Cloud-Edge Collaboration Based Power IoT Scene Perception Mechanism

Sujie Shao[1](✉), Congzhang Shao[1], Cheng Zhong[2], Shaoyong Guo[1], and Pengcheng Lu[2]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
buptssj@bupt.edu.cn

[2] Xiong'an New Area Power Supply Company of State Grid Hebei Electric Power Co., Ltd., Xiong'an 071600, Hebei, China

Abstract. Fast and high-quality scene perception is an important guarantee for the efficient, stable and reliable operation of the power Internet of things, which can assist the decision-making of upper-level applications. The transmission delay of scene perception based on cloud computing is high, so it is difficult to meet the needs of real-time decision-making the mode based on edge computing is not competent for all real-time perception tasks due to the limited computing resources. For this reason, this paper proposes a scene awareness mechanism of the power Internet of things based on cloud-edge collaboration. A scene information awareness architecture based on cloud-edge collaboration is constructed, and a scene information processing flow that distinguishes dynamic instances, static instances and general instances is designed to support local scene information edge awareness and global scene cloud synthesis. Focusing on the construction of high-precision neural network recognition model of high-frequency dynamic examples, using the idea of transfer learning, a neural network model training framework based on cloud-edge collaboration is designed. Simulation results show that the scene perception mechanism proposed in this paper can effectively reduce the perception processing delay and model training time on the basis of accurately perceiving the scene, and improve the adaptability of the perception model to high dynamic scenes.

Keywords: Cloud-edge collaboration · Power IoT · Scene perception · Transfer learning · Edge intelligence

1 Introduction

In recent years, with the vigorous development of new-generation information technologies such as 5G, cloud computing, and artificial intelligence, power IoT technology

This work is supported by the Science and Technology Project of State Grid Corpora-tion of China: Research and application of key technologies for wireless communica-tion network coverage of urban power underground pipe gallery (Grant No. 5700-202113189A-0-0-00).

F. Fang and F. Shu (Eds.): GameNets 2022, LNICST 457, pp. 100–117, 2022.
https://doi.org/10.1007/978-3-031-23141-4_8

has received extensive attention and development. Scene information perception is an important cornerstone to realize the technology vision of smart power IoT technology, which involves the collection and processing of video image data and various sensor data in power IoT scenarios. In actual scenarios, all kinds of information in the Internet of Things are changing rapidly, which requires real-time and accurate perception.

Cloud computing mode is a method of power IoT scene information perception [1], which meets the needs of scene information perception to a certain extent. However, the model based on cloud computing has great limitations, which are mainly reflected in the following aspects:

- Real-time. In the actual scenario of the Internet of Things, the sensing information changed rapidly. In order to make the upper-layer applications and personnel make smooth decisions, the scene information needs to be sensed in time. However, uploading the scene information to the cloud server for processing and then returning it will result in excessive delay.
- Resource utilization. In the power IoT scenario, there is a lot of redundant and repeated information, uploading unprocessed raw data directly to the cloud computing center for processing will occupy a large amount of network bandwidth, resulting in a decrease in the efficiency of network resource utilization.
- Accuracy. The characteristics of instances in different sub-scenarios of the Internet of Things are different, to generate perception models for specific scenarios was difficult to the cloud computing mode.

As an emerging distributed computing framework [2], the edge computing model was proposed to solve the limitations of the cloud computing model, and its idea is to process data at the edge of the network. With lower processing delay and can reduce the load of the network, the edge computing mode solved the limitations of the cloud computing mode to a certain extent. However, servers deployed at the edge of the network often have limited computing resources, when the computing power required for computing tasks in the scene exceeds the server or faces some more complex computing tasks, it will not be able to meet the demands of scene information processing.

In recent years, cloud-edge collaboration has begun to receive extensive attention from researchers. It combines the advantages of cloud computing and edge computing. The main idea is to complete data preprocessing and some computing tasks on edge servers, and complete computing tasks that edge servers cannot perform on cloud servers, finally complete computing tasks quickly and with high quality.

This paper proposes a cloud-edge collaboration based power IoT scene perception mechanism. According to different types of information in the scene, a scene-aware architecture that distinguishes instance types is designed. For the construction of high-precision recognition models for high-frequency changing dynamic instances, a cloud-edge collaborative neural network model training framework is proposed, and finally of power IoT scene information.

The main contributions of this paper are as follows:

- Proposes a cloud-edge collaborative scene information perception architecture for the Internet of Things. This architecture divides a wide variety of information in the power IoT scene into different types of instances, and then designs corresponding perception algorithms for different instance types to complete perception fusion at the edge, finally synthesize the global scene in the cloud server, reduced processing latency while ensuring the perception quality.

– Proposes a neural network model training framework based on cloud-edge collaboration is, which solves the problem of building a high-precision recognition model for high-frequency changing dynamic instances. The framework adopts the idea of migration learning, assists the training of the neural network model on the edge server with the help of the cloud server, and shares some parameters of each edge neural network model through the cloud server during the training process, which reduces the burden on the edge server and improves the convergence speed of the model in highly dynamic scenarios of the neural network model.

2 Related Work

Many literatures have carried out research on power IoT or IoT scene perception. T Kobzan et al. [10] proposed an IoT architecture scheme based on smart factory network, which transforms the current traditional network infrastructure into a more flexible SDN (Software Defined Network)-based infrastructure to provide context awareness for IoT and other services. P Hořejší et al. [11] applied ICT-enhanced solutions to smart factories and developed a new smart solution that reduced the workload of employees and improved the efficiency of factories. Wang Yaonan et al. [12] proposed a machine vision perception control system scheme based on cloud computing, which solved the contradiction between the high real-time requirements of visual control applications and the extremely high computational complexity to a certain extent. Zhang Ansi et al. [13] studied the perception method, equipment data fusion and analysis technology of digital workshop manufacturing field equipment data, which provided a guarantee for the realization of digital workshop equipment safety, stability, high efficiency and green production. The above scene perception work is based on cloud computing, which meets the needs of scene perception to a certain extent, but it is still difficult to meet the needs of actual scenes when dealing with tasks with high real-time performance.

