Noname manuscript No.
(will be inserted by the editor)

An Edge-Assisted Cloud Framework Using a Residual Concatenate FCN Approach to Beam Correction in the Internet of Weather Radars

Hao Wu* · **Qi Liu*#** · **Xiaodong Liu** ·
Yonghong Zhang · **Zhiyun Yang**

Received: date / Accepted: date

Abstract Internet of Things (IoT) has been rapidly developed in recent years, being well applied in the fields of Environmental Surveillance, Smart Grid, Intelligent Transportation, and so on. As one of the typical earth-based meteorological observation methods, networked Doppler weather radars, i.e. the Internet of weather Radars (IoR) can detect the signals of large-area water particles in the atmosphere with high resolution, but suffer from beam blockage due to surrounded mountains, buildings, as well as other obstacles. In addition,

* Hao Wu and Qi Liu contribute equally to the article.

# Corresponding author.

Hao Wu
School of Computer and Software
Engineering Research Center of Digital Forensics, Ministry of Education
Nanjing University of Information Science and Technology
E-mail: wuhao@nuist.edu.cn

Qi Liu
School of Computer and Software
Nanjing University of Information Science and Technology
E-mail: qi.liu@nuist.edu.cn

Xiaodong Liu
School of Computing
Edinburgh Napier University
E-mail: x.liu@napier.ac.uk

Yonghong Zhang
Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET)
Nanjing University of Information Science Technology
E-mail: zyh@nuist.edu.cn

Zhiyun Yang
School of Computer and Software
Engineering Research Center of Digital Forensics, Ministry of Education
Nanjing University of Information Science and Technology
E-mail: yangzhiyun32@163.com

how to establish a distributed platform for large-scale radar data analytics becomes critical and challenging, especially considering optimised strategies on the storage, processing and exchange of radar raw data, beam/echo signal, and final products etc. In this paper, an edge-assisted cloud framework is proposed to facilitate effective and proficient communication and progression, where echo signal from a single site radar can be analysed and pre-processed at the edge, and then trained in the cloud with elastic resources and distributed learning ability. A Residual Concatenate Fully Convolutional Network (RC-FCN) is presented for beam blockage correction, which is integrated into the framework to be compared with other deep learning models, including FCN, ResNet, VGG, etc. According to experiment results, better performance and efficiency have been achieved using the proposed framework and its fitted RC-FCN model.

Keywords Edge Computing · Internet of Radars · Residual Concatenate · Beam Blockage Correction · Weather Radar

Mathematics Subject Classification (2020) MSC code1 · MSC code2 · more

1 Introduction

Weather radar is an important equipment of weather observation in meteorological department. In order to make the observation more accurate and timely, the deployment of weather radar is very intensive, especially in coastal areas. The weather radar outputs data every six minutes generally. The amount of data generated by weather radar network is very huge. Therefore, it is an important and difficult problem to save massive radar data orderly. At the same time, the data detected by different radar stations are copyrighted. In the process of transmission and storage, it is necessary to ensure the security and privacy of radar data.

On the other hand, there are many quality problems in the direct output data of weather radar. When directly applied to the quantitative analysis method of meteorological operation, it is easy to cause obvious errors. Common quality problems include non precipitation target blocking, ground clutter and signal attenuation. Therefore, before the data are put into use, the quality control operations such as blocking correction, clutter removal and attenuation correction are needed.

In this paper, an edge-assisted cloud framework based Residual Concatenate Fully Convolutional Neural Network (RC-FCN) is proposed, which can achieve the correction of beam blockage independent of terrain data and ensure the access performance of radar data. Blocking correction is regarded as an image in-painting problem in this study. The system adopts the combination of edge and cloud computing. The edge part is each radar station which is mainly responsible for detection, data storage and preprocessing. The cloud computing part is high performance server hardware system optimized for artificial intelligence and remote access software system. In this way, the whole system can harmonize the contradiction of weather radar AI system in data storage, data transmission, and efficient invoking of AI algorithms. In this study, the problem of weather radar beam blockages is mainly solved.

The blocking part of radar image is just similar to the missing data of image. RC-FCN is a multilayer neural network with image restoration function. It consists of an convolutional encoder and a deconvolutional decoder network. During training, the input of network is the blockages contained weather radar images. The output of the network is the repaired image corresponding to the blocking positions. The label is the real image corresponding to the blocking positions. After training with large amount data finished, the network can automatically fill in the missing data of the blocking part according to the context information of the radar image. In fact, the training process of the network is to find the relationship of data distributions between the blocked and normal part in weather radar images.

Experiment shows that the method in this paper is superior to the traditions in critical success index (CSI), false alarm rate (FAR) and probability of detection (POD)[26]. Due to the different models and principles of radar equipment, the traditional methods need to adjust the algorithm parameters according to the actual situation to adapt different types of radar-based data. The method proposed in this paper is based on radar image. Radar image is the advanced product of radar system, which has strong generality. The training of this method is based on radar image. Therefore, this can be applied to almost all weather radar systems without parameter adjustment of specific equipment. This method is universal and convenient.

2 Related Work

2.1 Edge and Cloud Combined Computing

Compared with cloud computing, edge computing has its unique advantages. For example, it can achieve faster network service response to meet the basic needs of the industry in real-time business[12,46,1], application intelligence[38], security and privacy protection[42]. Edge computing technologies have been widely used in the field of Internet of Things (IoTs)[23,8,30,35], Internet of Vehicles (IoVs)[41,40,19], smart city[31] and so on. Atmospheric detection and weather monitoring are special IoTs application scenarios[18]. The network of Radars (IoRs) is also a special type of IoTs. Because the scanning range of a weather radar is limited, usually covering a radius of about one hundred to four hundred kilometres. A network composed of multiple radars is needed to completely cover a certain area. Each radar generates several GB of data every day. How to process so much data become a challenging problem. The radar node for lightweight data processing and the comprehensive analysis processed by cloud high-performance computer will be a better solution. At present, there are many computing systems that combine the advantages of edge computing and cloud computing. Zamora-Izquierdo[45] proposed an exchangeable low-cost hardware based and multi software platform supported edge computing platform for Precision Agriculture management. The local cyber-physical systems can gather information and execute control actions. The cloud platform can collect information for analysis. Wu[36] proposed a distributed deep learning-driven task offloading algorithm for mobile devices, edge cloud server, and central cloud server. The algorithm can give the optimised decisions in the mixture edge and cloud computing environments. Edge computing and cloud computing are not opposite, but more often to make up for their shortcomings. Duc[5] surveys problems in joint edge-cloud environments. The research shows that machine learning technologies can solve the scheduling and allocation of computing resources in the environment. Miao[21] also uses LSTM model to predict the task offloading and migration in mobile-edge cloud computing and reduce the task delay. Network security and efficiency are also important features of cloud computing and edge computing. Wang[34] proposed a trust evaluation mechanism and a service parameter template combined architecture that improve the security and efficiency of IoT-Cloud systems.

2.2 Typical Beam Blockage Correction Methods

Due to the obstruction of tall buildings, trees and other obstacles near the radar, the missing or deviation of echo data detected by the radar is common. Even the slightly block, the electromagnetic wave emitted by the radar can not propagate forward completely, which make the echo be weak (partial block) or disappears (complete block).

In order to reduce the influence of beam blocking on radar data, most research methods are based on digital elevation model (DEM). Wang [9]takes Quzhou radar in Zhejiang, China as an example to study the reliability of using ASTER GDEM V2 and SRTM3 V4 data, to simulate the radar beam blockages.

However, DEM data is not enough to completely determine the areas which are blocked for its low update frequency. It is difficult to get the latest DEM data in time. Therefore, many researchers proposed other methods to detect the blockages. McRoberts [20] proposed a new spatial analysis technique to objectively identify the area of precipitation estimation affected by beam blockage. Li [14] proposed a method based on spatial distribution statistics and fuzzy logic, which can detect clutter and beam blocking on the ground independent of DEM data. However, this method only deleted the target object, and did not correct the beam blockages and clutter.

