J Supercomput
DOI 10.1007/s11227-016-1934-1

Intelligent monitor system based on cloud and convolutional neural networks

Binbin Yong[1] · Gaofeng Zhang[1] · Huaming Chen[2] · Qingguo Zhou[1]

Abstract Nowadays, cloud-based services are widely developed. The deployment of cloud technology has boosted the development and application of web services. It reduces the overhead of software virtual machine, and supports a wider range of operating systems. Moreover, it enhances the utilization of infrastructure. With the development of artificial intelligence (AI) technology, especially artificial neural network (ANN), intelligent monitor systems are being raised and developed in our daily life. However, a simple task with a single ANN costs a lot of time and computation resources. Hence, we propose using a cloud-based system to share computation resources for ANN to reduce redundant computation. In this paper, we present an intelligent monitor system, which is based on cloud technology, to provide intelligent monitor services. The system is designed with hybrid convolutional neural networks. It has been used for several intelligent monitor tasks, such as scene change detection, stranger recognition, facial expression recognition and action recognition.

✉ Qingguo Zhou
zhouqg@lzu.edu.cn; zhouqg@gmail.com

Binbin Yong
yongbb14@lzu.edu.cn

Gaofeng Zhang
zhanggaof@lzu.edu.cn

Huaming Chen
hc007@uowmail.edu.au

1 School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China

2 School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia

Published online: 30 November 2016

Keywords Cloud computing · Artificial neural network · Intelligent monitor system · Convolutional neural network

1 Introduction

The service of intelligent monitor system is becoming more and more useful nowadays. With this technology, we could watch the conditions of some places over the network. In fact, it is more useful that this kind of monitoring systems could be helpful by informing us with emergencies, for example, the slip of an elderly or the cry of a child, even the intrusion of a stranger. In some conditions, facial expressions [1–3] could give us some key information in the human–computer interaction scene. To achieve these goals, we need to capture the real-time frames of the monitored scene. Then, these frames would be sent to a server to recognize the relevant information automatically. The recognized results or emergency conditions will be sent to us via network. Hence, one of the most important factors in this kind of system is pattern recognition accuracy in these frames.

Since proposed in 2006 [4], deep learning technology is more and more widely used in the field of machine learning, especially for machine vision. In the year 2012, convolutional neural networks (CNNs) have made a great promote in the recognition of ImageNet datasets [5]. One of the biggest advantage of CNNs is that it could extract features automatically without human intervention. Based on CNN, many machine vision research questions on image and video recognition were conducted [6–9].

However, another important factor in intelligent monitor system is that it needs a lot of computation resources in CNN for a single user. That is why CNN had not been widely used since it was first proposed [10]. Relying on the highly development of virtualization techniques, hardware techniques, massive storage techniques and distributed computing technology etc, the cloud computing technology [11,12] has been popular among lots of users to share computation resources so as to reduce the cost and make full use of the computers.

In this paper, we design an intelligent monitoring service system based on cloud technology to recognize scenario at special places. Our system is a fusion of some trained CNNs through cloud server. These CNNs are used to make different feature recognition based on the frames from webcam streams. As shown in Fig. 1, our system consists of three parts including webcams, CNNs cloud server and mobile terminals. We use one webcam to capture the real-time frames from remote scene. These frames are sent to our cloud server. Our cloud server will collect datasets from users and train CNN models. The recognition monitor results are computed by cloud server and transmitted to the mobile terminals of users. For similar monitor applications, users share trained models, which save a lot of computing resources.

The core of our monitoring system is the hybrid CNN cloud server. In this part, we develop a series of trained CNNs applications for different pattern recognition tasks. These tasks include changes recognition, face recognition, facial expression recognition and action recognition.

As shown in Fig. 2, we make recognition tasks from some live webcams. The frame rates of our webcam are sixty frames per second with a resolution of 640×480. The

