computing is designed to solve the problems of heavy cloud load pressure, system delay, and human resource consumption in traditional artificial video surveillance systems based on the cloud computing paradigm. This scheme adopts the framework scheme of edge computing device, edge server, and cloud working together, and realizes real-time analysis and processing of collected data by offloading forward reasoning calculation of cloud to edge computing device, which reduces the system response delay caused by video data transmission and also alleviates the degradation of monitoring quality caused by human factors. Putting video data and detection algorithm parameter files on the edge server is beneficial to alleviate the consumption of cloud data storage resources; Use the powerful computing power of the cloud to realize model parameter training and regularly transmit parameter files to edge servers. Through the inspection of monitoring data of coalbed methane wells, it is proved that the design scheme is feasible. A lightweight deep convolution neural network based on a multi-scale convolution kernel is proposed, which can be effectively applied to edge embedded devices. Because of the large number of model parameters and the high proportion of computing resources, it is difficult for embedded devices with small storage space and running memory to run in real-time in a traditional deep convolution network. In this paper, based on Mobile Net series networks and Ghost Net, a new architecture of MMGNet is proposed. Compared with traditional convolutional networks, the number of parameters and memory ratio are effectively reduced, and the network detection accuracy is also maintained at a good level.

Reference

[1] A. Voulodimos, N. Doulamis, & A. Doulamis. (2018). Deep learning for computer Visionpp. A brief review. *Computational Intelligence and Neuroscience*, 1-13.

[2] K. Gai, M. Qiu, & H. Zhao. (2016). Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications*, 59, 46-54.

[3] Y. Mao, C. You, & J. Zhang. (2017). A survey on mobile edge computing. *The Communication Perspective*. *IEEE Communications Surveys & Tutorials*, 19(4), 2322-2358.

[4] J. Redmon & A. Farhadi. (2017). YOLO9000pp. Better, faster, stronger. *IEEE Conference on Computer Vision & Pattern Recognition*, 6517-6525.

[5] J. Redmon & A. Farhadi. (2018). YOLOv3pp. An incremental improvement. *Computer Science*, 4(1), 1-6.

[6] A. Bochkovskiy, C. Y. Wang, & H. Liao. (2020). YOLOv4pp. Optimal speed and accuracy of object detection.

[7] W. Liu, D. Anguelov, & D. Erhan. SSDpp. Single-shot multi-box detector. *Springer Cham*.

[8] C. Perera, J. Y. Hung, & K. W. Chen. (2021). A deep learning approach to identify blepharoptosis by convolutional neural networks. *International Journal of Medical Informatics*, 148(3), 104402.

[9] G. Castellano, C. Castiello, & C. Mencar. (2020). Crowd counting from unmanned aerial vehicles with fully-convolutional neural networks. *2020 International Joint Conference on Neural Networks (IJCNN)*.

[10] T. Elsken, J. H. Metzen, & F. Hutter. (2018). Neural architecture search. A survey. arXiv.

[11] H. Song, X. Liu, & H. Mao. (2017). efficient inference engine on compressed deep neural network. *Deep Compression and EIE Hot Chips 28 Symposium, IEEE*.

[12] Y. He, J. Lin, & Z. Liu. (2018). AMCpp. Auto ML for model compression and acceleration on mobile devices.

[13] Z. Liao & G. Carneiro. (2016). On the importance of normalization layers in deep learning with piecewise linear activation units. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV),* IEEE.

[14] C. Y. Wang, H. Liao, & Y. H. Wu. (2020). CSPNetpp. A new backbone that can enhance the learning capability of CNN. *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* IEEE.

[15] T. Y. Lin, P. Dollar, & R. Girshick. (2017). Feature pyramid networks for object detection. *Conference on Computer Vision and Pattern Recognition (CVPR),* IEEE Computer Society.

[16] L. Ming & X. Hu. (2015). Recurrent convolutional neural network for object recognition. *IEEE Conference on Computer Vision & Pattern Recognition*, IEEE Computer Society, 3367-3375.

[17] Z. Liao & G. Carneiro. (2016). On the importance of normalization layers in deep learning with piecewise linear activation units. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE.

[18] X. Xia, X. Cui, & N. Bing. (2017). Inception-v3 for flower classification. *2017 2nd International Conference on Image, Vision, and Computing (ICIVC)*, IEEE.

[19] K. He, X. Zhang, & S. Ren. (2016). Deep residual learning for image recognition. IEEE.

[20] A. G. Howard, M. Zhu, & B. Chen. (2017). Mobile nets: Efficient convolutional neural networks for mobile vision applications.

[21] M. Sandler, A. Howard, & M. Zhu. (2018). Mobile net V2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.