Since edge computing was proposed, it has been widely used in various fields, especially in power IoT scenarios. W Shi et al. [2] introduced the concept of edge computing and summarized its technical challenges and application prospects. Z Zhou et al. [14] outline key techniques for deep learning models for network edge training/inference, and discuss future research opportunities for edge intelligence. Jung Woo-Kyun et al. [15] proposed an application measure of IoT smart sensors, smart IoT, and small data processing combined with edge computing, which provided a reference for SMEs to apply IoT. Z Song et al. [16] implemented an edge-based sensing system, including deep learning-based image recognition, data model retraining, etc., which effectively utilized the resources available at the edge and greatly reduced network traffic. The above-mentioned mode based on edge computing can better solve the problem of real-time task response, but limited by the computing power of edge servers, it is still difficult to perform more complex computing tasks such as deep neural network training.

In recent years, the cloud-edge collaboration model has been proposed to meet the needs of actual scenarios. This model combines the advantages of the cloud computing model and the edge computing model, and can ensure real-time response while being competent for computing tasks. Chen Yuping [17] analyzed the realization principles and research ideas of technologies such as resource collaboration, intelligent collaboration, and application management collaboration, and pointed out the current challenges and future development directions of cloud-edge collaboration. At present, some works have applied the cloud-edge collaboration model to practical scenarios. Zhao Yu et al. [18] proposed a target detection framework combining edge computing and federated learning, which improved the detection accuracy by 18% in video surveillance scenarios and effectively reduced model training time. T Jing et al. [19] proposed a cloud-edge collaboration architecture suitable for the Internet of Things, which meets the complex network data computing and real-time analysis requirements of local data in the Internet of Things. C Ding et al. [9] proposed a cloud-edge collaboration cognitive service framework, which uses the cloud server to accelerate the training of lightweight neural networks in the edge server, improved the training speed of the model and the accuracy of cognitive services. Xu Mengwei et al. [20] proposed a new type of machine learning model training mode based on cloud-based private data of mobile terminals, which effect exceeds the traditional centralized/federated training mode, and greatly reduces the computational and energy overhead for model training. However, there is no mature research on the application of cloud-edge collaboration mode to power IoT scene perception in the existing work, and the cloud-edge collaboration mechanism needs to be further improved.

3 Cloud-edge Collaboration Based Scene Information Perception Architecture

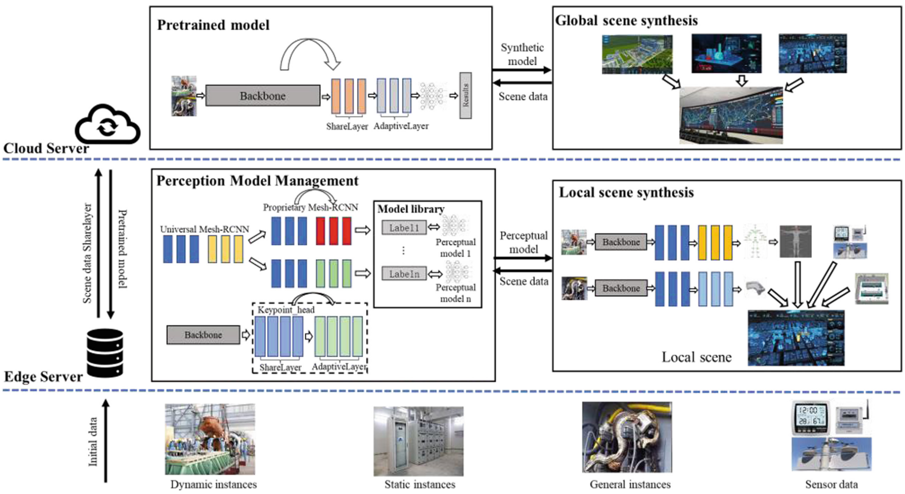


Fig. 1. Cloud-edge collaboration based scene information perception architecture

This paper designs a cloud-edge collaboration based scene information perception architecture, as shown in Fig. 1. The information involved in the power IoT scene includes sensor data (voltage, temperature, humidity, etc.), image data (personnel, inspection robots, faulty equipment, etc.). This paper divides the scene information into static instances, dynamic instances, general instances and sensor data four types. After the terminal device collects the scene information, the edge server obtains the local scene through three stages of instance extraction, instance perception and scene fusion, and then each edge server uploads the synthesized local scene to the cloud server, and finally the cloud server synthesizes the global scene, detailed process is as follows.

3.1 Instance Extraction

The instance extraction stage completes the classification of scene information. In the power IoT scene, the initial scene information includes two categories of sensor data and image data, and the local scene perception process is shown in Fig. 2.