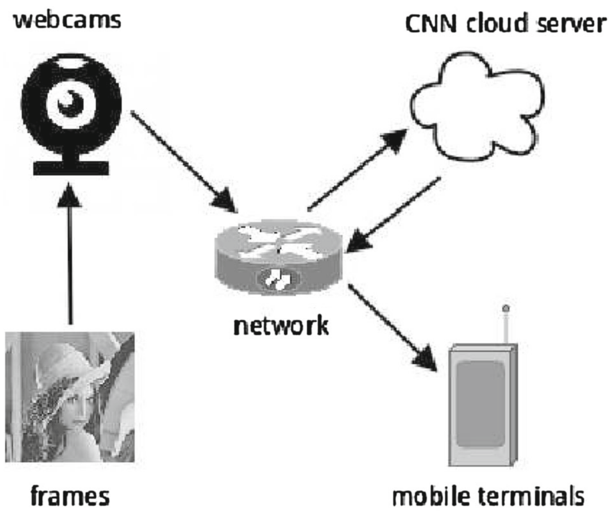


Fig. 1 The cloud server structure for intelligent monitor system

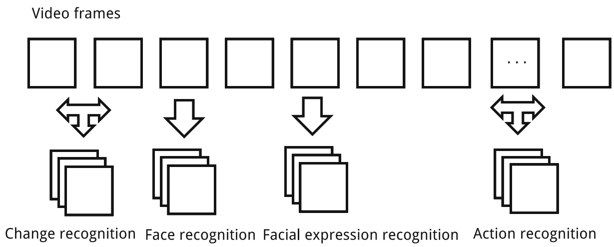


Fig. 2 The monitor and recognition of webcam frames

resolutions of different webcams are not so important because our cloud server will compress the frame images into same size to input the corresponding recognition algorithms. In fact, our cloud server firstly takes one frame out of three frames to compose a 20-frame sequence. When receiving these frames, a corresponding CNN application is invoked to response the request. The predicted results by CNN applications are sent to the terminals preset by the users.

The rest of the paper is organised as follows: Sect. 2 reviews the related technologies in our system. Section 3 articulates the specific recognition applications provided in our monitoring system services. Section 4 discusses some details of cloud server. Section 5 concludes with a summary.

2 Related technologies

2.1 Artificial neural networks and MLPs

Nowadays, artificial neural network (ANN), especially deep neural network (DNN), is concerned as a most popular research aspect in machine learning field [13–15].

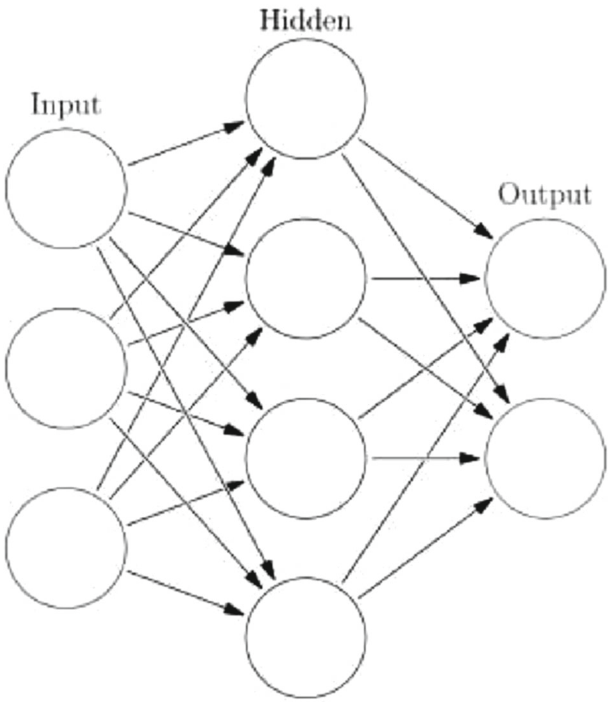


Fig. 3 The architecture of artificial neural network

ANN is generally regarded as a network used to approximate a function that maps an input vector to an output vector. Multilayer perceptron (MLP) [16–18] is a kind of implementation of ANN. As shown in Fig. 3, an MLP includes three parts, which are input layer, hidden layer and output layer. The input layer is normally a vector stands for features. There may be more than one hidden layer with some hidden neurons. Every hidden neuron is abstracted as a nonlinear activation function, such as sigmoid or tanh function. The output layer is a vector whose values are the weighted sums of the output of hidden units. In some cases for classification, the output layers are often implemented with a logistic regression function sigmoid for binary classification or a softmax function for multi-classification tasks [19,20]. MLP is proved to be effective in classifying tasks. Hence, in our system, MLP is used as a classifier after feature extraction.

2.2 Convolutional neural networks

More and more studies confirm that CNN is effective in object classification [21, 22]. In this kind of tasks, a CNN is usually a two-dimensional (2D) convolutional neural networks. As shown in Fig. 4, a 2D CNN is generally built with input layer, convolutional layer, subsampling layer, fully connected (FC) layer and output layer. The first layer is called input layer, which receives an array from an image matrix as

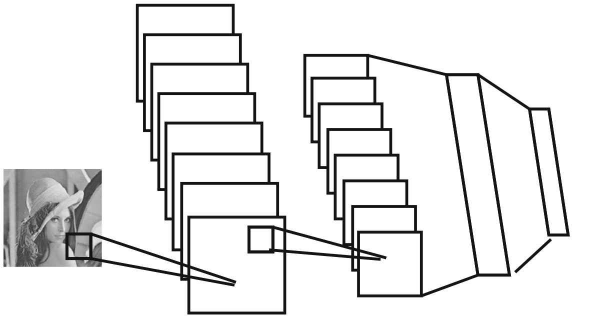


Fig. 4 The structure of convolutional neural network

the input of the CNN. The second layer is often a convolutional layer. The neurons in convolutional layer are connected only to a subset of neurons in the previous layer, and the weights are shared between a subset of neurons in the convolutional layer to reduce the number of parameters that need to be learned. After convolutional layer, there is usually a subsampling layer. In this paper, the subsampling is indeed max pooling operation which is extremely popular [23,24]. That is, the maximum value of features over a small patch is passed to next layer. Generally speaking, a CNN may contain several couple convolutional-subsampling layers according to the specific application scenarios. These layers are used to extract features from a 2D matrix input of an image. Hence, a few FC layers are added after the convolutional-subsampling layers. These FC layers are similar to traditional MLP neutral networks, which are used for classification tasks. The output layer of MLP for classification task is usually a softmax function.