2.3 Relevant Image Inpainting Methods

The block correction problem can be considered as an image inpainting problem. The parts of blocked can be regarded as the missing part of images. In recent years, deep learning has achieved very obvious step forward in the field of image in-painting[33,13,4]. Cai[2] proposes a GAN based model, which can generate multiple possibilities for image in-painting task. Liu [16] proposed to use partial convolution to solve the problem of color difference and blurriness in image restoration. Yu [44] proposed a method which can compose the new image structure, and use the surrounding image features to help the process of network training. Song [28] proposes a method, which divides the image restoration task into inference and translation, and uses deep learning network to model each step. In order to make the repaired region image smoother and clearer, Nazeri [22] proposed a new image restoration algorithm, which can repair the missing region more precisely.

In recent years, due to the excellent performance of deep learning and the AI computing capacity increase[24], the field of meteorological research gradually began to use this technology and other related methods to solve related problems, such as precipitation prediction[32], analysis and prediction of ENSO phenomenon[6], landslide early warning[15]. They all achieved good results. Shi [39,25] proposed a new method of combining convolution with LSTM network and a Trajectory GRU model to achieve the prediction of time series data of weather radar images which has achieved good results. Because this kind of thinking is the first in the field of meteorology, many researchers have carried out many follow-up studies based on this [11,37,43]. Besides the precipitation forecast, deep learning has also achieved good results in the field of ENSO analysis and prediction. The study of Ham [6] which utilized the convolution method not only improves the prediction skills of ENSO, but also fully proves the feasibility of the application of deep learning and other methods in the big data of Geoscience.

In general, deep learning technology is developing rapidly in the field of image in-painting, and has made great progress in recent years. Therefore, by applying similar technologies to block correction, it has great potential and is expected to be a better correction results. However, due to the occlusion of objects, atmospheric refraction, equipment failure and other reasons, some of the radar data are missing or abnormal. In order to solve the problems, this study focuses on the correction of radar data missing due to various reasons.

3 An Edge-Assisted Cloud Framework for Internet of Weather Radars

Fig.1 shows the structure of the Edge-Assisted Cloud Framework for Internet of Weather Radars proposed in this study. Due to the limited detection range of weather radar, it needs hundreds of weather radars to achieve the meteorological detection within the territory for a vast country or region. The detection network composed of hundreds weather radars is called radar network. If each radar is regarded as an Internet of Things device, such a network can also be regarded as an Internet of Radars, which is a special type Internet of Things that characterized by huge amount of big data. At the same time,

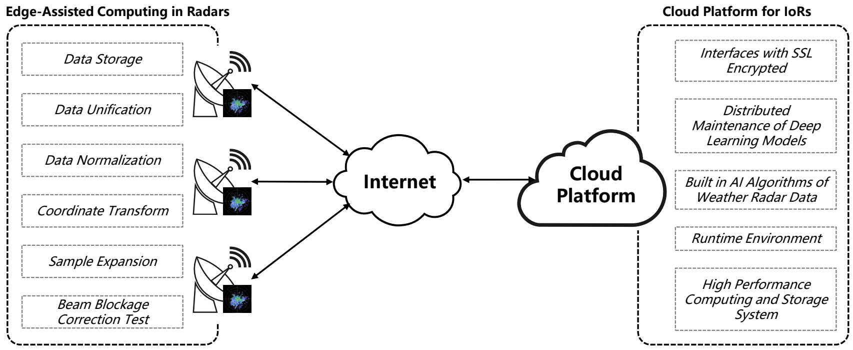

Edge-Assisted Computing in Radars
Data Storage
Data Unification
Data Normalization
Coordinate Transform
Sample Expansion
Beam Blockage Correction Test
Internet
Cloud Platform
Cloud Platform for IoRs
Interfaces with SSL Encrypted
Distributed Maintenance of Deep Learning Models
Built in AI Algorithms of Weather Radar Data
Runtime Environment
High Performance Computing and Storage System

Fig. 1 An edge-assisted cloud-assisted framework for Internet of Weather Radars

because of the copyright of radar data, it needs a way to manage such a large amount of radar data in a centralized way and ensure that the data of each radar can not be stolen and can only be accessed by the authorized users.

The cloud platform is mainly responsible for the training of network model. At the same time, it also manages the computing resources and provide secure remote call services. For example, encrypted interfaces, distributed maintenance of deep learning models, software runtime environment, etc. For privacy reasons, the cloud does not accept the original radar data from the edges. Only the preprocessed radar image data sets need to be uploaded to the cloud platform for AI model training. After the training, the models will be downloaded to each edge end for other services.

3.1 Distributed Edge Computing for Radar Data Preprocessing and Storage

In this study, all the data detected by the radars are stored in their own station. The computing system fo each edge radar station is mainly responsible for data preprocessing, for example, data unification, data normalization, coordinate transform and sample expansion, etc. Besides, after the training of artificial intelligence model. In addition, the edge side can download the models from the cloud and run the model locally. In general, the edge side is mainly responsible for lightweight computing tasks.

The framework mainly includes the following modules:

Authority Management: This module is mainly used to verify whether the request sent by each user is legal, so as to prevent users or malicious attackers from abusing resources. Through authority management, the security of the server can be guaranteed to a certain extent. At the same time, the permission management can distinguish the use space of each user and prevent the training/testing tasks on the server from being too chaotic.

Feature Extraction: This module provides various feature extraction methods for feature extraction. The interface needs to select the specified training data set and parameters for training. After feature learning, the interface gen-

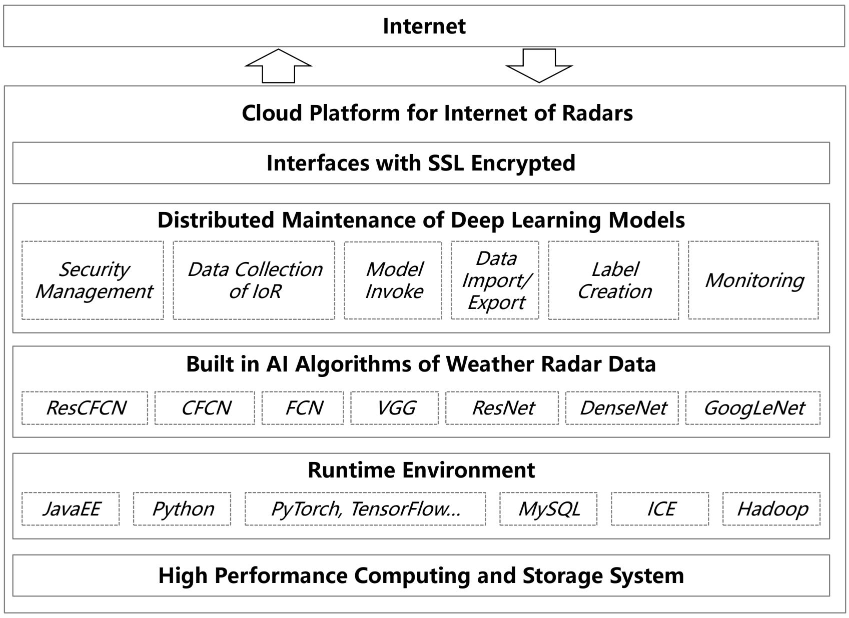


Fig. 2 Cloud platform structure for Internet of Weather Radars

erates the corresponding network model and stores it for the next calculation call.

Training and Testing: This module contains training and testing processes. In the training process, users need to specify the feature extraction network and training data set, and different models can be trained to obtain different parameters. The test process needs to specify the detailed model and test data. All prediction results of the model will be stored in the system database. Users can query the detailed analysis results through task ID.

Assisting Utilities: The module is used to manage training and test data, as well as test and training results. It mainly includes the following eight functions: upload and download of training and test data sets, download of test data results, query of existing data sets, query of network model training progress, query of training progress, query of trained network model, query of trained classifier knowledge and query of classifier test progress.