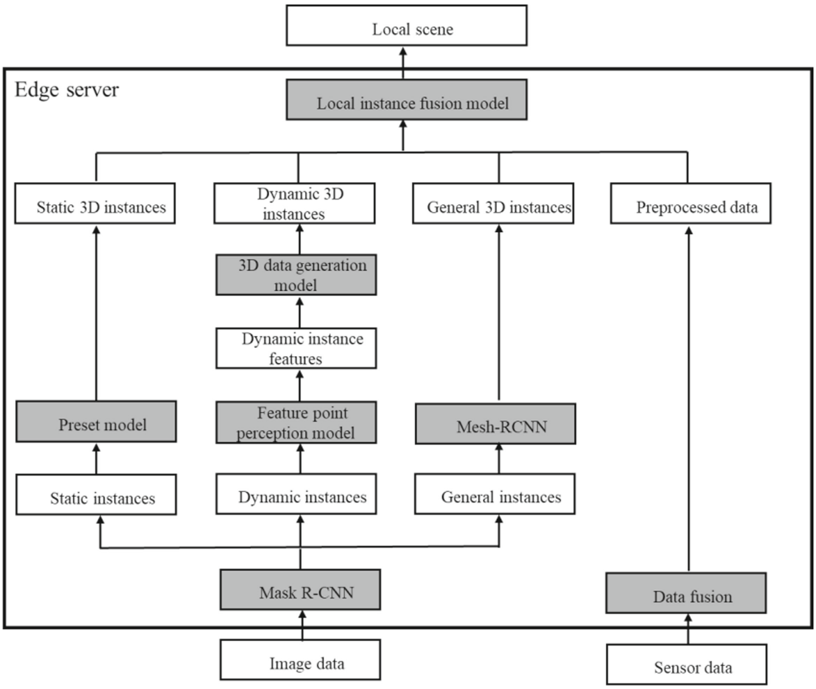


Fig. 2. Local scene perception process

In the Internet of Things scenario, there are many kinds of sensor data, which need to be fused to facilitate use. In this paper, the fusion analysis method based on small sample learning is used to fuse the sensor data [13]. Small sample learning can be divided into one-shot K-way and N-shot K-way. The difference between the two is that the number of samples in the K classes to be merged is different: N-shot K-way has N samples in each class. Samples, while one-shot K-way has only one sample per class. Due to the variety of data formats in the power IoT scenario, it is difficult to summarize the characteristics of a certain type of data with a single sample. Therefore, this paper uses the N-shot K-way algorithm to fuse the sensor data. The specific algorithm is as follows:

For the sensing data, first construct K support set classes, each with N samples $(S_1, \ldots, S_N)$, the purpose of the sensing algorithm is to determine which support set class the collected sensor data should belong to, and the optimization goal is as follows:

$$C\big(\hat{x}, (S_1, \ldots, S_N)\big) = \arg\max_i \left(\sum_{n=1}^{N} P\big(\hat{x}, x_{i_n}\big) \right) \tag{1}$$

where $x_{i_n} \in S_n$, $i \in K$. Through the N-shot K-way few-shot learning process, the sensory data in the scene are fused into K classes.

The image data is obtained by the image acquisition device, and contains a variety of instance types. In this paper, the image data is divided into three categories: static instances, dynamic instances and general instances. Mask-RCNN is one of the commonly used models for object detection, which can detect objects in images and generate masks. This paper uses the Mask-RCNN model to extract and classify the instances, and then use the corresponding perception algorithms to complete the perception processing. 3.2 introduces the detailed perception algorithms of the three instances.

3.2 Instance Perception

Static Instances. Static instances are objects with relatively fixed characteristics in power IoT scenarios, including warehouses, generator sets, and underground pipe galleries. The scene information perception architecture proposed in this paper deploys the 3D model of static instances in the edge server in advance. After the instance extraction stage Mask-RCNN identifies the static instances in the scene, it directly calls the preset 3D model to participate in the fusion of local scenes.

Dynamic Instances. Dynamic instances are objects whose characteristics change in real time in power IoT scenarios, such as workers and inspection robots. For this type of instance, this paper uses the method of first detecting key points, and then synthesizing the model based on the parameters of the key points to generate 3D models. Taking the generation of a 3D model of the human body as an example, the keypoint_head branch of Mask RCNN is used to detect the parameters of the key points of the human body, and then the detected parameters are input into the SMPL parameterized human body model to synthesize the 3D model of the human body to complete the perception of the human body instance. In order to further improve the perception accuracy of dynamic instances, this paper will also train the dynamic instance perception model in real time based on the collected scene information. The detailed training method will be introduced in Sect. 4.

General Instances. A general example is abnormal information in power IoT scenarios, such as illegal intrusions, faulty devices, etc., which are characterized by unpredictable characteristics. The 3D model of this type of instance can only be directly perceived from the image information. In this paper, Mesh-RCNN is used to perceive general instances.

Mesh-RCNN is improved from Mask-RCNN. It is a model after adding mesh branches on the basis of Mask-RCNN, which can directly generate 3D models from two-dimensional image data. After Mask-RCNN identifies general instances in the initial image, the sub-images are segmented through masks, and then Mesh-RCNN is used to generate mesh 3D data of instances based on the sub-images. In actual scenarios, general instances with similar characteristics may appear multiple times in a period of time. In order to improve the perception accuracy of general instances, this paper trains the corresponding Mesh-RCNN models for different types of instances.

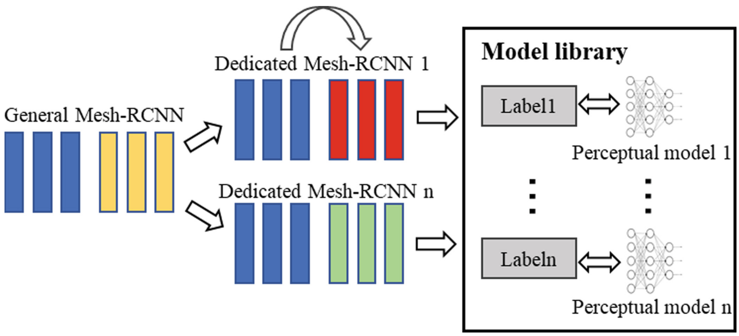


Fig. 3. Mesh-RCNN training differentiation process

Figure 3 shows the training differentiation process of Mesh-RCNN. This process is mainly completed on the edge server. First, the Mask-RCNN is trained based on the labeled data, so that it can detect several specific types of general instances, and label labels for different types, and then train the corresponding Mesh-RCNN for each label.

Under the initial conditions, the Mesh-RCNN model corresponding to each label is initialized as a general Mesh-RCNN model pre-trained on the cloud server, and then the cloud-edge collaborative perception architecture begins to perceive general instances in the scene information and collect general instance data in real time. When the instance data corresponding to a certain label is accumulated to a certain amount, the mesh label of the instance data will be obtained through manual labeling, and the Mesh-RCNN model corresponding to the label will be trained. When the general instance corresponding to a label is detected again, the Mesh--CNN model that has been specifically trained will be called.

3.3 Scene Fusion

After the instance perception phase is completed, the 3D models of static instances, dynamic instances, and general instances will be fused at the edge server, and then the sensor data will be matched to the 3D scene, therefore obtains the local scene. The merged local scene will retain two copies: one is saved on the edge server to provide real-time services to users, and the other is uploaded to the cloud server, where the global scene is synthesized.