In the recognition of human action, three-dimensional (3D) CNN is used in this paper, which will be presented in Sect. 3.4.

The default nonlinearity function used in our CNNs is the rectifier function, which is defined as $\max(0, x)$. This function is a very popular choice of activation function nowadays [25–27].

2.3 Principal component analysis method

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components [28,29]. The number of features after principal components analysis is generally less than the number of original variables. This transformation is executed in the goal that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. Just as important, the method is sensitive to the relative scaling of the original variables.

2.4 OpenCV

Open source computer vision[1] (OpenCV) is a library of programming functions mainly aimed at real-time computer vision [30,31]. We will use this library to detect some features in our system, especially for face detection.

2.5 Cloud computing

Generally speaking, cloud computing includes three types of models, which are infrastructure as a service (IaaS) [32,33], platform as a service (PaaS) [34–36] and software as a service (SaaS) [37–39]. The development of IaaS is based on the development of virtualization technology. The famous IaaS operators include Amazon and AT&T. The PaaS model is dependent on the development of distributed computing technology. And the typical platforms based on PaaS are App Engine by Google and Azure by Microsoft. As an emerging cloud technology, the SaaS has achieved rapid development since the beginning of the century. It avoids the management of the operating system and provides applications based on cloud technology. In fact, the system structure designed in our paper is a SaaS structure. We use cloud technology to share intelligence algorithms and computation resources. At the same time, we collect training datasets from users to improve our services and share trained results to users.

3 Recognition applications

In this part, the main recognition applications will be discussed here.

3.1 Changes recognition

The first recognition task is to recognize the changes in the webcams. Our system will do recognition task every six frames, that is an interval of 0.1 s. Because changes recognition is a very frequent request, a simple algorithm is used in our cloud server to reduce frequently computing. We capture two frames in 0.1 s and compare the differences. We use a fairy simple but an effective way to implement this. Firstly, we convert the two frames into gray images whose values of each pixel range from 0 to 255. Then, we set the value of each pixel to 0 or 1 by comparing it to 128. This operation eliminates the influence of the strength of the light. At last, we count the number of different pixel values of these two frame images, and divided by the number of the total pixels in one frame to get the similarity ratio. In our implementation, we found that two frames are unchanged with a similarity ratio value bigger than 92%. In general, it has little influence if an unchanged frame is wrongly classified. It only increases a little computation with other recognition. Hence, we can set a higher threshold of similarity ratio to make a strictly detection of change. In our system, the similarity ratio of two frames is set as 96% to determine a change action.

[1] http://opencv.org/.

3.2 Face recognition

Our system provides service to recognize strangers automatically. The way is to train a CNN model with collected face images. To protect privacy of users, the face recognition models are independent of different users. The face recognition CNN model is composed of two conv-subsampling layers that have 20 convolution kernels of size 5×5 and max-pooling size of 4×4, followed by an FC layer with 2000 hidden units and an output layer with two units activated by sigmoid function.

The system gathers training face pictures for training CNN model before recognition tasks. We collect 20 face pictures for one person and ten persons with total 200 face pictures as dataset. The CNN model is trained with 160 face pictures. The remaining 40 pictures are used to test.

We trained our CNN application on a server with a NVIDIA Tesla K40 GPU. The learning rate is set as 0.01 and batch size is set as 20. We iterated seven epochs to reach a best validation score of 100% accuracy. However, we may need to add new image datasets of familiar persons to our system. If new persons are added to the application, the datasets will be updated and the model of the application will be replaced with a new trained one.

3.3 Human facial expression recognition

Human facial expression recognition is a long researched important but also difficult subject [40–43]. It is an important part in the human–computer interactive system. Generally speaking, human facial expressions are categorized into seven classes, which are angry, disgust, fear, happy, sad, surprise and neutral. In our experiment, we found that the expression of fear is similarity to the expression of surprise. Hence, we use surprise expression stands for these two expressions to get a good prediction result. As shown in Fig. 5, we collect 400 facial expression images with six type expressions. 320

Fig. 5 The human facial expressions collected by cloud server

Table 1 The MLP classification accuracy of different number of hidden units

nHidden	Train acc	Test acc
10	1.0	0.883720930233
20	1.0	0.813953488372
30	1.0	0.93023255814
40	1.0	0.813953488372
50	1.0	0.953488372093
60	1.0	0.790697674419
70	1.0	0.813953488372
80	1.0	0.813953488372
90	1.0	0.860465116279
100	1.0	0.790697674419

