Cloud-Edge Fusion Based Abnormal Object Detection of Power Transmission Lines Using Incremental Learning

SHUHUA ZHANG$^{1,2}$, JIYE WANG$^2$, JIE TONG$^2$, JUN ZHANG$^2$, AND MINGHAO ZHANG$^2$

$^1$School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China
$^2$China Electric Power Research Institute Company Ltd., Beijing 100192, China

Corresponding author: Jiye Wang (wangjiye@epri.sgcc.com.cn)

This work was supported by the Basic Prospective Project of State Grid Corporation of China under Grant 5442AI190007.

ABSTRACT The accurate and timely abnormal object detection is of crucial importance for the safe operation of power grid. It is rather difficult, however, to completely manually recognize such objects based on the uploaded pictures in the cloud server. To meet the demand of accuracy and timeliness, this paper proposes to combine the cloud/edge fusion framework and deep learning techniques for abnormal object detection. Specifically, we first train the model of abnormal object detection by using YOLOv4 in the cloud server, and then apply the trained model to detect whether there is an abnormal object for each captured picture in edge servers. As the data sample is not very large at the early stage of the system, we use some enhancement techniques to enlarge the number of pictures, and afterwards new real-time data streams are also used for incremental learning. Our experiments show that the proposed framework can accurately and timely detect the abnormal objects near power transmission lines.

INDEX TERMS Edge-cloud fusion, object detection, YOLOv4, incremental learning.

I. INTRODUCTION

The security and reliability of power transmission lines are of crucial importance to the stability of a smart grid. Many factors, such as abnormal events (e.g., mountain fires/smokes) and artificial destruction (e.g., by a working engineering machinery), can threaten the security of grid infrastructures [1] (see FIGURE 1). When high voltage transmission lines are longer and more dispersed, traveling through complicated terrain conditions (e.g., mountains), it becomes more difficult to continually monitor transmission lines for abnormal object detection via human, mobile robots [2] or helicopters, as it involves extensive labor efforts and high startup and maintenance cost.

More recently, the intelligent vision based automated smart inspection system [3] has been proposed to monitor the power transmission lines, where a series of processing techniques are involved, such as image acquisition and enhancement, segmentation, feature extraction, object location and classification. However, the computation load for model training and testing is often shifted to the cloud server far away from smart devices mounted on the transmission lines, and the bandwidth of the smart devices is also limited such that there will be a large delay for uploading the captured images. Hence, it still remains challenging to accurately and timely detect whether there is an abnormal object near transmission lines such that the administrator can be immediately alerted.

FIGURE 1. The existence of (a) a tower crane on a construction site and (b) fires/smokes near transmission lines will threaten the safety of grid infrastructures.
and some actions can be taken to ensure the safety of the smart grid.

In this paper, we utilize the framework of cloud-edge fusion in the smart inspection system to tackle the above-mentioned problem. With the explosive number of Internet of Things (IoT) devices and the expanding application scenarios of IoT, the network scales dramatically such that the conventional cloud computing paradigm cannot efficiently handle the unprecedented volume of data generated by the large-scale IoT devices. Now it is well known that the cloud computing paradigm suffers from high latency, poor security capability, the bottleneck of bandwidth, etc., and under this circumstance, edge computing as a new computing model is now evolving. Edge computing [4], [5], also termed as cloudlet, fog computing, and edge-clouds, extends data processing to the network edges with close proximity to the data producers, i.e., IoT devices. It enjoys the advantages of low latency, saving bandwidth, protecting data privacy, etc., and is more feasible for applications which require real-time decision-making action on-site, as opposed to the cloud computing. In addition, the fifth generation (5G) wireless technology will also fuel edge computing as it enables data centers at the edges of the networks. As such, recent years have witnessed the growing interests of edge computing from both the academia and industry.

On the other hand, as we move into the highly connected digital economy, artificial intelligence (AI) with many advantages in bridging gaps between the capabilities of machines and humans in such areas as automatic pattern identification and anomaly detection, is now taking the edge computing into a whole new level. The potential of AI is vast and appealing, and there are many kinds of AI-enabled devices that are powerful and enable data collection, storage, computation, and analysis at the edges. The combination of AI and edge computing offers the intelligence at the edges, the so-called edge intelligence [6], which enables many applications, such as smart city, AR/VR, smart retailers, smart home appliance, autonomous driving, intelligent inspection, to name a few, and it thus can be visioned that the edge intelligence will be the next frontier of the IoT.