Denote the scene initial data as Ori, sensor data as Sen, dynamic instance as Dym, static instance as Sta, general instance as Gen, local scene perception result as Loc, global scene perception result as Gol; instance extraction process as F, instance perception The process is P and the scene fusion process is B. From the perspective of data flow, the perception process of local scenes can be expressed as the following formula:

$$Ori \rightarrow^{F} Sen\,,\; Dym\,,\; Sta\,,\; Gen \overset{(P_{sm}+P_{D_{mm}}+P_{com}+P_{sen})+B}{\longrightarrow} Loc \tag{2}$$

The synthesis process of the global scene can be expressed as the following formula:

$$Loc_1, Loc_1, \ldots Loc_n \rightarrow^{C} \text{Gol} \tag{3}$$

In the above-mentioned scene information perception architecture based on cloud-edge collaboration, after the perception model is obtained, the scene information will be directly perceived by the edge server, and only the generation and training process of the perception model involves the collaborative work of the cloud server and the edge server. In the generation and training of the instance-aware model, the sensing data and static data-aware models are fixed models that are preset on the edge server, although the general instance needs to download the perception model from the cloud server, the training process on the edge server is independent, none of the three involve the collaboration of cloud servers and edge servers. In fact, only the training of dynamic instance-aware models involves cloud-edge collaboration, and the part where the scene perception speed and quality can be optimized is also here. Therefore, this paper will focus on the optimization of dynamic instance-aware model training methods, and the details will be introduced in Sect. 4.

4 Neural Network Model Training Framework Based on Cloud-edge Collaboration

A typical task of dynamic instance perception in power IoT scenarios is the detection of human pose key points, this paper takes this as an example to study the dynamic instance perception algorithm. To this end, a neural network model training framework based on cloud-edge collaboration is proposed, as is shown in Fig. 4. The core idea is to train the neural network perception model for specific scenarios on the edge server, and pre-train the neural network model with the help of the cloud server to reduce the load of edge server.

Figure 4 shows the cloud-edge collaborative training framework proposed in this paper. A deep neural network model, called CloudNet, is deployed on the cloud server; a lightweight neural network model, called EdgeNet, is deployed on the edge server, which is obtained by reducing some of the backbone network layers by CloudNet. The EdgeNet backbone network parameters are the same as those of the corresponding remainder in CloudNet. In real-time training, in order to obtain a human pose perception model for a certain scene, this paper trains EdgeNet based on the scene information collected in real time.

The paper [22] pointed out that the higher layers of the neural network can extract the general features of the data. In the human pose keypoint recognition task, the features of

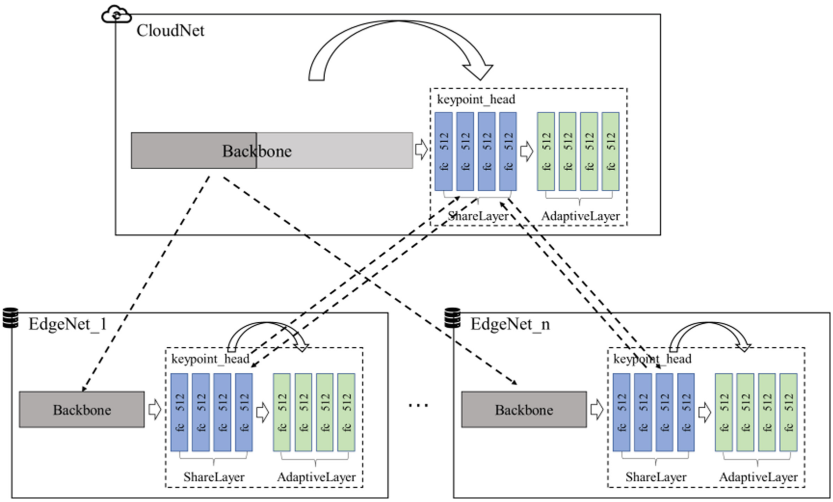


Fig. 4. Neural network model training framework based on cloud-edge collaboration

human pose overlap in each sub-scenario. For this reason, this paper sets the high m layer of the keypoint_head part of the ROI head of EdgeNet as ShareLayer, which is used to extract the common features of the instances in each sub-scenario, and its parameters are shared by each EdgeNet; meanwhile set the rest of the keypoint_head as AdaptiveLayer, which is different in each EdgeNet to adapt to the different characteristics of instances in each sub-scenario.

Regarding the acquisition of labels required for EdgeNet training, this paper uses CloudNet to automatically generate and obtain labels required for training. This is because CloudNet has a large number of neural network layers and can obtain higher recognition accuracy. CloudNet is distributed to the edge server during the initialization phase, which is recorded as Label-builder. Compared with the traditional manual labeling method, automatic label generation with CloudNet can reduce the time required for label creation and speed up model training and updating.

The training process of EdgeNet can be divided into three stages: EdgeNet initialization, EdgeNet learning, and Sharelayer update. The detailed process is as follows:

4.1 EdgeNet Initialization

In the EdgeNet initialization phase, a large amount of data is used to pre-train CloudNet on the cloud server, and then some of its parameters are sent to the edge server to assist EdgeNet training.

Write W_c as the parameter of CloudNet, W_{c-b} v is the parameter of the part shared with EdgeNet in the CloudNet backbone network, W_{c-rs} is the parameter of ShareLayer in the ROI head of CloudNet; W_c is the parameter of EdgeNet, W_{c-b} is the parameter of the EdgeNet backbone network, W_{e-r} is the EdgeNet ROI head The parameters of W_{e-rs} and W_{e-ra} are the parameters of ShareLayer and AdaptiveLayer in the ROI head of EdgeNet respectively, and N is the data volume of the training set. Given a training set $\{x_i, y_i\}_{i=1}^{N}$, EdgeNet will be trained to optimize the following loss function:

$$f(W_e) = \frac{1}{N}\sum_{i=1}^{N}\theta(y_i, \sigma(f(x_i; W_e))) \tag{4}$$

After the edge server receives the CloudNet sent by the cloud server, it first separates W_{c-b} and W_{c-rs} in CloudNet as the parameters of W_{e-b} and W_{e-rs} in EdgeNet, then initializes W_{e-ra} to a random value, and then combines the three parts of, W_{e-b}, W_{e-rs} and W_{e-ra} to form W_e, that is, the parameters of EdgeNet; then optimize W_e by formula (4) to complete the training of EdgeNet. This paper uses CloudNet to assist EdgeNet training, freezes the parameters of the backbone network in EdgeNet during the training process, and only fine-tunes the parameters of the ROI head. Use W_e^* to represent the updated value of W_e, and the connector $\cup$ to connect the two parameter sets. The detailed steps of EdgeNet initialization are shown in Algorithm 1.