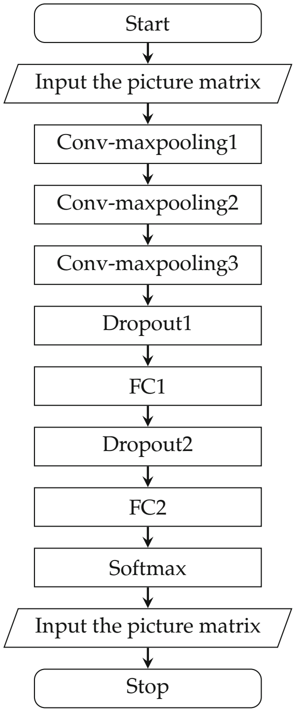
images are used to train, and 80 images are used to test. In fact, the training datasets will increase with the submission of different users.

To recognize facial features, two steps are usually implemented. Firstly, we need to make feature extraction. Then, a wide variety of classifiers are tried in feature classification. In our paper, two ways are tested to make facial expression recognition.

The first way is a PCA-based method. In general, each frame captured from our webcams is an image of size 640×480. Firstly, we convert the image into gray scale images. Secondly, we detect and cut the head area with a rectangle of size 200×200 by opencv. Then, we get a matrix of size 200×200. Thirdly, we extract 100 features by PCA method to create an input vector of size 100. At last, we create a MLP classifier with one hidden layer to classify the features into different facial expressions. The input vector has 100 units of the features extracted by PCA method. The output layer has six output units stands for different facial expressions. For the hyper-parameter of the number of hidden units, we try the number range from 10 to 100 to find a most suitable hidden layer structure. The experimental results are summarized in Table 1. Results show that, when the number of hidden units is set as 50, the best test accuracy of the first way is approximated as 0.95.

Generally, feature extraction is a complex work, especially by manual means. Hence, a CNN model is used to extract facial expression features automatically in the second way. As shown in Fig. 6, our CNN model is made up of 3 conv-maxpooling layers with 32 convolutional kernel filters and max pooling size of 2. After three conv-maxpooling layers, we use two FC layers and a softmax input layer to classify the six types of facial expressions. Using dropout technology in ANN is a simple way to avoid over-fitting [44]. It disconnects some connections in FC layers randomly to achieve a better generalization when training. We use dropout rates of 0.5 and 0.25 at layer FC1 and layer FC2. The numbers of hidden units of FC1 and FC2 are both 3000. We set the learning rate as 0.003 and the batch size as 25. After 2880 iterations, we achieve a best validation accuracy of almost 1.0. The train history is shown in Fig. 7; the validation accuracy fluctuates a little when it reaches almost 1.0.

Fig. 6 The flow chart of CNN used for facial expression recognition



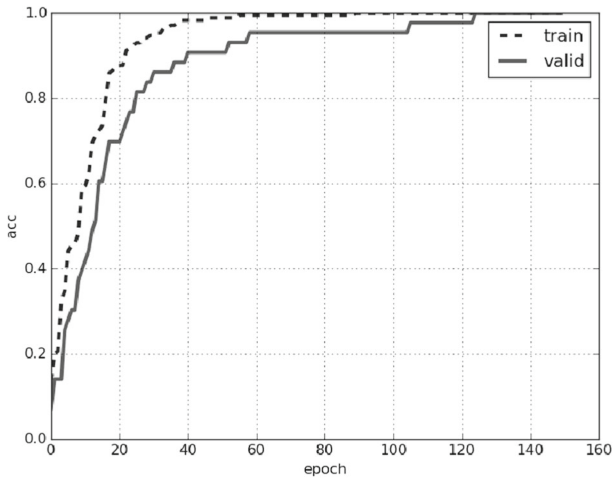


Fig. 7 The train history of CNN for facial expressions recognition

Overall, as we can see from the experimental results, using CNN in facial expression recognition is a better way than PCA-based method. Hence, in our implementation of recognition services, CNN method is adopted (Fig. 8).

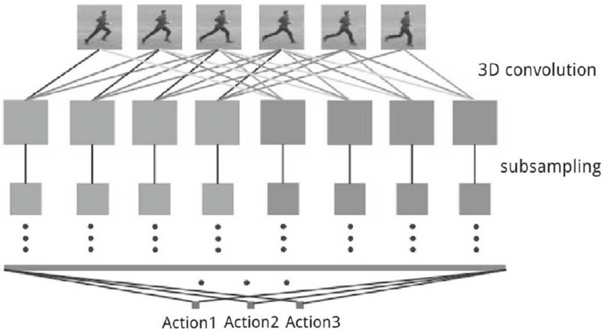


Fig. 8 The structure of 3D CNN

3.4 Action recognition

Action recognition is a very meaningful topic for behavior monitor. It is similar for different persons, so the recognition model can share datasets for different users. In this paper, a 3D CNN is used for human action recognition [45]. 3D CNN is a little different from 2D CNN. It extracts both space and time features by 3D Convolutional kernels. These 3D convolutional filters make convolution operations in both space and time dimensions. Hence, it will capture the inherent relations between continuous frames.