To achieve the goal of AI, a variety of deep learning techniques have been proposed, which leverages artificial neural network (ANN) to learn the powerful representation of the raw data. Among them, the convolutional neural networks (a.k.a. ConvNets, or CNNs), a special kind of multi-layer neural networks, are specially designed to recognize visual patterns from images where the objects/aspects in the images are assigned an importance (in terms of the learnable weights and biases) with the minimal pre-processing. As some CNNs only perform well for small-scale datasets, certain models or problems exclusively, YOLOv4 incorporates the advantages of the existing CNNs with many useful tricks to achieve high speed and accuracy, which makes it a good fit for the abnormal object detection of transmission lines.

In the paper, we exploit the cloud-edge fusion framework for model training in the cloud server and testing in the edge servers. As model training using deep learning techniques (especially for object detection) often requires high computational resource, we shift the training task to the cloud server and use the trained model for abnormal object detection in the edge servers when receiving uploaded images from terminal devices. If there is an abnormal object, the edge server will send a message, which includes the detected result and the image, to the cloud server for further action. In addition, as in the early stage of the system being deployed, the sampling data for model training are often limited such that the derived model cannot make an accurate prediction, we also use a novel technique, named incremental learning, that applies new sampling data on the previously trained model to enhance the accuracy of the model. As the inference operation is conducted in the edge server near to terminal devices, and only images where an abnormal object is detected by the model are uploaded to the cloud server, the bandwidth can be greatly saved and the results can be quickly obtained by the system administrator. At the same time, we apply YOLOv4 for model training, which can also contribute to the timeliness of abnormal object detection of power transmission lines while the accuracy is not sacrificed.

The reminder of this paper is organized as follows. In Section II we briefly introduce the related work, and detail the system design in Section III. Section IV presents the experimental results and finally, Section V concludes the whole paper.

II. RELATED WORK

In this section, we will briefly present work related to ours, which includes two fields, namely cloud-edge fusion techniques and object detection using CNNs.

A. CLOUD-EDGE FUSION

Recent years have witnessed the increasing amount of Internet of Things (IoT) devices, and it is rather challenging to efficiently manage the explosively generated data traffic in the IoT era. Even though cloud servers are rich in computation and storage capacity, the concerns about privacy, cost and performance are raised when transporting the high volume data from IoT devices to the cloud server for application-oriented analysis. Edge computing as a new paradigm has been proposed where the analysis load is shifted to the network edge with a closer proximity to the data source. Compared with the cloud computing, edge computing enjoys the advantages of low latency, high security, low bandwidth consumption, context-awareness, etc.

In many applications, the emerging paradigm of edge computing and cloud computing are not exclusive. In fact, the computation and storage capacity of edge servers are very limited and the deep learning techniques such as model training for object recognition, which requires high computational power, cannot be done. Hence, researchers have proposed to combine the edge computing with the cloud computing...
(which forms the so-called cloud-edge computing framework) such that the artificial intelligence (AI) techniques can be applied to from a new research area, i.e., edge intelligence.

Wang et al. [7] propose a tensor-based cloud-edge computing framework to offer high-quality, personalized, and proactive services for human in Cyber-physical-social systems (CPSSs). There are two planes in the proposed framework, one is the cloud plane and the other is the edge plane. The former one is in charge of large-scale, long-term, and global data processing, while the latter one is for processing small-scale, short-term, local data. Xu et al. [8] present COM, a computation offloading scheme, for IoT-enabled computing, where the major contributions are the dynamic schedules for data or control constrained computing tasks and the proposed NSGA-III scheme, a non-dominated sorting genetic algorithm III, which can address multi-objective optimization problem of task offloading in cloud-edge computing. In [9], a weighted cost model is proposed for the minimization problem of execution time and energy consumption of IoT applications with multiple IoT devices, cloud servers and edge servers, and the Memetic based application placement technique is also proposed. Shuja et al. [10] propose to map and translate ARM vector intrinsics to x86 vector intrinsics, and analyze the code offloading framework in heterogeneous cloud-edge architectures. Han et al. [11] present a general model for job dispatching and scheduling, which is a fundamental problem in edge-cloud systems, to minimize the job response time. Khan et al. [12] propose to solve the problem of offloading cost estimation for mobile cloud application models and present a mathematical model for calculating the computation offloading cost (in terms of time and energy consumption).