input : $\{x_i, y_i\}_{i=1}^{N}$, W_c, CloudNet
output: EdgeNet, Label-builder
1. The cloud server sends CloudNet (parameter value W_c) to the edge server;
2. The edge server processes CloudNet to obtain three copies of W_c, W_{c-b}, W_{c-rs};
3. Edge servers build EdgeNet;
4. Initialize W_{e-b}, W_{e-rs}, W_{e-ra} of EdgeNet to W_{c-b}, W_{c-rs}, random values respectively;
5. $W_e = W_{e-b} \cup W_{e-rs} \cup W_{e-ra}$;
6. $W_e^* \leftarrow arg\min_{W_e} f_e(W_e)$;
7. Return EdgeNet as W_e^* and Label-builder as W_c

Algotithm 1. EdgeNet initialization

4.2 EdgeNet Learning

In the EdgeNet initialization phase, EdgeNet completed the initial training with the assistance of CloudNet. In actual scenarios, the edge server will continuously receive the instance information collected by the terminal from the scene. The paper [21] pointed out that real-time information is very important to improve the quality of the model. To this end, this paper further trains EdgeNet using real-time collected instance information. However, the initially collected instance information has no labels. To solve this problem, this paper draws on the method in [9] and uses the Label-builder obtained in the EdgeNet initialization phase to generate the labels required for training. After the edge server receives the instance information, it first uses the EdgeNet to obtain the perception results. Whenever the number of newly collected instances accumulates to M, the Label-builder will be used to generate the labels of the instances to retrain the EdgeNet.

$$f\left(W_e^*\right) = \frac{1}{M}\sum_{i=1}^{M}\theta\left(y_i, \sigma\left(f\left(x_i; W_e^*\right)\right)\right) \tag{5}$$

Since Label-builder can automatically generate labels in real time, EdgeNet learning is continuous, and its detailed process is shown in Algorithm 2.

input : EdgeNet, W_e^*, instance$\{x_i\}_{i=1}^{M}$, Label-builder
output: Updated EdgeNet
1. Edge server uses Label-builder to get label $\{y_i\}_{i=1}^{M}$ of instance $\{x_i\}_{i=1}^{M}$;
2. $W_e^{**} \leftarrow arg\min_{W_e^*} f_e(W_e^*)$, W_e^{**} is the updated value of W_e^*;
3. Returns the value of EdgeNet as W_e^{**}

Algorithm 2. EdgeNet learning

4.3 Sharelayer Update

As EdgeNet learning continues, EdgeNet is constantly adapting to changing scene information. To further accelerate the convergence speed of EdgeNet learning and extract common features of instances in different scenes, this paper sets a shared layer ShareLayer(W_{e-rs}) in the ROI head of EdgeNet (as described in the beginning of Sect. 4). In real-time training, each edge service will extract the parameters of Sharelayer in EdgeNet and upload it to the cloud server after retraining the EdgeNet on it for a certain number of times. After collecting all the parameters of the Sharelayer in EdgeNet, the cloud server will use the Fedavg [23] algorithm to summarize the gradients generated by the Sharelayer in each EdgeNet during training, update the Sharelayer with the aggregated gradient and send it to each edge server.

Write W_{c-rs}^* as the updated parameter of Sharelayer. The detailed update process of Sharelayer is shown in Algorithm 3:

input : N EdgeNets, training batch size n_i, W_{e-rs_i}, W_{e_i}, $i=1,2,\ldots,N$, Sharelayer copy on cloud server, W_{c-rs}

output: Updated ShareLayer

1. Each edge server separates the Sharelayer parameter (ie. W_{e-rs_i}) from EdgeNet and uploads it to the cloud server;
2. Calculate the gradient generated by each EdgeNet $g_i = W_{e-rs_i} - W_{c-rs}$, $i=1,2,\ldots,N$;
3. Calculate the weighted average gradient sum $g = \sum_{i=1}^{N} \frac{n_i}{n} g_i$, where $n = \sum_{i=1}^{N} n_i$;
4. $W^*_{c-rs} \leftarrow W_{c-rs} - \eta(g + \partial(W_{c-s}))$;
5. Returns the updated ShareLayer as W^*_{c-rs}, and sends it to each edge server;
6. Edge servers update their ShareLayer in EdgeNet to W^*_{c-rs}

Algorithm3. Sharelayer update

After the edge server sets the ShareLayer in the respective EdgeNet to W^*_{c-rs}, , it will call Algorithm 2 again to retrain the EdgeNet, which realizes further optimization of the EdgeNet. Sharelayer update and EdgeNet learning alternate in real time, but Sharelayer update will be started after a certain round of EdgeNet learning, until Sharelayer generates a more obvious gradient, so the calling frequency is slightly lower (Fig. 5).

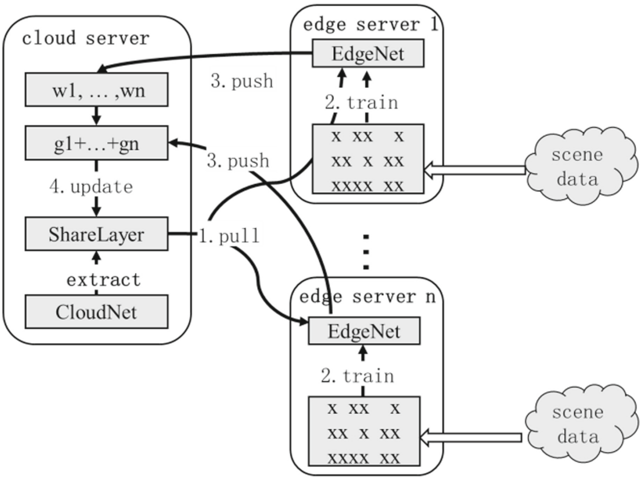


Fig. 5. Synchronous blocking parameter summary scheme

In general, the cloud-edge collaborative training framework proposed in this paper has two advantages: First, the framework can provide fast response services. This is because the edge server is closer to the user, and the number of EdgeNet network layers used for perceptual computing is less, and the computing time is short; secondly, the framework can better adapt to changes in dynamic instance characteristics of the scene.