3.2 Deep Learning Models and High Performance Computing for Remote Invocation

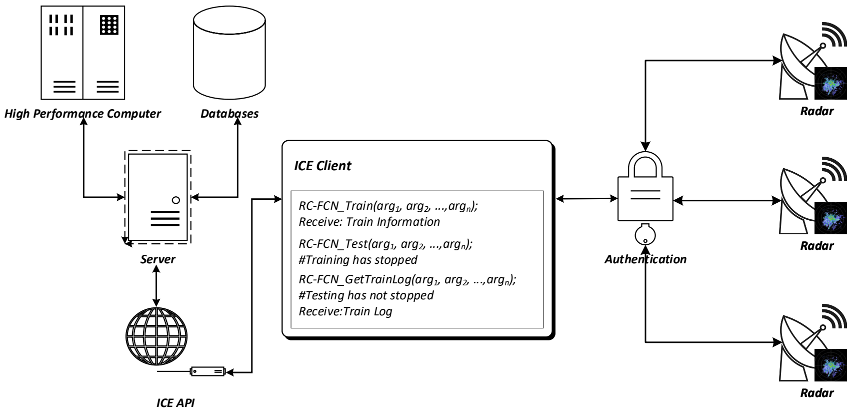
Fig.2 shows the cloud platform framework of this study. This framework uses restful style Java EE API and ICE (Internet Communications Engine) to achieve the process of remote sending and defining parameters to the deep learning model, invoking the specified Python deep learning program modules, obtaining the response results and other necessary functions. The framework is running on a high performance computing and storage system for the needs of giant GPU accelerated computing and big data storing ability. On this basis, the framework needs various software running environment, such as Java EE for network services, Python and various deep learning frameworks for deep learning algorithm, Hadoop and MySQL database system for distributed storage. The built-in weather data processing algorithms are running on the environment. The details of the algorithms are introduced in Section 4. In order to achieve the orderly access and privacy control of radar data, these algorithms are wrapped by the interfaces. Users can access these built-in algorithms through them with the help of SSL encrypted Internet of Weather Radars transmission network to manage and process radar data.

The cloud platform mainly includes model training and testing process. In the training process, the user needs to specify the feature extraction network and training data set. Different network training results in different radar correction models. In the test process, the correction knowledge of the correction algorithm and the preprocessed data set are required to complete the correction of radar data. The detailed results will be stored in the system database, and users can query the detailed correction results through the IDs of results. The reason why the cloud can also invoke models going testing stage is that some edge equipment have no ability to run the models.

3.3 ICE-based Distributed Interfaces between Edge and Cloud

ICE is similar to socket communication technology. It deals with all the underlying network interface programming, so that developers do not have to consider the details such as opening network connection, serialization and deserialization of network data transmission, number of attempts of connection failure, etc. ICE describes the interface of a service through neutral language, which is independent of the specific programming language, SLICE (Specification Language for ICE), so as to separate object interface and its implementation. The client and server can use different programming languages and the communication is efficient and safe. This method can cross platform and ensure the security of user data. With the help of ICE, the results of ICE program can be better called directly by the programming language used on the client side, so that the programming style is consistent. In the AI computing environment, the security of the system is also very important[3]. All communication channels can be configured with various industrial level security protocols,such as AES and RSA, to ensure the communication security in this process.

In this study, ICE framework technology is used to access restful style APIs, provide interfaces for users to access services, and ensure the security of user data while cross platform. Using ice framework, the results of ice program can be better called directly by the programming language used on the client side, so as to keep the programming style consistent. The communication process between the server and the client under the ice framework is shown in Fig.4.



```
RC-FCN_Train(arg1, arg2, ...,argn);
Receive: Train Information
RC-FCN_Test(arg1, arg2, ...,argn);
#Training has stopped
RC-FCN_GetTrainLog(arg1, arg2, ...,argn);
#Testing has not stopped
Receive:Train Log
```

Fig. 3 Cloud platform structure for Internet of Weather Radars

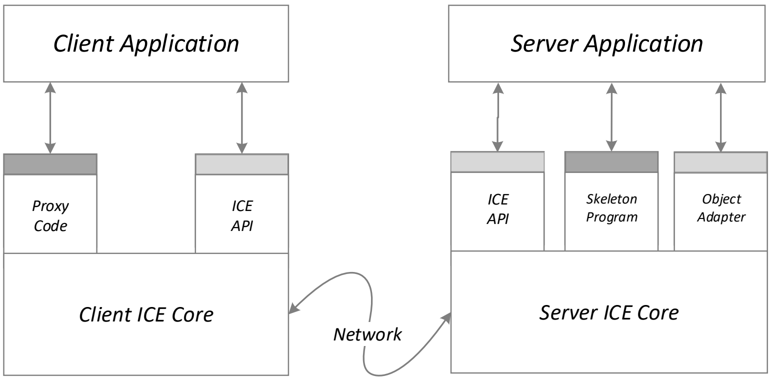


Fig. 4 Communication process between server and client in ICE framework

4 Detailed System Design of Edge-Assisted Computing in Radar Stations

4.1 Weather Radar Data Preprocessing

Deep learning relies heavily on big data and requires enough samples to train the network. However, there are various problems in the radar data observed naturally. For example, the sample distribution is uneven, and the image quality is poor. Before training the model, it is necessary to preprocess the original radar data which can help make the training successfully.

Radar has a large amount of data because of its wide range, large quantity and high degree of automation. However, the radar data with blocking is relatively small, which is not enough to support the deep learning model network for training. Therefore, it is necessary to expand the radar data with blocking properly by some means so as to train the model. Before expanding the data, we need to find out the causes and characteristics of blocking.

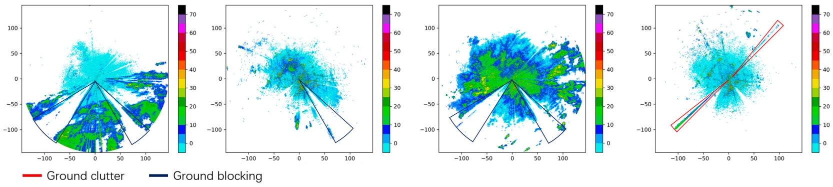


Fig. 5 Beam blockage samples of NUIST weather radar station. The red line circles the beam blockage caused by ground clutter and the blue line circled is ground blocking.

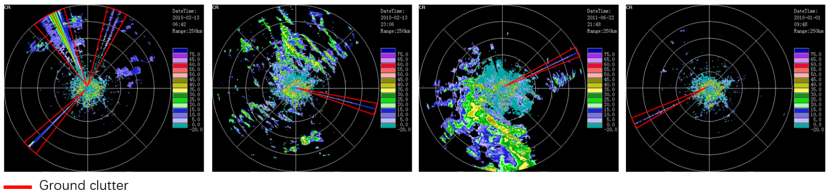


Fig. 6 Beam blockage samples of Guangzhou weather radar station. The red line circled the beam blockages are caused by ground clutter.

For the reflectivity image of Doppler weather radar, the radar is located in the centre of the image. As the radar rotates, it emits electromagnetic waves. Radar draws reflectivity image by receiving echo signal in a period. Fig.5 and Fig.6 shows two class weather radar reflectivity images which are from NUIST and Guangzhou weather radar stations. Different colours represent the reflected radar echo intensity of water vapour particles in the air. However, it can be observed that there are some significant lack of area which are circled with red and purple lines. These areas can be called beam blockages. The red line circle area has a small blocking range, which usually presents as a small range of echo data missing. The common causes for this phenomenon is that there are small objects in this direction, such as trees and buildings, which block the propagation of electromagnetic wave. The area circled by purple lines is a large area of signal loss. Most of the reason for this phenomenon is that there is a huge object block in the direction of electromagnetic wave emission, such as mountains, huge buildings.