As in cloud-edge fusion computing scenarios, both edge computing and cloud computing suffer from high network consumption, virtual machines can be used for migrating the overloads of physical machines which results in a migration cost. As such, in [13] the high network consumption problem is studied with the consideration of migration cost and communication cost, and accordingly three heuristic virtual machine migration algorithms are proposed. In addition, the fusion of edge and cloud computing makes it possible to use machine learning techniques on edge networks for edge intelligence [6]. In [14], the authors give a comprehensive survey on applying machine learning techniques (e.g., federated learning) for catching in edge networks.

B. OBJECT DETECTION USING CNNs

In computer vision tasks, image classification only outputs the class of one object in an image, object localization computes the location(s) of the targeted object(s) in an image and draws the bounding box(es), and object detection is the combination of these two tasks which first localizes the objects and then classifies them in an image. As in this paper we are interested in detecting what the abnormal objects are and where they are (if any), below we only briefly summarize the object detection techniques, which are categorized into two major families, named R-CNN model family and YOLO model family.

1) R-CNN MODEL FAMILY

The region-based convolutional neural networks (short for R-CNNs), are the first family of techniques with the objectivity of addressing object localization and recognition tasks, which refers to the methods combining region with CNNs designed for model performance. This family include R-CNN, Fast R-CNN, Faster-RCNN and MASK R-CNN. R-CNN [15] has three steps: extract region proposal by drawing candidate bounding boxes, compute CNN features for each region, and classify regions using a classifier model, and the experiments on VOC 2012 show that R-CNN can achieve the mean average precision (mAP) of 53.3%, an improvement more than 30% over the previous best results. Fast R-CNN [16] extends R-CNN by solving the three limitations, namely, a multi-stage pipeline for training, expensive training in terms of space and time, and the object detection is also slow. Instead, Fast R-CNN uses a single training model, and learn and output regions and classification directly. It is significantly faster for model training and inference, however, multiple candidate regions are also proposed for each image. Faster R-CNN [17] further improve the R-CNN model by introducing a region proposal network (RPN) to generate high-quality region proposals such that the object bounds and objectness scores at each position can be simultaneously predicted. Then RPN and Fast R-CNN are merged into a single network with ‘attention’ mechanisms. Hence, both the speed of training and detection are greatly improved. Mask R-CNN [18] is an extension of Faster R-CNN with a branch for object detection and a branch for predicting an object mask. As such, Mask R-CNN can kill two birds with one stone: it can not only detect objects in an image but also generate a high-quality segmentation mask. Besides, compared with Faster R-CNN, Mask R-CNN is simpler to train with only a little more overhead.

2) YOLO MODEL FAMILY

The collection of YOLO (abbr. for you only look once) models is a second family of approaches for object detection designed for real-time use and speed. It is firstly proposed by Joseph Redmon et al. in [19] where only a single neural network for end-to-end training is involved and the bounding boxes and class labels for each bounding box can be predicted directly. It is fast yet the predictive accuracy is very low. The variant YOLOv2 [20] is an update which is designed for higher model performance and can predict 9,000 object classes in an instance. Hence, it is also referred to as YOLO9000. YOLO9000 has a lot of training and architectural changes, e.g., using batch normalization and high-resolution input images. YOLOv3 [21] introduces some minor design changes to YOLO, such as some little representational changes and a deeper feature detector network. The model is bigger yet more accurate, and fast as usual. A state-of-the-art object detection technique of YOLO model
family is YOLOv4 [22]. Some universal features are used in YOLOv4, which include Weighted-Residual-Connections (WRC), Cross-stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT) and Mish-activation, Mosaic data augmentation, DropBlock regularization, and CIoU loss, and it can achieve a high-accuracy of 43.5% AP for the MS COCO dataset at a real time speed.

