Cooperative Training Video Surveillance Technology under the Edge Computing

Dong Lian\textsuperscript{1}, Aidong Xu\textsuperscript{2}, Songlin Chen\textsuperscript{1}, Xinchen Xu\textsuperscript{3}, Yixin Jiang\textsuperscript{2} and Wen Hong\textsuperscript{1}

\textsuperscript{1}National Key Lab of Commun. University of Elec. Science and Tech. of China, Chengdu, China
\textsuperscript{2}EPRI, China Southern Power Grid Co. Ltd., Guangzhou, China.
\textsuperscript{3}Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province, UESTC, Chengdu 611731, China

Abstract. Edge computing offsets the computing power devices to close the terminals, which avoids delay caused by long-distance transmission and network congestion under the cloud computing support scenarios. This paper presents the video surveillance model under the edge computing. A novel cooperative training edge computing system for the image processing is proposed to realize the shortest training and processing time. The simulation of the novel system is performed under Tensorflow software and the results verify the advantages of the new system.

1. Introduction

Edge computing is a new computing system and technology that sinks computing power from the cloud to the edge of the network. "Edge" is a relative concept that refers to any computing resource and network resource from the data source to the cloud computing centre data path.

The video surveillance system under the edge computing has real-time, flexibility and intelligence. Transmission line detection is an important application scenario of intelligent video systems in the power industry under edge computing, as shown in Figure 1. Using UAV in transmission line inspections is not restricted by terrain and landforms, and is especially suitable for patrolling work in rugged mountainous areas and multiple rivers.

Figure 1. UAV inspection based on edge computing
2. **Video Surveillance System Model under Edge Computing**

The video surveillance system under edge computing puts the functions of video pre-processing, acquisition, feature analysis, behaviour recognition, and decision-making on intelligent edge devices, removing redundant information from the image and make some judgments and decisions.

![Figure 2. Video surveillance system model under edge computing](image)

3. **Face Recognition Model Based on Convolutional Neural Network**

The proposed cooperative training edge computing system consists of a face data set, a training set and training models. It can make decisions, analyse, or judge real-time video surveillance data based on different behaviour characteristics, ensure the real-time nature of the surveillance video stream and realize the local processing of video surveillance data. The novel system aims to develop the shortest training and processing time realize the image feature.

![Figure 3. Convolutional neural network model](image)

The CNN model is trained, and the training data and test set data are according to a ratio of 25: 1. The training is divided into two typess of faces [0,1] and [1,0]. The entire model is divided into eight Layers: the convolution pooling layer is iterated three times for a total of six layers, and then the fully connected layer and the output layer.

### 3.1. Input Layer

The input layer is the input of the entire neural network. In this convolutional neural network, the input layer is a pixel matrix of a picture.

### 3.2. Convolution Layer

The input of the new node in the convolutional layer is the feature map of the upper network, and the feature map is the image itself. The input depth is D, then the depth D of the grayscale image is 1, and
the color image is mapped by the features of the three RGB color channels, and the depth is 3. as follows:

Input feature mapping group: \( X \in R^{M \times N \times D} \) is a three-dimensional tensor, where each slice matrix \( X^d \in R^{M \times N} \) is an output feature map, \( 1 \leq p \leq P \);

Output feature mapping group: \( Y \in R^{M \times N \times p} \) is a three-dimensional tensor, where each slice matrix \( Y^p \in R^{M \times N} \) is an input feature map, \( 1 \leq p \leq P \);

Convolution kernel: \( W \in R^{U \times V \times D \times p} \) is a four-dimensional tensor, where each slice matrix \( W^{p,d} \in R^{U \times V} \) is a two-dimensional convolution kernel, \( 1 \leq d \leq D, 1 \leq p \leq P \);

In order to calculate the output feature map \( Y^p \), the convolution kernel \( W^{p,1}, W^{p,2}, \ldots, W^{p,D} \) is used to convolve the input feature map \( X^1, X^2, \ldots, X^D \), then the convolution results are added, a scalar offset \( b \) is added to obtain the net input \( Z^p \) of the convolution layer. Get the output feature map \( Y^p \) after activating the function.

\[
Z^p = W^p \otimes X + b^p = \sum_{d=1}^{D} W^{p,d} \otimes X^d + b^p
\]

\[
Y^p = f(Z^p)
\]  

\( W^p \in R^{U \times V \times D} \) is a three-dimensional convolution kernel, \( f(\cdot) \) is a non-linear activation function, usually used ReLU function. If it is desired to output multiple feature maps from the convolution layer, the calculation process needs to be repeated many times to obtain multiple output feature maps \( Y^1, Y^2, \ldots \).