Besides the blockages of radial data, sometimes the radar signal will also appear some abnormalities caused by noise from ground. This will lead to redundant radial data in radar image. Fig. 6 mainly shows a typical case of data redundancy.The areas circled by red lines are the abnormal parts of the data.This kind of signal is not necessary for weather radar, which is not necessary for the field of weather research. Therefore, for this kind of signal, it is also regarded as the lack of radar data, which is considered as a kind of beam blocking situation in this paper.

To solve the above problems, the following Algorithm 1 is adopted in this paper. It shows the main steps of data preprocessing. The input of the algorithm is origin weather radar observation images series, and the output is the transformed dataset, including manually masked inputs and ground truth labels. The θ in the algorithm means the rotation angle of radar scanning which is 360 as usual. The D means the distance of scanning distance.

Algorithm 1 Main steps of data preprocessing

```
Input: The original weather radar observation image series, $X_n$.
Output: The updated transformed dataset with masked inputs and ground truth labels, ($T'_{360\times D}$, $T_{360\times D}$).
for each $x \in X_n$ do
    Cut the radar image, get the main echo area from $x$
    De-colour the $x$ to gray in one channel
    De-textures and normalize the pixels in $x$:
    for each pixel value $p_i$ in $x$ do
        Normalize pixel value $p_i$ with the standards
    end for
    Transform $x$ to Angle-Distance Coordinate System:
    for $\theta = 0$; $\theta < 360$; $\theta++$ do
        for $d = 0$; $d < D$; $d++$ do
            $t(\theta, d) = x(d \cdot cos\theta, d \cdot sin\theta)$
            Store $t(\theta, d)$ to $T_{360\times D}$ as LABEL
            Add random mask to $t(\theta, d)$ as $t'(\theta, d)$
            Store $t'(\theta, d)$ to $T'_{360\times D}$ as INPUT
        end for
    end for
end for
return ($T'_{360\times D}$, $T_{360\times D}$)
```

4.2 Data Normalization

Different colours and textures are usually used to represent the echo reflectivity with different intensities. This helps the human's eyes to observe different intensity detail information. But for the deep learning model, the complex and unnecessary texture features will increase the training difficulty and convergence speed of the network, and eventually lead to the reduction of model output accuracy.

On the other hand, different types of radars or different radar signal image processing programs use different colour codes when generating radar signal images. As we can see in the Fig. 5 and Fig. 6, the white background radar images and the black background radar images are from the different radar. They used different standards to visualize radar signals. Signal intensities are represented by different colours according to its value. But what kind of colour represents what kind of intensity, these two radars adopt different schemes.

In order to improve the generality of this research model, it is necessary to unify the radar intensity and colour representation under the same standard before the model training. On the other hand, radar echo reflectivity value is float data usually. When the radar image is generated, these floating-point numbers will approximate to the closest colour threshold range according to

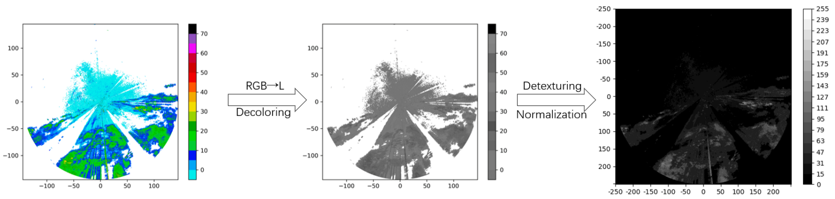


Fig. 7 Data normalization of NUIST weather radar station

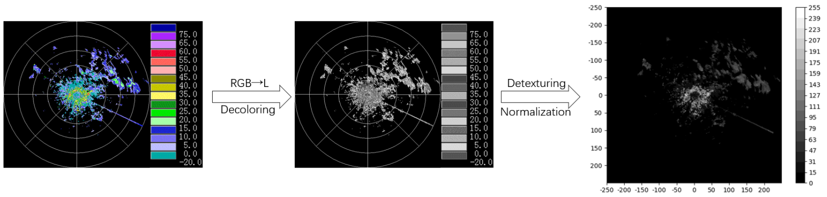


Fig. 8 Data normalization of Guangzhou weather radar station

the colour standard, and use the colour value to represent the intensity value of this range in the image. The number of colour standard is about 15-20 in weather radar field. In this paper, the colours of echo intensities value from different radar will be transport into the same colour space. It means that all the pixels value of the radar images will be represent with the same standard. Pixel values will be evenly distributed between 0-255 according to the same standard in the form of gray images. In this way, the model can be used in the radar data of one station after being trained by the radar data of another one.

Fig. 7 and Fig. 8 show the normalization progresses fo Guangzhou station and NUIST station. The general processing steps are similar. The first step is decoloring. In this step, colour images are transformed into gray scale images. The second step is normalization. In this step, the pixel values of gray scale images are redistributed according to one-to-one correspondence, evenly distributed between 0-16. In this study, echo data from two radar stations just have 16 intensity segments. So they can be represent by the values as (255, 239, 223, 207, 191, 175, 159, 143, 127, 111, 95, 79, 63, 47, 31, 15). A large value indicates a high intensity of radar reflected signal. Finally, we use 1-16 to index these regularized values as their classification numbers. Besides, the number 0 is used to represent the null area where the area with no radar echo.

The specific regularization rules of pixel colour values are shown in Table 1 and Table 2. It should be noted that the intensity colour of radar image in Guangzhou station is represented by a kind of texture. Each intensity segment contains multiple colour values that do not repeat each other.

Table 1 Colour Transform Rules of NUIST Station Radar Images

Colour	Intensity	Gray Value	Normalization	Index
⬛	>70	0	255	16
🟪	65-70	111	239	15
🟪	60-55	107	223	14
🟥	55-50	67	207	13
🟥	50-55	62	191	12
🟥	45-50	72	175	11
🟧	40-45	127	159	10
🟧	35-40	172	143	9
🟨	30-35	202	127	8
🟩	25-30	170	111	7
🟩	20-25	96	95	6
🟩	15-20	126	79	5
🟩	10-15	125	63	4
🟦	5-10	40	47	3
🟦	0-5	125	31	2
🟦	<0	146	15	1

Table 2 Colour Transform Rules of Guangzhou Station Radar Images

Colour	Intensity	Gray Value	Normalization	Index
🟦	>70	114,115	255	16
🟪	65-70	87	239	15
🟪	60-65	84	223	14
🟥	55-60	39,40,52,53,249	207	13
🟥	50-55	26,37,38,50	191	12
🟥	45-50	35	175	11
🟫	40-45	3,223	159	10
🟨	35-40	221,222	143	9
🟨	30-35	13,217,218	127	8
🟩	25-30	2,175,187,199	111	7
🟩	20-25	184,185	95	6
🟩	15-20	168,179,192,191,113	79	5
🟦	10-15	101,102,103,111, 125,126,245	63	4
🟦	5-10	97,109,234,247, 246,229,121,127,230	47	3
🟪	0-5	95,106,107,108,119	31	2
🟦	-20-0	150,151,162	15	1

4.3 The Transform from Polar to Cartesian Coordinate System

Deep neural network usually contains a large number of parameters, which will occupy a very large memory capacity. The more parameters, the slower the training speed of the network. Therefore, in the design of the network, it is important to save memory as much as possible. The input scale of the network has a great influence on the network parameters. The more dimension of input parameters, the more layers and parameters of network are needed to get good training effect. However, the valid data of the original radar image exists in the rectangular image as a circle. Effective information only takes up the part of the radar scanning circle in the images centre. So, it is necessary to drop the invalid data of the images.