III. SYSTEM DESIGN

As mentioned earlier, the working engineering machineries such as crane, tower crane, etc., may directly destruct power lines, or indirectly destruct other grid infrastructures (e.g., transmission towers), and the fires or smokes may also threaten the safety of grid infrastructures. Hence, once there are objects/events such as engineering machineries, fires or smokes, which can be detected by our so-called abnormal object detection approaches, near power transmission lines, the intelligent inspection system should immediately send an alert, together with the corresponding picture, to the administrator, such that they can send some workers to the site and check if these objects/events will indeed threaten the grid infrastructures. Clearly, timeliness and accuracy are two key aspects for a good intelligent inspection system. That is, once there is an abnormal object, the system should detect it as soon as possible, and to save the human cost for checking on-site, the system should have an accuracy high enough. To that end, we use the framework of cloud-edge fusion, and the state-of-the-art object detection method named YOLOv4, such that edge intelligence can be implemented for inference at edges.

The edge intelligence can be implemented in different ways. In the cloud computing, the model training and inference are both completed in the cloud server. However, for intelligent inspection system, it is impractical to continuously upload the captured images to the cloud for inference since this way has many disadvantages, e.g., occupying too many bandwidths and incurring a high detection delay. A possible way is to train the model in the cloud and make an inference at the edge based on the trained model where the data is partially offloaded to the edge. Since the inference is shifted to the edges with close proximity to the terminals devices (i.e., smart cameras), the bandwidth and the delay for uploading images to the cloud can be greatly reduced.

FIGURE 2 depicts the system design. The cameras mounted on the transmission lines keep capturing images, and send them to the edge servers. At the early stage of the system where no trained model is available, these pictures are all transferred to the cloud for model training of object detection. After finishing the training process, the cloud offloads the model with parameters to the edges such that inference can be done at edges. With the trained model, the edges will first make an inference for the uploaded images by the nearby cameras and only the images with abnormal object(s), together with an alarm message, will be sent to the cloud. The system administrator will verify this message, and these new pictures in turn will also serve as a new sample for further model training, i.e., the so-called incremental learning, and then the model parameters stored at edges will be updated accordingly.

For object detection of transmission lines, we use the YOLOv4 model on our sample dataset for its high speed and accuracy. YOLOv4 has three building blocks, namely, Head using YOLOv3 [21], Neck using SPP [23] and PAN [24], backbone using CSPDarknet53 [25]. Some tricks are involved in YOLOv4. For example, bag of freebies (BoF) such as CutMix, Mosaic data augmentation, DropBlock regularization and Class label smoothing, and bag of specials (BoS) like CSP, MiWRC and Mish activation, are used for the backbone.

YOLOv4 decomposes the image into a grid of cells, and each grid cell is used to determine whether there is a bounding box such that its center falls within the cell. As a result, each cell will predict a bounding box by computing the coordinates, width, height and confidence, and the classification result as well. As a concrete example, the image in FIGURE 4 is firstly decomposed into a grid of 7 × 7 cells, where each cell predicts 2 bounding boxes, and thus 94 proposed bounding box predictions can be obtained. The final set of bounding boxes and class labels consists of the class probabilities and the bounding boxes. This way we can determine whether there is an abnormal object based on the input image. To fasten the model training, we apply the YOLOv4 on the VOC dataset, 1 and feed the parameters of the trained model as input to further train a YOLOv4 model on our dataset for abnormal object detection.

At the early stage of the deployed system, we only collect a few of samples, while deep learning techniques require a large number of samples to guarantee a high generalization capability. As such we use some image enhancement techniques, including gray scale image enhancement (in terms of contrast, saturation, brightness, etc.), adding noises to the original images, and geometric transformation. The obtained

1http://host.robots.ox.ac.uk/pascal/VOC/voc2012.
images with these techniques can be regarded as new samples under different situations, and thus enrich the sample dataset.

When the system works, the camera will capture more and more images, some of them include objects which may threaten the safety of the grid infrastructures and can be detected by the system. These newly captured images are appended to the old sample dataset stored in the cloud, and the so-called incremental learning techniques [26], [27] are applicable. Specifically, we apply the trained model based on the old sample dataset on the new sample dataset for further learning, and derive a model using both the old and the new sample dataset. As the class labels are often pre-defined, the incremental learning only involves the case of sample size increasing. To make sure that the incremental learning will not forget the old sample dataset, the size of new dataset should be much smaller than the old dataset.