In a convolutional layer with input \( X \in R^{M \times N \times D} \) and output \( Y \in R^{M \times N \times p} \). The number of filters required for each output feature map is \( D \), and an offset. Assuming the size of each filter is \( U \times V \), then the number of parameters required is \( P \times D \times (U \times V) + P \).

3.3. Pooling Layer
The pooling layer can further reduce the number of nodes and reduce the parameters in the neural network.

Two pooling methods: Max Pooling: For region \( R^d_{m,n} \), taking the maximum activity value of all neurons in the region as the representation.

\[
y^d_{m,n} = \max_{i \in R^d_{m,n}} x_i
\]

Where \( x_i \) is the activity value of each neuron in the region.

Mean Pooling: Taking the average value of all neuron activity values in the area.

\[
y^d_{m,n} = \left( |R^d_{m,n}| \right)^{-1} \sum_{i \in R^d_{m,n}} x_i
\]
Sub-sampling the input features \( (X^d) \) and region \( (M' \times N') \) to get the output feature map \( \{Y^d = \{y^d_{m,n}\}, 1 \leq m \leq M', 1 \leq n \leq N' \} \) of the pooling layer.

4. Simulation under Tensorflow

A lot of artificial intelligence framework Tensorflow and Open-cv libraries are used. Before entering the recognition model, because of the large sample data, using tensorflow-gpu to accelerate. First, get real-time video data by reading the local edge front-end camera to collect yourself The training data set. Using the Yale face database from Yale University, the ORL face database from Cambridge University, and the FERET face database from the U.S. Department of Defense to produce the training test set needed for the convolution model.

The iteration termination condition of the entire training model is set to stop training when the training recognition rate reaches 95%, and save the training model as face_overtrain.

The last is the process of face recognition. First, the obtained local video image frame needs to be grayed out, and then entered into a face recognition model after face detection. Through the analysis of the experimental results, during the training process:

4.1. Relationship Between Ideal Recognition Rate and True Recognition Rate

Take the learning rate as 0.001. At the beginning, the training recognition rate is higher than the real recognition rate. After a brief confrontation, as the number of training times continue to increase, it finally reaches the set threshold accuracy > 0.95, and the true recognition rate is higher than the training recognition rate. The training recognition rate indicates that the training model is available and effective.

4.2. The Trend of Training Recognition Rate

Taking the learning rate = 0.001, take 100 frames of pictures each time as a single complete training. The overall recognition rate trend fluctuates greatly in the early stage. When the number of training times reach a certain level, the overall trend is stable and eventually converges.

![Figure 4. Training recognition rate code](image)

![Figure 5. Loss and accuracy when learn rate = 0.001](image)
4.3. Adjust the Relationship Between Learning Rate and Training Recognition Rate, Loss Curve

When the learning rate = 0.001, loss curve attenuation is obvious and gradient descent, indicating that the learning rate is appropriate. After learning to a certain stage, the overall fluctuation is relatively stable and eventually to a stable value. When learn rate = 0.01, the overall attenuation of the loss curve is slower than 0.001, the recognition rate growth is also slower, and there are obvious fluctuations, but it also tends to a stable value, and the stable value at this time is not within the expected value range.

![Figure 6. Loss and accuracy when learn rate = 0.01](image)

When learn rate = 0.0001, the change curve of the loss curve is not obvious, and it is difficult to converge. The recognition rate accuracy does not reach the expected value. At this moment, it is expected to increase the number of training times to achieve the desired recognition rate, but the total training time consumed will increase accordingly.

![Figure 7. Loss and accuracy when learn rate = 0.0001](image)

5. Conclusion

The computational complexity of video surveillance image processing and security attacks on video equipment have become the bottlenecks of real-time control requirements and security assurance in industrial automation. The architecture of edge computing that can provide an image real-time processing and security protection is introduced. A novel cooperative training edge computing system for the image processing is proposed to realize the shortest training and processing time. The simulation of the novel system is performed under Tensorflow software and the results verify the advantages of the new system.