As a test, we experiment the action recognition model with a public KTH dataset.[2] This dataset has boxing, handclapping, handwaving, jogging, running and walking six types of actions. For each action, we choose 100 videos as samples.

In general, we want to find the internal relations between adjacent frames to represent the action features. Hence, the local details in a frame are not so important for the algorithm. To reduce computation, we compress a frame image into 16×16 size. At the same time, we select continuous 20 frames as an action frame sequence. That is to say, the input of the 3D CNN model is matrices of 20 images of size 16×16. There are 32 3D convolutional filters in 3D convolutional layer. The size of these filters is set as a grid of $5 \times 5 \times 5$. The subsampling layer uses a 3D max-pooling with a size of $3 \times 3 \times 3$. It means that 27 adjacent units are sampled into one unit whose value is maximum. In the classification layer, we use an FC layer with 200 hidden units and a dropout rate of 0.5. The output layer is set as a softmax function used to classify different actions. As shown in Fig. 9, we trained our 3D CNN for 50 epochs and reached a best valid accuracy of 0.7917.

4 The cloud server

The cloud server is an important part in our system to provide monitor services. Our cloud server is designed on SaaS structure and multi-tenancy environment, as shown in Fig. 10. In this situation, users can access the shared or private services via

[2] http://www.nada.kth.se/cvap/actions/.

Fig. 9 The train history of 3D CNN for action recognition

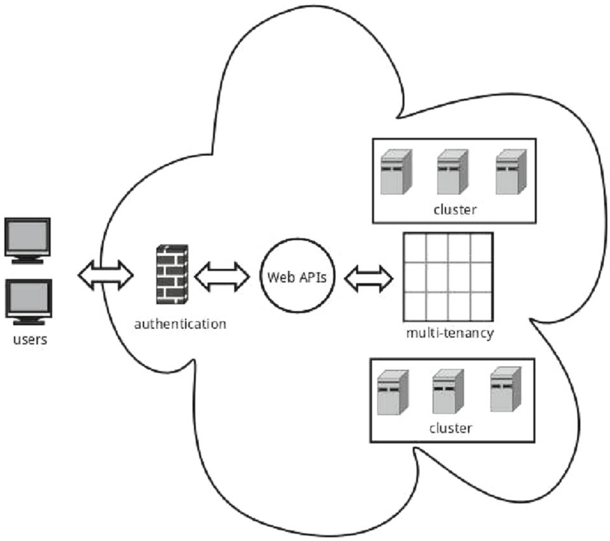


Fig. 10 The structure of cloud server

a preset account. The cloud service is designed with a series web APIs (application programming interface). The server is build on top of the distributed computing cluster.

4.1 The video stream from multi-webcams

Generally, the inputs of recognition algorithms are 2D images. One of the factors that caused wrong recognition is that the capturing images by webcams were not facing the opposite human face directly. Therefore, our system supports monitoring from multi-webcams. It increases some computations, but makes a more accurate monitoring result. In fact, suppose a webcam could detect face around 30°, six webcams will detect face from any angles on the same elevation. Hence, in our system, multi-webcams are available for users. And our algorithm simply summarizes the recognition results from these webcams.

4.2 Web APIs

Our cloud system provides a set of APIs for users to operate. It includes authentication APIs, configuration APIs, intelligent monitor APIs and data collection APIs.

Authentication APIs are used for login validation. Configuration APIs are used for users to set their personal information, such as the addresses of webcams and the informed terminals. Intelligent monitor APIs are interfaces for authenticated users to access the monitor services provided by our monitor cloud server. It continuously uploads frame data from webcams and returns monitor results to preset terminals. Data collection APIs are used for users to upload their training datasets, such as face images and facial expression images. In some cases, for example, face recognition task, the datasets and algorithms are used, respectively, for different users. In the case for similar tasks, such as facial expression recognition and action recognition, the datasets and algorithms are shared by all users. Hence, a more accurate recognition algorithm can be trained with a large scale of shared data. It is important that, with the increase of the numbers of datasets, the structures of our recognition models will change gradually. For example, more hidden neuron units are added to our model to avoid over-fitting.

4.3 Distributed computing cluster

Training and prediction are main parts in our services. Generally, a large number of images need to be processed and trained. Hence, this kind of computation requires an enormous amount of computing power. It is a critical challenge for traditional parallel technologies, such as message passing interface (MPI), to process so much data. To achieve faster convergence in training higher-dimensional ANN algorithms, parallel and distributed computing technologies are introduced to our system.