This is because the framework trains EdgeNet based on real-time scene instance information, and introduces Label-Builder and ShareLayer so that training updates can be continuously performed in real-time and adapt to changes in scene information more quickly.

5 Simulation

5.1 Settings

Human posture key point detection is a key and representative task in the scene perception of power IoT. This paper takes this task as an example to verify the effect of the proposed cloud-edge collaborative power IoT scene perception mechanism.

The CPU model of the server used in the simulation experiment is Xeon E5–2600 v2@2.1 GHz, and the running memory size is 62.5 GB. The human pose keypoint detection framework improves from the Dectectron2 framework (based on the Mask RCNN model) commonly used in object detection. CloudNet and EdgeNet are improved from the keypoint_rcnn_R_50_FPN_3x model provided by Dectectron2. The detailed network structure is shown in Fig. 6 Among them, the original CloudNet model has the same structure. The backbone network of EdgeNet removes the res4 and res5 parts of the model, and the rest is the same as the original model. ShareLayer is the upper 4 layers of Mask RCNN's FPN, Box Head, Box_predictor and keypoint_head, and AdaptiveLayer is the lower 4 layers of keypoint_head. Simulate real-time scene data with images from COCO2017 annotated with human pose keypoints.

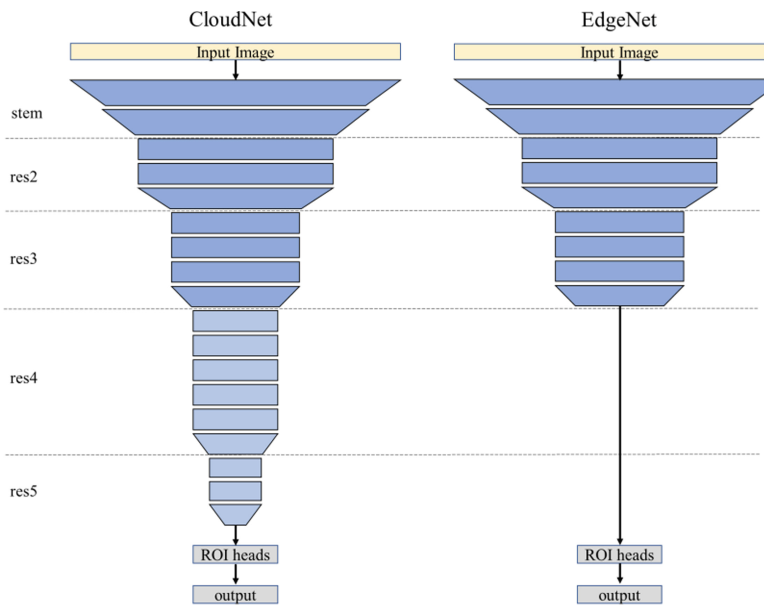


Fig. 6. Structure of EdgeNet and CloudNet (based on keypoint_rcnn_R_50_FPN_3x)

5.2 Results

Figure 7 shows the comparison of perception and transmission delays for the human pose keypoint detection task using the two mechanisms. Compared with the cloud computing model, the cloud-edge collaboration mechanism proposed in this paper reduces the perception delay by 39.8%. This is because EdgeNet is obtained by removing the backbone network res4 and res5 from the keypoint_rcnn_R_50_FPN_3x model, and the number of network layers is less; the transmission delay is reduced increased by 26.9% because edge servers are closer to users. The total processing delay obtained by the combined calculation of the two is reduced by 38.6%, which indicates that the collaborative perception architecture proposed in this paper can effectively improve the perception speed of scene information.

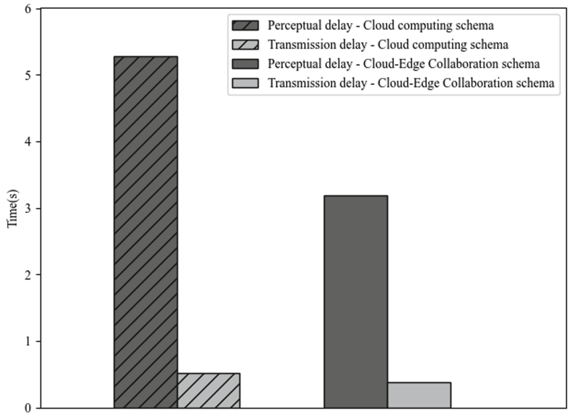
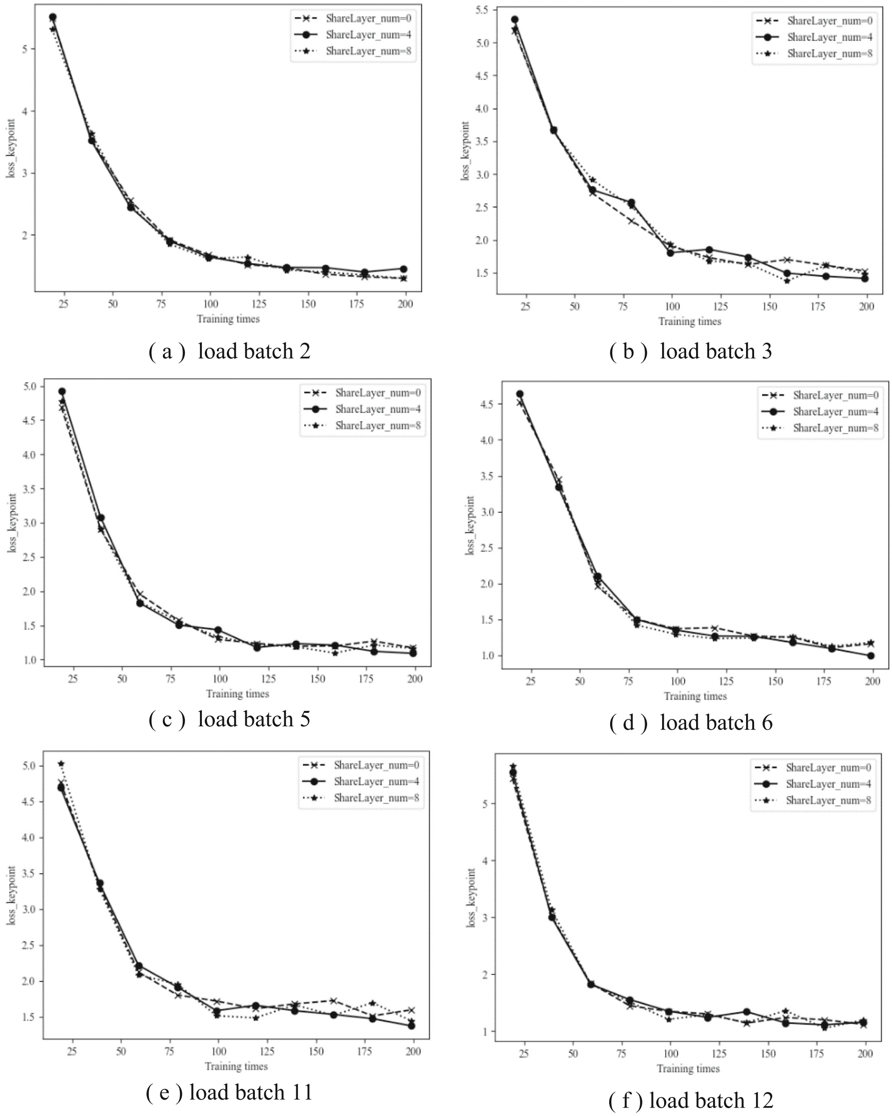