The drop method of the radar image redundant data is opposite to that of radar image construction. Generally, the weather radar takes the radar station as the centre, fixes a certain elevation angle, rotates while transmitting electromagnetic wave. In fact, the original radar data is the echo reflectivity intensity data obtained according to each rotation angle. The round visual radar image is processed based on the echo intensity data from different angles. Therefore, in this paper, the images are expanded from the centres which is opposite of the construction progress. The interval angle of expanding is 1 degree and the expanded radius is 250 pixels. Therefore, the final 500×500 radar image will be expanded into a 360×250 rectangular image. The x-axis of the original image is the east-west direction, and the y-axis is the north-south direction. The data in transformed image are only valid radar scanning data. The x-axis of the transformed image is the distance between the reflected object and the radar transmitting point, and the y-axis is the rotation angle of the radar. This process is equivalent to transforming radar image from polar coordinate system to Cartesian rectangular coordinate.

The amount of data after redundancy removal is 36% of the original data. From this method, the valid data is retained, and the amount of data is greatly reduced. The Fig. 9 and Fig. 10 shows the changes before and after data processing. On the other hand, it can be seen that the ground clutter and ground blocking areas have changed from fan-shaped areas to rectangular areas. This change will also help to blockage corrections in next steps.

4.4 Sample Expansion Method

In the case of natural observation, the possibility of weather radar beam blockages is relatively small. The data of beam blockages is not enough to support the deep learning training for large amount of data demanding. Therefore, in this study, large-scale training data will be generated manually according to the block data of natural observation. There is a difference between the handcrafted data generated by program and the data observed by nature. However, the manual blockage areas are much larger and positions are more random than that of natural observation.

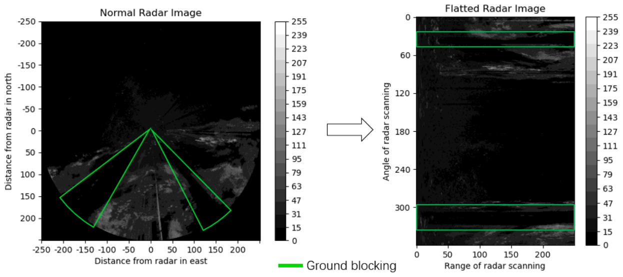


Fig. 9 Coordinate system transform of NUIST weather radar station

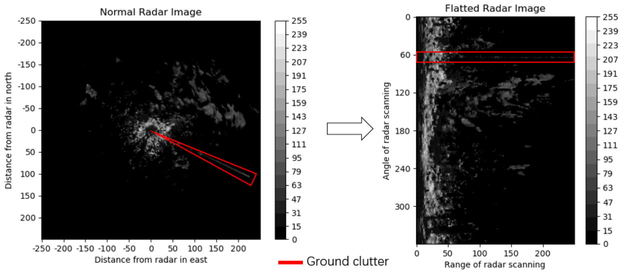


Fig. 10 Coordinate system transform of Guangzhou weather radar station

As shown in the figures on the left of Fig. 9 and Fig. 10, most of the obstructions are fan-shaped on the radar image in natural observation. After the above transformation, these fan-shaped blockages are almost perfectly transformed into rectangles in the Angle-Distance coordinate system. According to this phenomenon, the rectangle is used to cover the original data to simulate the beam blockage in natural observation. The positions and widths are generated randomly with uniform distribution. The starting indexes of block areas in y-axis direction are between [0, 360] and the heights are between [1, 18]. The two cases of manual blockages and original images are shown in Fig. 11. The black rectangular areas circled in red are the artificial blocking areas.

5 The Model Design in Cloud Computing Platform

5.1 Residual Concatenate Fully Convolutional Neural Network

In this study, weather radar beam blockage correction is regarded as a classification problem. The model can classify all the unknown pixels at the same time in the condition of known areas. The classification rules of the model are obtained by big data training progress. A detailed description of the model is given below.

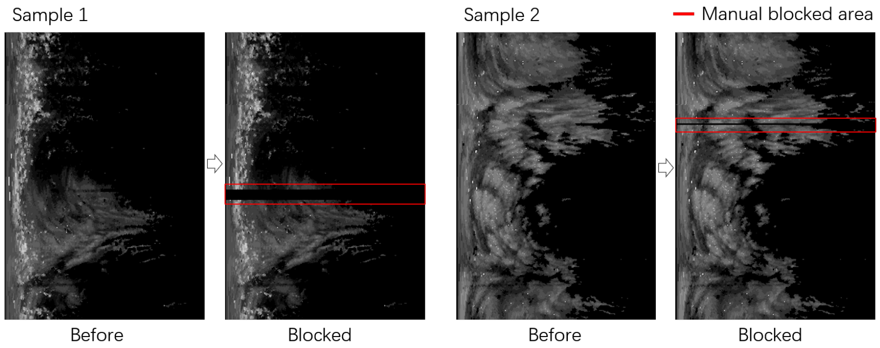


Fig. 11 Two manual block samples of Guangzhou Station

5.1.1 Network Structure

Fully Convolutional Network (FCN)[17] is a widely used model in the field of image semantic segmentation. It can classify each pixel in the image and determine the category of each point. Its segmentation effect has been remarkably improved compared with traditional methods. However, if the FCN model is directly applied to radar image correction, the correction effect is not fine enough and the accuracy is low. Based on the idea of FCN model, an improved model which names Residual Concatenate Fully Convolutional Neural Network (RC-FCN) is proposed for radar image correction.

Fig. 12 shows the basic structure of RC-FCN. It consists of an encoder and a decoder. The encoder part is composed of continuous residual convolution modules and the decoder part is composed of continuous de-convolution modules. The structure of each residual convolution module are same. As the feature map passes through each residual convolution module, its size will change to half of its original size. Detail introduction of residual convolution module will be introduced in next part.

Starting from the input layer, the feature map will be reduced to 1/32 of the original size after five times of residual convolution module. After several convolutions, the feature map will be de-convoluted five times again.In contrast to convolution, each de-convolution module will enlarge the size of the input feature map by twice. After five consecutive de-convolution, the feature map of the last convolution output is enlarged 32 times. The output will be restored to the original size. It should be noted that in order to improve the reconstruction performance of the feature map in the de-convolution process, the input feature map of each de-convolution operation contains not only the feature map output from the previous stage, but also the convolution feature map with the same size in symmetrical position. These two kinds of feature maps connect the two tensors together by means of concatenate. This structure is similar to FCN, but FCN only connects x32, x16 and x8. In this paper, through experiments, it is found that concatenating each set of symmetric convolution and de-convolution feature maps can improve the image reconstruction accuracy. In this paper, this kind of network is called CFCN. The letter, C, indicates that the concatenation between the encoder network and the decoder network is complete. In this way, it can effectively improve the network image reconstruction ability. After encoder network coding, the decoded output image of decoder is much more close to the original image. The detailed comparison can be seen in Table 3, and the specific analysis will be described in the next section.

Since the length and width of the original image are not necessarily 32 times, the image may not be able to be restored to the same size as the original image after five consecutive downsizing and five successive upsizing. Therefore, when the image is input the first convolution module, the length and width of the image should be adjusted by adjusted convolution module to the nearest size of which can be divided by 32 with no remainder. In this paper, the input image is 250 pixels wide and 360 pixels high. After adaptive convolution, the output image size is 256×384.