Our computing cluster consists of many distributed compute nodes. To accelerate these ANN (or CNN) algorithms, the compute nodes are composed of GPUs to perform parallel computing. In fact, traditional MPI or MapReduce [46] is not very suitable for our system to manage computation resources. Because it needs a lot of iterations to train a neural network, and the training process is a compute-intensive and data-intensive task. Hence, we design a compute nodes management system (CNMS) to manage all compute nodes.

As shown in Fig. 11, the request tasks from users are divided into four types of tasks, which are prediction task, training task, datasets task and other task. Prediction queue is used to organize the monitor request tasks. Training queue stores the tasks to train monitor algorithms. Datasets queue handles the requests for operating (mainly

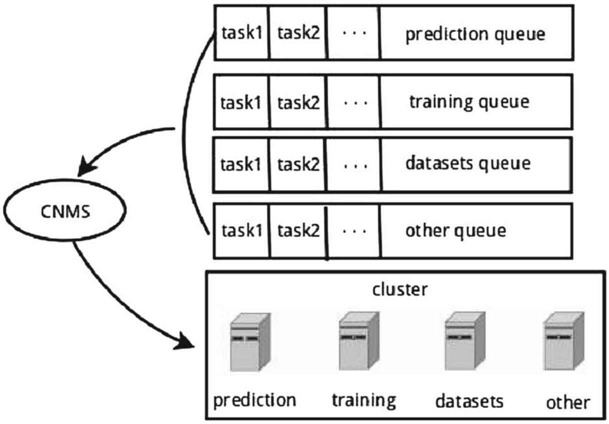


Fig. 11 Task scheduling and resource management

collecting) training data. The other queue is the requests for users to submit basic configuration.

When a request comes, the CNMS places the request in the corresponding queue. When scheduler is performed, CNMS goes through the four types of task queues. It pops a task and finds underutilized compute nodes to be assigned to these tasks. In general, prediction tasks of monitor system need to be processed in real time. Hence, prediction tasks have a higher priority. In the specific implementation, 50% compute nodes are assigned to prediction tasks. 40% compute nodes are used to train algorithms. 5% compute nodes are used to collect training data. In these data collecting nodes, parallel computation based on GPU is not required. However, it requires fast network environment and high-speed storage system. The last 5% compute nodes are used to answer basic requests from users. The assignment strategy is not absolute. In fact, CNMS will distribute more compute nodes to training tasks when other compute nodes are free.

5 Conclusion

Intelligence monitor system is becoming more and more important in the coming days. Cloud-based services are developing at a fast pace. The service of intelligence monitor system based on cloud will appear in the future. More and more works will be done under the control of this kind of system. As two hot issues in the field of computer science, cloud technology and machine learning have made great progress in the past decade. The combination of these two technologies is a direction in the next generation cloud technology.

This paper designs a remote monitoring system based on the cloud. Some useful monitoring tasks such as changes recognition, face recognition, facial expression recognition and action recognition are discussed here. CNN models are used in our system to do recognition tasks. We also tested the recognition accuracy for each task and the impact factors. Our work proves that designing such a monitoring system is feasible. We firstly designed the detailed structure of a monitoring system. Then, we presented some useful monitor applications and the detailed structure of each recognition application. At last, we discuss about the work of the cloud server.

However, work still remains to be done to construct a mature cloud-based monitor system, such as real-time performance, security and resource utilization. The influence of the number of nodes in the cluster, the data transfer rate and the execution time of tasks needs a more in-depth research. We also need to enrich the services in this kind of system. In fact, to train a universal deep model for human facial expression recognition, large amounts of training data need to be collected, for reasons such as racial difference and individual difference. Hence, the system constantly collects data from different users to gradually improve the recognition applications.

Section 4.1 simply summarizes the monitor results from different webcams. In fact, designing an algorithm to simultaneously process the frames from a scene but different webcams is a better choice. Work still remains to refine the concepts presented in our monitor cloud system and to improve the recognition accuracy in some applications. To achieve this, firstly we need to improve the accurate of CNN model for each task. Then, we need to collect more datasets from users. At last, parallel computation technology needs to be fully developed to improve the computational efficiency of our cloud server.

Acknowledgements This work was supported by Dongguan's Recruitment of Innovation and entrepreneurship talent program, National Natural Science Foundation of China under Grant Nos. 61402210 and 60973137, Program for New Century Excellent Talents in University under Grant No. NCET-12-0250, Strategic Priority Research Program of the Chinese Academy of Sciences with Grant No. XDA03030100, Gansu Sci. and Tech. Program under Grant Nos. 1104GKCA049, 1204GKCA061 and 1304GKCA018, Google Research Awards and Google Faculty Award, China.