Fig. 7. Comparison of perception delay and transmission delay between the two schemas

In order to investigate the improvement of the model training effect of the proposed cloud-edge collaborative training framework, this paper simulates the process of collecting data and training EdgeNet in the actual scene. The specific settings are as follows: Under the initial conditions, the pre-trained CloudNet is deployed in the cloud server, and the training starts. At first, Algorithm 1 is called to initialize EdgeNet based on the preset 100 image data. Then simulate the process of real-time training EdgeNet in the actual scene, take 100 images as a batch, load 12 batches of pictures, and call algorithm 2 to retrain EdgeNet after each batch is loaded, and each edge server is loaded separately. After three batches of images, Algorithm 3 will be called to summarize the gradients generated by ShareLayer in the first two EdgeNet training sessions and retrain EdgeNet based on the updated gradients. That is, in the loading of every three batches of images, the first two calls to Algorithm 2 retrain EdgeNet, and the last call to Algorithm 3 to retrain EdgeNet.

Loss_keypoint is the error between the output value of Mask-RCNN on the training set and the real value. This paper uses this as a reference to measure the adaptability of the model to the new information features in the scene. Figure 8 compares the change of loss_keypoint with the number of iterations in the training process when the loading batches are 2, 3, 5, 6, 11, and 12, and the ShareLayer in EdgeNet is set to be the upper 0, 4, and 8 layers of keypoint_head, respectively. Among them, ShareLayer is the first 4 layers of keypoint_head, which is the training method proposed in this paper. ShareLayer is the first 0 layers of keypoint_head, that is, the classic transfer learning method without ShareLayer. ShareLayer is the first 8 layers of keypoint_head, that is, the parameters of EdgeNet on each edge server are completely shared. When the loading



(a) load batch 2 (b) load batch 3 (c) load batch 5 (d) load batch 6 (e) load batch 11 (f) load batch 12

Fig. 8. The change of loss_keypoint with the number of training when the loading batches are 2, 3, 5, 6, 11, 12 and shareLayer is set as the first 0, 4, and 8 layers of keypoint_head.

batch is 2, the EdgeNet of the three structures starts to converge when the number of iterative training is about 130. At this time, the algorithm 3 synchronization ShareLayer is not enabled, and the final convergence value of loss_keypoint is also similar; when the loading batch is 3, it is enabled for the first time. Algorithm 3 synchronizes the Sharelayers of each EdgeNet. After 200 iterations of training, setting ShareLayer_num = 4 converges to the smallest loss_keypoint value, which is 4.856% lower than setting ShareLayer_num = 8, and 7.428% lower than setting ShareLayer_num = 0. When loading batches of 5, 6, 11, and 12, setting ShareLayer_num = 4 also obtained the smallest loss_keypoint. This is because, on the one hand, the setting of ShareLayer can extract the common features of each sub-scenario [22] and is jointly trained by each EdgeNet, so the convergence speed is faster; on the other hand, the 4-layer AdaptiveLayer trained by each EdgeNet allows EdgeNet to learn different the unique characteristics of sub-scene enhance the adaptability of the model to different scenes, so it achieves better results than setting ShareLayer to 8. It is worth noting that the data set of the simulated collection data and training process in this paper is randomly divided from COCO2017, and the distribution of information features of each sub-scenario is relatively uniform. If the characteristics of the datasets used in each EdgeNet training are quite different, the collaborative training algorithm proposed in this paper will have a more prominent performance.

Figure 9 shows the comparison of the training time of ClouNet, EdgeNet0, and EdgeNet. EdgeNet0 has the same network structure as EdgeNet, but the parameters are randomly initialized during training, and all parameters are adjusted in the process. As can be seen from the figure, the training time of EdgeNet is reduced by 37.9% compared to CloudNet, because the number of layers of the EdgeNet backbone network is small; the training time of EdgeNet0 is reduced by 23.2%, this is because EdgeNet only adjusts the parameters of the ROI head part during the training process, which reduces the amount of computation. This shows that the neural network model training framework based on cloud-edge collaboration proposed in this paper can effectively reduce the training time of the model.