In this model, the image correction problem is regarded as a segmentation problem, and the essence is to classify each pixel one by one. Therefore, cross entropy loss is used as the loss function in this model. In order to achieve this conversion, each pixel of the image needs to be one-hot coded. As this is a 17 classification problem, each pixel is transformed into a one-hot vector with 17 dimensions. The loss function formula is shown in Formula 1.

$$\mathrm{Loss}(\mathbf{x}, \mathbf{y}) = \sum_{N}^{i=0} loss(x_i, y_i) \tag{1}$$

Where $\mathbf{x}$ represents all predicted pixels and $\mathbf{y}$ represents the index of real pixel class. N represents the number of pixels to be predicted. Detail of $loss(x_i, y_i)$ is shown as Formula 2.

$$\mathrm{loss}(x, \mathrm{class}) = -\log\left(\frac{\exp(x[\mathrm{class}])}{\sum_j \exp(x[j])}\right) \tag{2}$$

Where class means the index of pixel class which is from 0-17. The $x[\mathrm{class}]$ is negative log likelihood loss, which is calculated as Formula 3. It can be understood as next. For example, suppose $x = [1, 2, 3]$, $class = 2$, then $f(x, class) = -x[2] = -3$.

$$f(x, class) = -x[class] \tag{3}$$

5.1.2 Residual Convolutional Module Structure

A large number of experiments show that the residual convolution can effectively predict the blocking areas in the radar images. Detailed results can be seen in Table 4. Experiments show that, compared with VGG[27], GoogLeNet[29], DenseNet[10], the ResNet[7] has the best performance in image in-painting. In

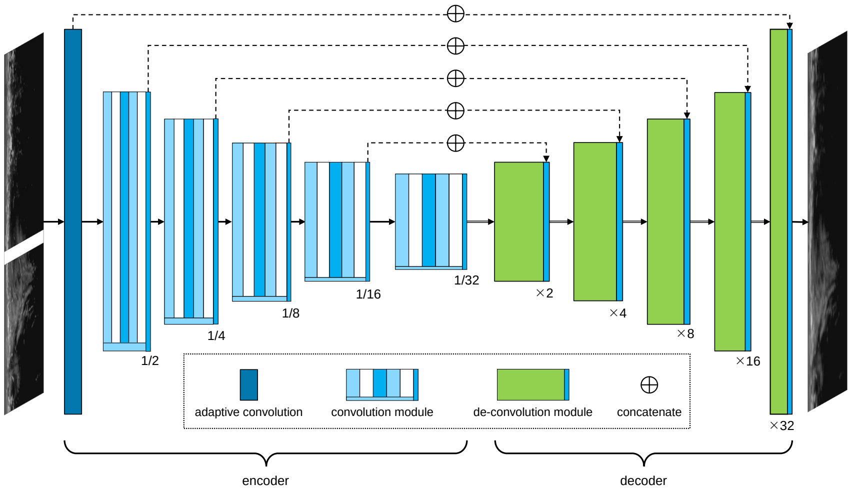


Fig. 12 Residual concatenate fully convolutional neural network

this study, the convolution module in the encoder is designed in the form of residual network. The structure of the residual module is shown in Figure 13.

In this progress, input is represented by x, and supposing that the target mapping which want to learn is $f(x)$. The part in the dotted line box in the Figure 13 needs to fit the residual mapping $f(x)-x$ of identical mapping. Residual mapping is often easier to optimize in practice because of its skipping connections. When the target mapping $f(x)$ is very close to identical mapping, the residual mapping is also easy to capture the subtle fluctuations of identical mapping. With residual module, inputs can forward propagate faster through the residual connections across layers.

6 Experiment

In this study, a lot of experiments have been done in early stage, and the performance of different models in image reconstruction and local blocking area prediction are studied.

6.1 Training and Test Detail

During the training stages, continuous weather radar image data are used. The data is from Guangzhou weather radar station detecting from 2011 to 2014 and NUIST weather radar station in the form of image. The data set mixes the images from two radar stations. The purpose of this is to improve the universality of the model after training, so that the model can correct the data from the two radar stations. Because the data to be processed by the model come from different radars in practical application, multi-source data need to be used for training, so that the model can have higher practical value.

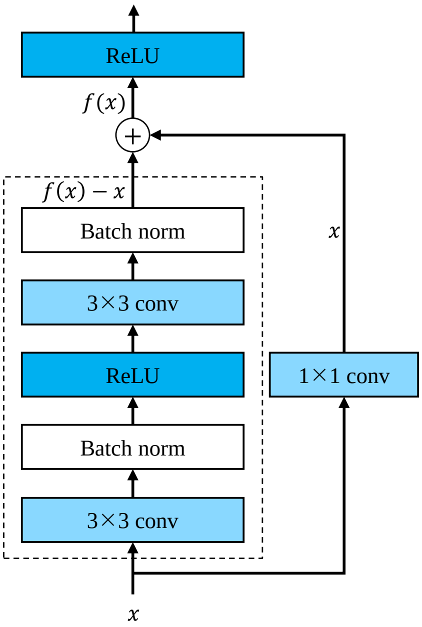


Fig. 13 Residual module structure

The images have been preprocessed in the way of section 3, and the blocking areas are generated randomly by the program. A total of 10000 weather radar images were used in the experiments, including 8000 for training and 2000 for testing. The size of the image used in the experiment is 360 height and 250 width. The position and width of the blocking areas are random. The minimum width is 3 pixels and the maximum is 16 pixels. The same data set was used for training and testing in all experiments. However, the number of training epochs of each model is different. General training until the loss value is no longer significantly reduced.

The left right of Fig. 14 shows the change of accuracy of training data set and test data set with the number of training epochs. The right side of Fig. 14 shows the change of loss during training. It can be seen from the figure that after about 800 epochs, the changes of accuracy and loss are significantly slowed down, but the overall change is still going better. However, in this process, sometimes the accuracy suddenly drops or the loss increases suddenly. With the continuous training, the frequency of this sudden change has a decreasing trend. In order to avoid this kind of accidental deterioration during testing, the model parameters are saved when the test accuracies of the test set reach the best value during the 1000 training epochs.

In the stage of network training, many groups of hyperparameters are tried in order to find the best parameters. For RC-FCN model, many attempts show that when the learning rate is 0.001, the mini-batch is 8, and the stochastic

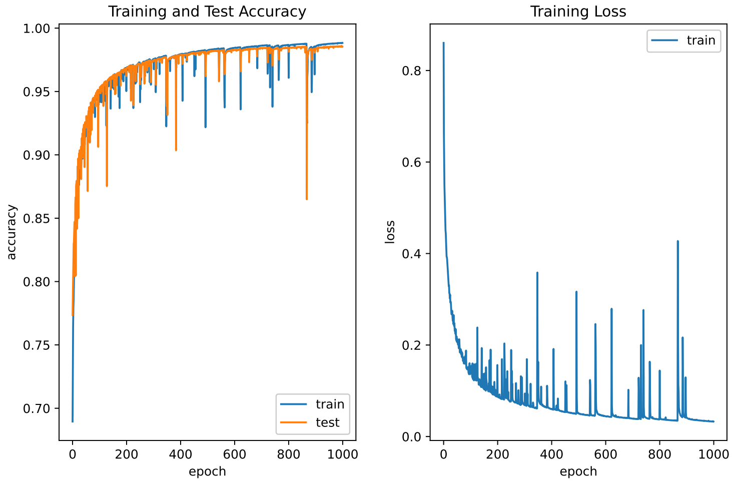


Fig. 14 Training and test accuracy and training loss changes

gradient descent algorithm is used for optimization, the prediction result of the model is the best. However, these hyperparameters are limited by the computational power of this study. The GPU video card memory used in the experiment is only 11GB, which can not accommodate a larger mini-batch. Maybe a larger mini-batch can get better training results.