References

1. El-Shazly EH, Abdelwahab MM, Taniguchi RI (2015) Efficient facial and facial expression recognition using canonical correlation analysis for transform domain features fusion and classification. In: 2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), pp 639–644. doi:10.1109/SITIS.2015.57
2. Deshmukh S, Patwardhan M, Mahajan A (2016) Survey on real-time facial expression recognition techniques. IET Biometr 5(3):155. doi:10.1049/iet-bmt.2014.0104
3. Chen J, Chen Z, Chi Z, Fu H (2016) Facial expression recognition in video with multiple feature fusion. IEEE Trans Affect Comput PP(99):1. doi:10.1109/TAFFC.2016.2593719
4. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527. doi:10.1162/neco.2006.18.7.1527
5. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc., pp 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf. Accessed 6 Dec 2012
6. Bappy JH, Roy-Chowdhury AK (2016) CNN based region proposals for efficient object detection. In: 2016 IEEE International Conference on Image Processing (ICIP), pp 3658–3662. doi:10.1109/ICIP.2016.7533042
7. Basaeed E, Bhaskar H, Al-Mualla M (2015) CNN-based multi-band fused boundary detection for remotely sensed images. In: 6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15), pp 1–6. doi:10.1049/ic.2015.0109
8. Chen L, Wang S, Fan W, Sun J, Naoi S (2015) Beyond human recognition: a CNN-based framework for handwritten character recognition. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp 695–699. doi:10.1109/ACPR.2015.7486592

9. Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware CNN model. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp 1134–1142. doi:10.1109/ICCV.2015.135
10. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278. doi:10.1109/5.726791
11. Chalaemwongwan N, Kurutach W (2016) Mobile cloud computing: a survey and propose solution framework. In: 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp 1–4. doi:10.1109/ECTICon.2016.7561437
12. Celesti A, Fazio M, Celesti F, Sannino G, Campo S, Villari M (2016) New trends in biotechnology: the point on NGS Cloud computing solutions. In: 2016 IEEE Symposium on Computers and Communication (ISCC), pp 267–270. doi:10.1109/ISCC.2016.7543751
13. Guang-ping Y, Jing-yang W, Ming-zhe Y, Yang Y (2015) Research on prediction method of sludge bulking based on ANN and grey Markov model. In: 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), pp 1622–1627. doi:10.1109/CYBER.2015.7288188
14. Belhouchet K, Bayadi A, Bendib ME (2015) Artificial neural networks (ANN) and genetic algorithm modeling and identification of arc parameter in insulators flashover voltage and leakage current. In: 2015 4th International Conference on Electrical Engineering (ICEE), pp 1–6. doi:10.1109/INTEE.2015.7416698
15. Buch H, Dudani KK, Pipalava DP (2015) Prediction of total transfer capability using ANN in restructured power system. In: 2015 5th Nirma University International Conference on Engineering (NUiCONE), pp 1–5. doi:10.1109/NUICONE.2015.7449620
16. Barman D, Chowdhury N, Singha RK (2012) To predict possible profit/loss of a movie to be launched using MLP with back-propagation learning. In: 2012 International Conference on Communications, Devices and Intelligent Systems (CODIS), pp 322–325. doi:10.1109/CODIS.2012.6422203
17. Azarloo A, Farokhi F (2012) Automatic musical instrument recognition using K-NN and MLP neural networks. In: 2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), pp 289–294. doi:10.1109/CICSyN.2012.61
18. Alves EDS, Filho JBOS, Galliez RM, Kritski A (2013) Specialized MLP classifiers to support the isolation of patients suspected of pulmonary tuberculosis. In: 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, pp 40–45. doi:10.1109/BRICS-CCI-CBIC.2013.18
19. Kumarawadu S, Watanabe K, Kiguchi K, Izumi K (2002) Adaptive output tracking of partly known robotic systems using SoftMax function networks. In: Proceedings of the 2002 International Joint Conference on Neural Networks, 2002. IJCNN '02, vol 1, pp 483–488. doi:10.1109/IJCNN.2002.1005520
20. Ahmed N, Campbell M (2011) Variational Bayesian learning of probabilistic discriminative models with latent softmax variables. IEEE Trans Signal Process 59(7):3143. doi:10.1109/TSP.2011.2144587
21. Cao Z, Principe JC, Ouyang B, Dalgleish F, Vuorenkoski A (2015) Marine animal classification using combined CNN and hand-designed image features. In: OCEANS 2015—MTS/IEEE Washington, pp 1–6
22. Albora AM (2016) Wave-CNN method approach of archaeogeophysics studies. In: 2016 24th Signal Processing and Communication Application Conference (SIU), pp 173–176. doi:10.1109/SIU.2016.7495705
23. Masci J, Meier U, Ciresan D, Schmidhuber J, Fricout G (2012) Steel defect classification with max-pooling convolutional neural networks. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp 1–6. doi:10.1109/IJCNN.2012.6252468
24. Masci J, Giusti A, Ciresan D, Fricout G, Schmidhuber J (2013) A fast learning algorithm for image segmentation with max-pooling convolutional networks. In: 2013 IEEE International Conference on Image Processing, pp 2713–2717. doi:10.1109/ICIP.2013.6738559
25. Dahl GE, Sainath TN, Hinton GE (2013) Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp 8609–8613. doi:10.1109/ICASSP.2013.6639346
26. Hara K, Saito D, Shouno H (2015) Analysis of function of rectified linear unit used in deep learning. In: 2015 International Joint Conference on Neural Networks (IJCNN), pp 1–8. doi:10.1109/IJCNN.2015.7280578