FIGURE 3 shows the snapshot of our system for abnormal object detection of power transmission lines. The system consists of five functions, namely, basis services (real-time monitoring and network topology description), device management, alarm management, maintenance management, and user management. For real-time monitoring service, the system displays the real-time captured pictures (the middle part) and alarming messages including the object type and the location (e.g., near which tower), if abnormal objects are recognized by the system (the rightmost part).

Note that the trained model cannot 100% accurately predict the class label of each image. However, as long as there is an abnormal object, even though it matters what the object is, it does not mean that an image with a wrongly classified label is useless. For example, for an abnormal object, say fires in an image, if it is classified as smokes due to the difficulty of accurately locating the bounding boxes for fires and smokes when they coexist in some situations, the system administrator can still be alerted for the abnormal object. Hence, we apply a two-stage strategy to avoid the ignorance of abnormal object. First, the system detects whether there is an object with one of the pre-determined labels, e.g., engineering machinery, fire/smokes, etc., and if yes, the system outputs the predicted class label of this image. If the predicted label is inaccurate, the administrator can verify the identified label by the system such that the new sample dataset has correct labels for model training in the incremental learning.

IV. PERFORMANCE EVALUATION

A. EXPERIMENT SETUP

We first collect a dataset of 926 samples with 7 categories (i.e., engineering machinery, crane, tower crane, fires, smokes, covering object, hanging object), and then apply image enhancement techniques to obtain two datasets, namely, Set(1,1) with $926 \times 5$ and Set(1,2) with $926 \times 12$ samples, respectively. We use YOLOv4 model and Faster R-CNN model on Set(1,0), Set(1,1) and Set(1,2), respectively, where Set(1,0) denotes the original dataset with 926 samples. Afterwards, we collect 2000 new samples, and divide them into two datasets, i.e., Set 2 and Set 3 where each dataset has 1000 samples. Set 2 will be used for incremental learning, and Set 3 for validation. Note that for each sample, there can be multiple objects/labels such that the total number of labels may be larger than the number of samples. For instance, in a construction site there are multiple engineering machineries. The configuration of our system is as follows: the cloud datacenter uses FUJITSU Server PRIMERGY RX4770 M4, with CPU frequency of 2100 MHz and 112 Cores, and the edge servers are all RX350S, with CPU frequency of 2200 MHz and 16 cores. Each smart camera captures a picture every 5 minutes.

We first compare YOLOv4 with Faster R-CNN for different training sets, and then show the comparison results by YOLOv4, YOLOv3, and SSD [28] on the training set of Set(1,1). In addition, we conduct the comparison study on our approach and the-state-of-the-art abnormal object detection algorithm of power lines, i.e., RCNN4SPTL [29].
RCNN4SPTL is also a deep learning based framework which explores the region proposal network (RPN) to generate the aspect ratio of region proposals for size alignment, and applies end to end training model to improve the efficiency.

### B. PERFORMANCE METRIC

We use the following metrics for comparison:

- **Accuracy**: defined as the number of true positives (i.e., a picture with abnormal objects is identified as a positive) plus the number of true negatives (i.e., a picture without abnormal objects is identified as a negative) to the total number of tested pictures;
- **Detection Time**: defined as the time for detecting abnormal object(s) in a picture;
- **Precision**: defined as the number of true positives over the number of predicted positives;
- **Recall**: defined as the number of true positives over the number of actual positives.

### C. EXPERIMENT RESULTS

FIGURE 5 illustrates the results using Faster R-CNN and YOLOv4 model on different datasets. We find that for Faster R-CNN, the accuracy of the model using Set(1,2) (referred to as Faster R-CNN 12X) is the best, followed by Faster R-CNN 5X which utilizes five times of original samples, and by using the original samples only, the accuracy of Faster R-CNN 1X is the lowest as the sample size is too small. We also notice that Faster R-CNN 12X performs better only slightly than Faster R-CNN 5X. An interesting finding is that YOLOv4 5X performs best, and YOLOv4 12X comes next. This may indicates that by using the image enhancement techniques, the benefit does not always grows with the newly generated samples by the image enhancement process.