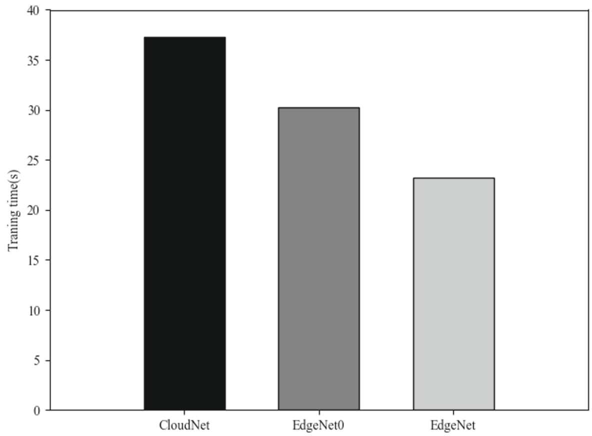


Fig. 9. ClouNet, EdgeNet0, EdgeNet training time comparison

6 Conclusion

In this paper, a cloud-edge collaboration perception mechanism is proposed for the power IoT scene perception problem. The various types of data in the power IoT scene are summarized into four categories: sensor data, static instances, dynamic instances, and general instances, and corresponding algorithms are designed to complete the perception. Dynamic instance perception is a key link in scene perception. This paper takes the task of human pose key point detection as an example to study and design a cloud-edge collaborative neural network model training framework. EdgeNet is trained based on real-time scene information to adapt to the dynamic changes of instance features in the scene, and during the training process, CloudNet is assisted and the ShareLayer shared by each EdgeNet is introduced to improve the model training effect. The simulation results of the prototype system show that the cloud-edge collaborative perception mechanism proposed in this paper can realize the rapid perception of power IoT scene information while ensuring the perception quality. The proposed cloud-edge collaborative neural network model training framework can effectively speed up the convergence of model training and adaptability to new scenarios.

Limited by the experimental conditions, this paper does not verify the proposed cloud-edge collaborative perception mechanism in actual scenarios, and the perception algorithms for other types of instances other than dynamic instances also need to be verified. In addition, there is also a lack of a dynamic allocation mechanism for cloud-side server computing resources. Future work will further improve the cloud-edge collaboration mechanism, and test and verify the effect of the mechanism in actual scenarios.

References

1. Chen, Y.: Research on 3D scene perception technology of mobile robot based on convolutional neural networks [Master's Thesis]. Harbin Institute of Technology, Harbin, 2018(in Chinese with English abstract)
2. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016). https://doi.org/10.1109/JIOT.2016.2579198
3. Shaoguang, Y., Bingyi, F., Zengsen, W., Guo Chengyu, Q., Xiuchao.: China research on key technologies and applications of cloud-edge collaboration. Inf. Commun. Technol. **14**(04), 31–36 (2020). (in Chinese with English abstract)
4. Wu, G., et al. MECCAS: collaborative storage algorithm based on alternating direction method of multipliers on mobile edge cloud. In: 2017 IEEE International Conference on Edge Computing (EDGE) Honolulu, pp. 40–46 (2017)
5. Li, Y., Xu, L.: The service computational resource management strategy based on edge-cloud collaboration. In: 10th International Conference on Software Engineering and Service Science (ICSESS) Beijing, pp. 400–404 (2019)
6. Meng, C.: Research on Content Placement and Cache-aware Scheduling for Streaming Media in Cooperative Cloud-Edge Environment[Master's Thesis]. Wuhan University of Technology, Wuhan (2019). (in Chinese with English abstract)
7. Zhou, C., Lin, Z., Du, C., Wang, Z., Li, F.: Research on key technologies of tunnel robot based on cloud edge collaboratio. In: 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT) Jeju, pp. 661–666 (2020)

8. Zhang, Y., Wang, X., He, J., Xu, Y., Zhang, F., Luo, Y.: A transfer learning-based high impedance fault detection method under a cloud-edge collaboration framework. IEEE Access. **8**, 165099–165110 (2020)
9. Ding, C., Zhou, A., Liu, Y., Chang, R., Hsu, C., Wang, S.: A cloud-edge collaboration framework for cognitive service. IEEE Transactions on Cloud Computing **10**, 1489–1499 (2020)
10. Kobzan, T., Heymann, S., Schriegel, S., Jasperneite, J.: Utilizing SDN infrastructure to provide smart services from the factory to the cloud. In: Proceedings of the 15th IEEE International Workshop on Factory Communication Systems (WFCS), Sundsvall, pp. 1–4 (2019)
11. Hořejší, P., Novikov, K., Šimon, M.: A smart factory in a smart city: virtual and augmented reality in a smart assembly line. IEEE Access. **8**, 94330–94340 (2020)
12. Yaonan, W.A.N.G., Tiejian, C.H.E.N.: Key technologies of machine vision perception and control for smart factory. ZTE TECHNOL. J. **22**(05), 26–30 (2016). (in Chinese with English abstract)
13. Zhang, A.: Research on data perception, fusion and visualization in digital workshop[PhD thesis]. Guizhou University, Guiyang, 2019(in Chinese with English abstract)
14. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., Zhang, J.: Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc. IEEE **107**(8), 1738–1762 (2019)
15. Jung, W.-K., et al.: Appropriate smart factory for smes: concept, application and perspective. Int. J. Precis. Eng. Manuf. **22**(1), 201–215 (2020). https://doi.org/10.1007/s12541-020-00445-2
16. Song, Z., Cheng, J., Chauhan, A., Tilevich, E.: Pushing participatory sensing further to the edge. In: 2019 IEEE International Conference on Edge Computing (EDGE), Paris, pp. 24–26 (2019)
17. Chen, Y., Liu, W., Cheng, H.: Survey of cloud-edge collaboration. Comput. Sci. **48**(03), 259–268 (2021). (in Chinese with English abstract)
18. Yu, Z., Jie, Y., Miao, L., Jinlong, S., Guan, G.: Federated learning based intelligent edge computing technique for video surveillance. J. Commun. **41**(10), 109–115 (2020). (in Chinese with English abstract)
19. Jing, T., shen, J., Jia, T., Yutong, J., Ning, Z.: Application of cloud edge collaboration architecture in power IoT. In: 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, pp. 18–22 (2020)
20. Xu, M., Liu, Y., Huang, K., Liu, X., Huang, G.: Autonomous learning system towards mobile intelligence. J. Softw. **31**(10), 3004–3018 (2020) (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4091.htm
21. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, pp. 843–852 (2017)
22. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks?. In: 2014 Neural Information Processing Systems (NIPS), Montreal, pp. 3320–3328 (2014)
23. McMahan, H., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). Seattle, US (2017)