27. Xiong S, Guo W, Liu D (2014) The Vietnamese speech recognition based on rectified linear units deep neural network and spoken term detection system combination. In: 2014 9th International Symposium on Chinese Spoken Language Processing (ISCSLP), pp 183–186. doi:10.1109/ISCSLP.2014.6936574
28. Alberto AC, Limeira GA, Nadal J (2015) A principal component analysis approach to heart rate turbulence assessment in Chagas disease. In: 2015 Computing in Cardiology Conference (CinC), pp 681–684. doi:10.1109/CIC.2015.7411002
29. Meraoumia A, Korichi M, Chitroub S, Bouridane A (2015) Hidden Markov models principal component analysis for multispectral palmprint identification. In: 2015 5th International Conference on Information Communication Technology and Accessibility (ICTA), pp 1–6. doi:10.1109/ICTA.2015.7426898
30. Guo-rong Z, Chang-zhen X, Yan Z (2011) A method of embedded video surveillance based on OpenCV. In: 2011 International Conference on E-Business and E-Government (ICEE), pp 1–4. doi:10.1109/ICEBEG.2011.5881671
31. Chaczko Z, Braun R (2010) Teaching Computer Vision for telemedicine systems using OpenCV. In: 2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET), pp 17–20. doi:10.1109/ITHET.2010.5480069
32. Amri SMA, Guan L (2016) Infrastructure as a service: exploring network access control challenges. In: 2016 SAI Computing Conference (SAI), pp 596–603. doi:10.1109/SAI.2016.7556042
33. Dawoud W, Takouna I, Meinel C (2010) Infrastructure as a service security: challenges and solutions. In: 2010 The 7th International Conference on Informatics and Systems (INFOS), pp 1–8
34. Akinbi A, Pereira E, Beaumont C (2013) Evaluating security mechanisms implemented on public platform-as-a-service cloud environments case study: Windows Azure. In: 2013 8th International Conference for Internet Technology and Secured Transactions (ICITST), pp 162–167. doi:10.1109/ICITST.2013.6750183
35. Boniface M, Nasser B, Papay J, Phillips SC, Servin A, Yang X, Zlatev Z, Gogouvitis SV, Katsaros G, Konstanteli K, Kousiouris G, Menychtas A, Kyriazis D (2010) Platform-as-a-service architecture for real-time quality of service management in clouds. In: 2010 Fifth International Conference on Internet and Web Applications and Services (ICIW), pp 155–160. doi:10.1109/ICIW.2010.91
36. Tian W, Su S, Lu G (2010) A framework for implementing and managing platform as a service in a virtual cloud computing lab. In: 2010 Second International Workshop on Education Technology and Computer Science (ETCS), vol 2, pp 273–276. doi:10.1109/ETCS.2010.126
37. Afify YM, Moawad IF, Badr NL, Tolba MF (2013) A semantic-based Software-as-a-Service (SaaS) discovery and selection system. In: 2013 8th International Conference on Computer Engineering Systems (ICCES), pp 57–63. doi:10.1109/ICCES.2013.6707171
38. Akande AO, Belle JPV (2014) Cloud computing in higher education: a snapshot of software as a service. In: 2014 IEEE 6th International Conference on Adaptive Science Technology (ICAST), pp 1–5. doi:10.1109/ICASTECH.2014.7068111
39. Alkalbani AM, Ghamry AM, Hussain FK, Hussain OK (2016) Sentiment analysis and classification for software as a service reviews. In: 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), pp 53–58. doi:10.1109/AINA.2016.148
40. Çam BC, Yalçın M, Yavuz HS (2016) Facial expression recognition based on locational features. In: 2016 24th Signal Processing and Communication Application Conference (SIU), pp 1449–1452. doi:10.1109/SIU.2016.7496023
41. Agada R, Yan J (2015) A model of local binary pattern feature descriptor for valence facial expression classification. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp 634–639. doi:10.1109/ICMLA.2015.185
42. Apte A, Basavaraj A, N.R. K, (2015) Efficient facial expression ecognition and classification system based on morphological processing of frontal face images. In: 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS), pp 366–371. doi:10.1109/ICIINFS.2015.7399039
43. Borah S, Konwar S (2014) ANN based human facial expression recognition in color images. In: 2014 International Conference on High Performance Computing and Applications (ICHPCA), pp 1–6. doi:10.1109/ICHPCA.2014.7045337
44. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929. http://dl.acm.org/citation.cfm?id=2627435.2670313

45. Ji S, Xu W, Yang M, Yu K (2013) 3D convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221. doi:10.1109/TPAMI.2012.59
46. Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107. doi:10.1145/1327452.1327492