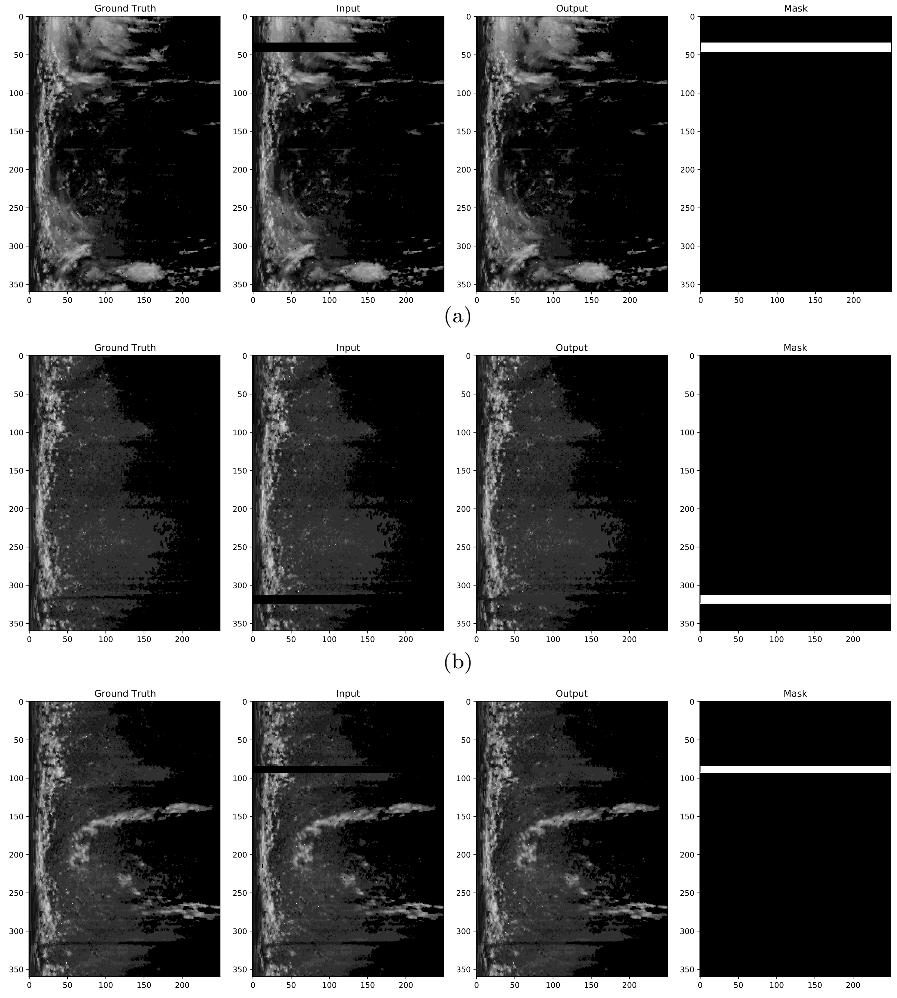
6.2 RC-FCN Experiment Comparison

The RC-FCN method proposed in this study is a deep improvement of FCN. Therefore, in the process of research, comparative experiments for each improvement have been carried out. At the same time, the results of other classical convolution neural networks in this study are also compared. The experiments mainly focus on two aspects, one is the ability of radar image restoration, the other is the ability of radar image beam blockage correction.

6.2.1 Evaluation Criterion

Beam blockage correction of weather radar is implemented from the perspective of image restoration and image segmentation methods. The accuracy, structural similarity (SSIM) and correlation are used to evaluate the effects of different models. On the other hand, because the network proposed in this study is mainly to solve the weather radar beam blockage correction problem, the standards in meteorological industry, which are CSI, FAR and POD, are used to evaluate the correction effects of each intensity segment. Detailed formulation can be found in [26, 25].

Fig. 15 A sample of beam blockage correction. The left first image is the ground truth. The second is the input image, which can be seen a rectangle black area in the top, that is the area to be corrected. The third is the output from RC-FCN. The corrected area can refer to the position corresponding to the white area of the forth. The horizontal axis is the radial distance. The coverage radius of radar data used in this study is 250km. The longitudinal axis is the radar deflection angle, a total of 360 integers.



(a)

(b)

(c)

6.2.2 Overall Radar Image Restoration

Fig 15 shows a set of radar images corrected by RC-FCN. Visual observation shows that the network model can reconstruct the original image well. Without careful observation, it is difficult to find the difference between input and output in the unblocked area. In order to show the image reconstruction ability of the RC-FCN more quantitatively, the results of evaluation criterion and the comparison with other networks are given in detail in Table. 3. This table also gives the reasons why FCN can be used as the basis for weather radar image correction. It can be seen from the table that FCN has excellent effect in image reconstruction. No matter CSI, FAR, POD, Accuracy, Correlation or SSIM, FCN ranked first or second. Therefore, based on FCN, this study attempts to propose a new deep learning network with excellent image reconstruction ability and good correction effect. However, the original FCN image reconstruction effect is not ideal, and the restored image appears mosaic like. Therefore, all the feature maps of the symmetric positions of FCN are concatenated, which is called CFCN in this study. The experimental results show that although CFCN does not achieve the best results in CSI, FAR and POD, it has achieved better results in overall accuracy, correlation and SSIM than FCN. But most importantly, CFCN has excellent performance in image reconstruction. It can reconstruct the original radar image completely without mosaic after encoding and decoding the input.

6.2.3 Radar Beam Blockage Correction

The ability of image reconstruction is not enough. The focus of this study is to achieve the data correction of blocking areas, that is, prediction. On the basis of CFCN, we try to add various convolution modules in the process of this study, such as dense module, residual module, inception module and VGG structure. Experiments show that the coding and decoding network based on ResNet is the best in image correction task. Therefore, this study combines DenseNet and ResNet to improve the network structure of CFCN. It should be noted that DenseNet's concatenation idea has been achieve in CFCN. CFCN has concatenated the feature maps of symmetric part of coder and decoder.

Table 4 shows the accuracy comparison of RC-FCN and other networks in radar beam blockage correction task. Experiments show that the coding and decoding network based on ResNet has achieved remarkable results in image correction task. After adding residual module to CFCN, some of the RC-FCN's evaluation criterion indexes achieve the best results, and some achieve the second best results. Most importantly, the improved RC-FCN has achieved the best or second best results in the accuracy, correlation and SSIM.

Fig.16 shows the comparison of the corrected results of blocking area. Each graph shows the radial data of radar echo reflectivity at one angle. The horizontal axis is the radial distance, and the vertical axis is the normalized reflectivity intensity. The blue line is the true value, and the red line is the predicted value of the network. Through observation, it can be found that the fit of predicted value and real value is high. Network can accurately predict the trend of data changes. There are some errors, but overall, the error is small.

Although RC-FCN failed to achieve the best or second best performance in every index, it achieved good fusion of radar image reconstruction task and radar image beam blockage correction task. The skipping connection added in the network also helps to improve the accuracy of the correction task.

Table 3 Overall accuracies of radar image reconstruction. Bold represents the best result and the underline represents the second best. It should be noted that some too high intensity signal,such 60-65,65-70, are very little in the data. They cannot import evaluation index calculation formulas for calculation. So, there is only 14 different dbzs in the result table. Table 4 is the same.