6. Acknowledgement

This research is supported by the National Key R&D Program of China (2018YFB0904900, 2018YFB0904905).
7. References

[1] B. Du, R. Huang, Z. Xie, J. Ma and W. Lv, KID Model-driven Things-edge-cloud Computing Paradigm for Traffic Data as a Service, IEEE Network, vol. 32, no. 1, Jan./Feb. 2018, pp. 34-41.

[2] Yuanpeng Xie, Hong Wen, Bin Wu, Yixin Jiang, Jiaxiao Meng, A Modified Hierarchical Attribute-Based Encryption Access Control Method for Mobile Cloud Computing, IEEE Transaction on Cloud Computing, vol. 7, no. 2, PP. 383 - 391, April 2019.

[3] Yuanpeng Xie, Yixin Jiang, Jinsong Wu, Hong Wen, Jiaxiao Meng, Xiaobin Guo, Aidong Xu, Zewu Guan, Three-Layers Secure Access Control for Cloud-based Smart Grids, IEEE VTC 2015-Fall, 2015.09.

[4] Yuanpeng Xie, Yixin Jiang, Runfa Liao, Hong Wen, Jiaxiao Meng, Xiaobin Guo, Aidong Xu, Zewu Guan, A Hierarchical Key Management System Applied in Cloud-based Smart Grid, IEEE ICC 2015, 2015.11.02

[5] Yuanpeng Xie, Yixin Jiang, Runfa Liao, Hong Wen, Jiaxiao Meng, Xiaobin Guo, Aidong Xu, Zewu Guan, User Privacy Protection for Cloud Computing based Smart Grid, IEEE ICC 2015, 2015.11.02

[6] Bin Li, Bincheng Jia, Wangwang Cao, et al. Application Prospect of Edge Computing in Power Demand Response Business. Power System Technology, 2018, 42 (1): 79-87.

[7] Chen Feiling. Full-digital High-definition Intelligent Video Surveillance Solution for Substations. China Public Safety, 2013: 142-145.

[8] Chen Feiling. Development Trend of Smart Grid Security and Video Surveillance System [j]. China Public Safety, 2013 (7): 114-117.

[9] Wang Yong, Ji Yili, Zhu Huihua. Analysis of the Development Trend of Smart Grid Security and Video Surveillance Systems. Communications World, 2014 (3): 85-86.

[10] Cui Zhaolei. Development and Application of Edge Computing Technology in the Security Industry. China Security, 2018 (4).

[11] Songlin Chen, Hong Wen, Jinsong Wu, Wenxin Lei, Wenjie Liu, Aidong Xu, Yixin Jiang, Internet of Things Based Smart Grids Supported by Intelligent Edge Computing, IEEE Access, vol. 7, pp. 74089 – 74102, 03 June 2019

[12] Songlin Chen, Hong Wen, Jinsong Wu, Aidong Xu, Yixin Jiang, Huanhuan Song and Yi Chen, Radio Frequency Fingerprint-Based Intelligent Mobile Edge Computing for Internet of Things Authentication. Sensors 19(16): 3610 (2019).

[13] FeiYi Xie, Aidong Xu, Yixin Jiang, Songlin Chen, Runfa Liao, Hong Wen, Edge Intelligence based Co-training of CNN. ICCSE 2019: 830-834

[14] Run-fa Liao, Hong Wen, Fei Pan, Huanhuan Song, Aidong Xu, Yixin Jiang, A Novel PHY-Authentication Method with Convolutional Neural Network, in Proc. of 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA 2019) ICAICA2019, 29-31 March 2019, Dalian, China.

[15] Songlin Chen, Hong Wen, Jinsong Wu, Jie Chen, Wenjie Liu, Lin Hu, and Yi Chen, Physical-Layer Channels Authentication for 5G via Machine Learning Algorithm, Wireless Communications and Mobile Computing, Research Article (10 pages), Article ID 6039878, Volume 2018s (2018), https://doi.org/10.1155/2018/6039878.