As well-known, YOLOv4 model runs faster than Faster R-CNN, and in our experiments, we find that the object detection time for one image by YOLOv4 is about 26ms while the time by Faster R-CNN is around 800ms. At the same time, YOLOv4 can output more accurate results, as verified by the above-mentioned experiments. Next we further compare YOLOv4 with YOLOv3 and SSD. From Fig. 6 we observe that YOLOv4 outperforms YOLOv3 and SSD, while the results by YOLOv3 and SSD are comparable. Based on these findings, we utilize YOLOv4 model for abnormal object detection of transmission lines in our system.

To show the efficiency of incremental learning, we also append Set 2 to the training set Set (1,1) for further training based on the obtained model by YOLOv4. FIGURE 7 shows the comparison results. We find a reasonable fact that the model (referred to as Model 1) by using the training subset of Set(1,1) has a better performance on the testing subset of Set(1,1) that on Set 3. In addition, by using incremental learning, the trained model (referred to as Model 2) has a significant improvement. On the other hand, we also find that the accuracy of Model 2 on Set(1,1) is lower than that of Model 1 for some objects, which shows that the trained model after incremental learning forgets some data in the past. We believe that the root cause is that the sample size of Set(1,1) is too small, and when the size of the training set increases, the forgotten issue can be greatly reduced.

### TABLE 1. The comparison results of our approach and RCNN4SPTL [29].

| Class Label         | Precision Our Approach | RCNN4SPTL | Recall Our Approach | RCNN4SPTL | Detection Speed (seconds) |
|---------------------|------------------------|-----------|---------------------|-----------|---------------------------|
| Crane               | 0.93                   | 0.85      | 0.95                | 0.89      | 0.025                     |
| Fires               | 0.48                   | 0.45      | 0.51                | 0.46      | 0.028                     |
| Hanging Object      | 0.09                   | 0.82      | 0.93                | 0.85      | 0.026                     |
| Smokes              | 0.78                   | 0.72      | 0.81                | 0.75      | 0.027                     |
| Tower Crane         | 0.82                   | 0.75      | 0.87                | 0.8       | 0.026                     |
| Eng Machinery       | 0.91                   | 0.86      | 0.94                | 0.89      | 0.025                     |
| Covering Object     | 0.65                   | 0.58      | 0.68                | 0.62      | 0.027                     |

VOLUME 8, 2020
that our approach outperforms RCNN4SPTL in terms of accuracy and detection speed.

Finally, TABLE 2 presents the difference of our cloud-edge platform and cloud platform in terms of average computation time (ACT) and average communication cost (ACC). The ACT is defined as the average of object detection time in edge servers for cloud-edge platform or the cloud datacenter for cloud platform, and the ACC is defined as the average of communication cost for uploading all pictures to the cloud for cloud platform or for uploading the pictures with abnormal objects to the cloud for cloud-edge platform. We find that our cloud-edge fusion based framework outperforms significantly than the cloud platform. Clearly, when the ratio of the number of pictures with abnormal objects to the number of all pictures becomes smaller, our approach performs better.

V. CONCLUSION

In this paper we present a cloud-edge fusion based abnormal object detection system of power transmission lines. We first train the model using YOLOv4 on the cloud based on the dataset collected at the early stage, and then apply the trained model for inference at edges. The images with positive results will be uploaded to the cloud for incremental learning. The experiments show the efficiency of the proposed framework.

As in our system, each smart camera has powerful computation capability, and the training dataset with labels by each camera is often small, we can use federated learning to further improve the system performance. We will leave this for our future work.

REFERENCES

[1] J. Lin, B. Zhu, P. Zeng, W. Liang, H. Yu, and Y. Xiao, “Monitoring power transmission lines using a wireless sensor network,” Wireless Commun. Mobile Comput., vol. 15, no. 14, pp. 1799–1821, Oct. 2015.
[2] S. Lu, Y. Zhang, and J. Su, “Mobile robot for power substations inspection: A survey,” IEEE/CAA J. Automatica Sinica, early access, Jan. 25, 2017, doi: 10.1109/JAS.2017.7510364.
[3] D. Pham and R. J. Alcock, “Smart inspection systems: Techniques and applications of intelligent vision,” in Smart Inspection Systems: Techniques and Applications of Intelligent Vision. Amsterdam, The Netherlands: Elsevier, 2003, pp. 1–221.
[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” IEEE Internet Things J., vol. 3, no. 3, pp. 637–646, Oct. 2016.
[5] E. Ahmed, A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran, and M. Shoaib, “Bringing computation closer toward the user network: Is edge computing the solution?” IEEE Commun. Mag., vol. 55, no. 11, pp. 138–144, Nov. 2017.
[6] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” Proc. IEEE, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.
[7] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, “A cloud-edge computing framework for cyber-physical-social services,” IEEE Commun. Mag., vol. 55, no. 11, pp. 80–85, Nov. 2017.
[8] X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, and L. Qi, “A computation offloading method over big data for IoT-Enabled cloud-edge computing,” Future Gener. Comput. Syst., vol. 95, pp. 522–533, Jun. 2019.
[9] M. Goudarzi, H. Wu, M. S. Palaniswami, and R. Buyya, “An application placement technique for concurrent applications in edge and fog computing environments,” IEEE Trans. Mobile Comput., early access, Jan. 15, 2020, doi: 10.1109/TCM.2020.2967041.
[10] J. Shuja, S. Mustafa, R. W. Ahmad, S. A. Madani, A. Gani, and M. K. Khan, “Analysis of vector code offloading framework in heterogeneous cloud and edge architectures,” IEEE Access, vol. 5, pp. 24542–24554, 2017.
[11] Z. Han, H. Tan, X.-Y. Li, S. H.-C. Jiang, Y. Li, and F. C. M. Lau, “On-Disc: Online latency-sensitive job dispatching and scheduling in heterogeneous edge-clouds,” IEEE/ACM Trans. Netw., vol. 27, no. 6, pp. 2472–2485, Dec. 2019.
[12] A. U. R. Khan, M. Othman, A. N. Khan, J. Shuja, and S. Mustafa, “Computation offloading cost estimation in mobile cloud application models,” Pers. Ubiquitous Comput., vol. 97, no. 3, pp. 4897–4920, Dec. 2017.
[13] C. Ling, W. Zhang, H. He, and Y.-C. Tian, “Network perception task migration in cloud-edge fusion computing,” J. Cloud Comput., vol. 9, no. 1, p. 43, Dec. 2020.
[14] J. Shuja, K. Bilal, E. Alanazi, W. Alasmardy, A. Alashaikh, and Y. A. Zomaya, “Applying machine learning techniques for caching in edge networks: A comprehensive survey,” 2020, arXiv:2006.16894. [Online]. Available: https://arxiv.org/abs/2006.16894
[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.
[16] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1440–1448.
[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2961–2969.
[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.
[20] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6517–6525.
[21] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” Tech. Rep., 2018. [Online]. Available: https://arxiv.org/abs/1804.02767
[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv:2004.10934. [Online]. Available: http://arxiv.org/abs/2004.10934
[23] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[24] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8759–8768.
[25] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “CSPNet: A new backbone that can enhance learning capability of CNN,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 1571–1580.
[26] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, “Large scale incremental learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 374–382.
[27] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, “Class-incremental learning via deep model consolidation,” 2019, arXiv:1903.07864. [Online]. Available: http://arxiv.org/abs/1903.07864

[28] W. Liu, D. Anguelov, D. Erhan, C. Y. Fu, A. C. Berg, C. Szegedy, and S. Reed, “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 21–37.

[29] W. Zhang, X. Liu, J. Yuan, L. Xu, H. Sun, J. Zhou, and X. Liu, “RCNN-based foreign object detection for securing power transmission lines (RCNN4SPTL),” Procedia Comput. Sci., vol. 147, pp. 331–337, Jan. 2019.

SHUHUA ZHANG is currently pursuing the Ph.D. degree with North China Electric Power University, China. He is also a Senior Engineer with China Electric Power Research Institute Company Ltd. His research interests include integrated circuit, artificial intelligence, edge computing.

JIYE WANG is currently a Ph.D. Supervisor and a Senior Engineer with China Electric Power Research Institute Company Ltd. His research interests include energy Internet and smart grid.

JIE TONG is currently a Senior Engineer with China Electric Power Research Institute Company Ltd. His research interests include smart sensing, energy Internet, and edge computing.

JUN ZHANG is currently an Engineer with China Electric Power Research Institute Company Ltd. His research interest includes smart sensing.

MINGHAO ZHANG is currently an Engineer with China Electric Power Research Institute Company Ltd. His research interest includes smart sensing.