Index	dbz	Eval	Dense	Res	GoogLe	VGG	FCN	CFCN	RC-FCN*
13	55-60	CSI	0.2227	0.1345	0.1072	0.1235	0.5575	0.3951	**0.6228**
12	50-55		0.3927	0.2512	0.2178	0.2328	**0.8327**	0.6893	0.7864
11	45-50		0.4671	0.2842	0.2606	0.2761	**0.8505**	0.7768	0.8376
10	40-45		0.4591	0.2535	0.2256	0.2376	0.8420	0.7912	**0.8562**
9	35-40		0.4579	0.2539	0.2314	0.2400	0.8398	0.8132	**0.8735**
8	30-35		0.5116	0.3043	0.2859	0.2891	0.8440	0.8274	**0.8944**
7	25-30		0.5046	0.2780	0.2579	0.2619	0.8371	0.8235	**0.8912**
6	20-25		0.5338	0.3049	0.2847	0.2905	0.8535	0.8429	**0.9013**
5	15-20		0.5484	0.3137	0.2924	0.2972	0.8687	0.8609	**0.9206**
4	10-15		0.5896	0.3630	0.3422	0.3589	0.8913	0.8968	**0.9558**
3	5-10		0.6420	0.4322	0.4011	0.4286	0.9103	0.9266	**0.9641**
2	0-5		0.6175	0.3876	0.3599	0.3911	0.9020	0.9226	**0.9538**
1	-20-0		0.6770	0.4504	0.4243	0.4423	0.9073	0.9278	**0.9613**
0	NULL		0.9401	0.8823	0.8691	0.8822	0.9908	0.9935	**0.9952**
13	55-60	FAR	0.3087	0.5287	0.4925	0.5378	**0.1681**	0.2831	0.1868
12	50-55		0.3034	0.5014	0.4961	0.5217	**0.0996**	0.1871	0.1215
11	45-50		0.3218	0.5383	0.5556	0.5585	**0.0851**	0.1367	0.0935
10	40-45		0.3427	0.5799	0.5933	0.5948	0.0878	0.1200	**0.0765**
9	35-40		0.3402	0.5672	0.5878	0.5918	0.0828	0.1099	**0.0681**
8	30-35		0.3365	0.5521	0.5801	0.5726	0.0867	0.1024	**0.0567**
7	25-30		0.3340	0.5399	0.5670	0.5607	0.0833	0.0969	**0.0556**
6	20-25		0.3183	0.5394	0.5670	0.5586	0.0797	0.0822	**0.0509**
5	15-20		0.2970	0.5030	0.5435	0.5241	0.0681	0.0695	**0.0375**
4	10-15		0.2549	0.4440	0.4796	0.4680	0.0579	0.0488	**0.0206**
3	5-10		0.2204	0.3977	0.4153	0.4072	0.0456	0.0394	**0.0202**
2	0-5		0.2278	0.4133	0.4301	0.4209	0.0460	0.0373	**0.0200**
1	-20-0		0.1646	0.3601	0.3437	0.3235	0.0414	0.0302	**0.0191**
0	NULL		0.0342	0.0864	0.1013	0.0756	0.0065	0.0042	**0.0027**
13	55-60	POD	0.3284	0.2201	0.1650	0.2030	**0.8094**	0.4872	0.7353
12	50-55		0.4737	0.3402	0.2817	0.3172	**0.9164**	0.8181	0.8805
11	45-50		0.5999	0.4265	0.3894	0.4251	**0.9227**	0.8858	0.9152
10	40-45		0.6023	0.3893	0.3381	0.3644	0.9152	0.8860	**0.9204**
9	35-40		0.5957	0.3764	0.3432	0.3632	0.9071	0.9028	**0.9323**
8	30-35		0.6867	0.4821	0.4687	0.4669	0.9159	0.9127	**0.9448**
7	25-30		0.6711	0.4086	0.3869	0.3892	0.9044	0.9024	**0.9402**
6	20-25		0.7062	0.4692	0.4495	0.4542	0.9201	0.9111	**0.9467**
5	15-20		0.7069	0.4502	0.4382	0.4326	0.9262	0.9196	**0.9545**
4	10-15		0.7304	0.4929	0.4788	0.5041	0.9420	0.9398	**0.9751**
3	5-10		0.7775	0.5841	0.5373	0.5842	0.9511	0.9625	**0.9835**
2	0-5		0.7513	0.5266	0.4861	0.5377	0.9424	0.9563	**0.9724**
1	-20-0		0.7709	0.5972	0.5401	0.5588	0.9429	0.9552	**0.9793**
0	NULL		0.9724	0.9623	0.9630	0.9502	0.9974	0.9701	**0.9980**
Accuracy			0.8431	0.7202	0.7039	0.7151	0.9657	0.9831	**0.9831**
Correlation			0.9819	0.9584	0.9522	0.9577	0.9965	0.9966	**0.9977**
SSIM			0.8118	0.6775	0.6638	0.6735	0.9739	0.9770	**0.9848**

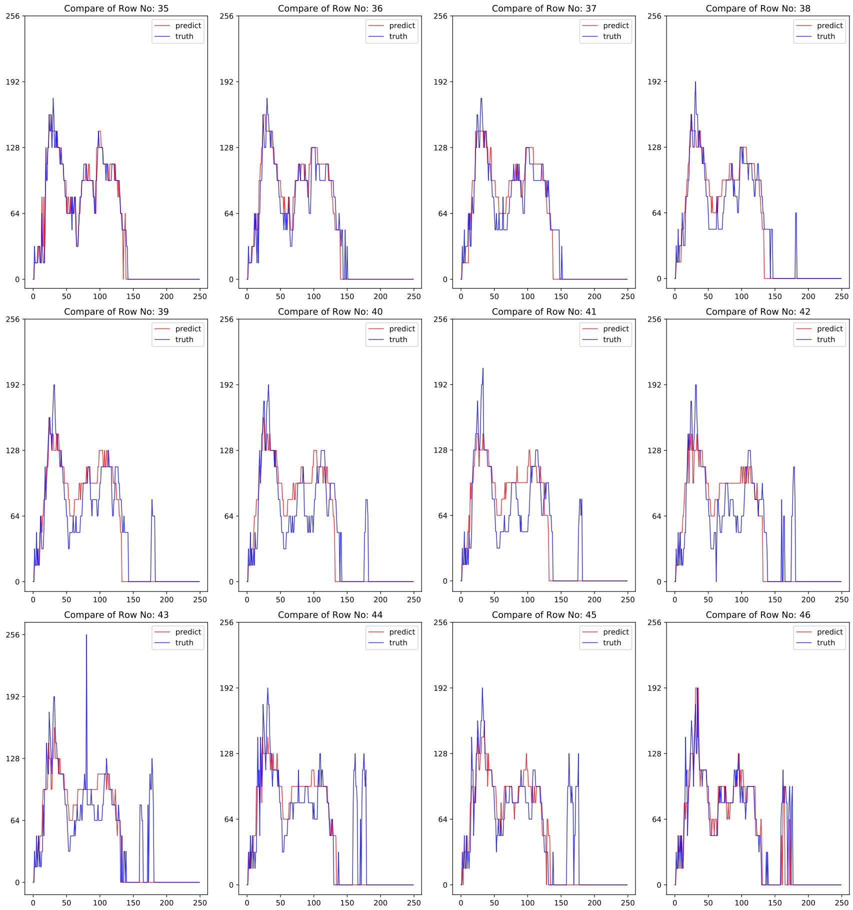


Fig. 16 Detail comparison of beam blockage correction

However, the prediction results of the model still have some disadvantages. As can be seen from Fig. 15, there is a sense of smearing in the target area of image prediction. This phenomenon shows that the general trend of the data is predicted correctly, but the precision is not high enough. This conclusion can be verified from Fig.16. In this figure, the red curve is the predicted value and the blue curve is the real value. It can be seen that the two curves overlap in general, but the complete overlap is relatively small. In addition, there are a few pixels that can not be effectively predicted. For example, in row 39 to row 42 of Fig.16, the data at about the 180th position do not be accurately predicted, and the predicted value is greatly different from the real value.

Table 4 Accuracies of corrected areas from generated images

Index	dbz	Eval	Dense	Res	GoogLe	VGG	FCN	CFCN	RC-FCN*
13	55-60	CSI	0.0150	**0.0258**	0.0167	0.0140	0.0066	0.0127	0.0114
12	50-55		0.0637	**0.0947**	0.0801	0.0737	0.0410	0.0149	0.0701
11	45-50		0.1320	**0.2189**	0.2084	0.1937	0.1109	0.0773	0.1079
10	40-45		0.1484	**0.2124**	0.2031	0.1919	0.1264	0.0915	0.1120
9	35-40		0.1351	**0.2030**	0.1891	0.1788	0.1253	0.1037	0.1253
8	30-35		0.1828	**0.2407**	0.2470	0.2292	0.1512	0.1381	0.1567
7	25-30		0.1628	**0.2200**	0.2179	0.2005	0.1513	0.1057	0.1222
6	20-25		0.1947	**0.2363**	0.2338	0.2180	0.1636	0.1229	0.1365
5	15-20		0.1903	**0.2180**	0.2214	0.2016	0.1641	0.0896	0.1044
4	10-15		0.2237	0.2463	0.2452	0.2351	0.2084	0.1214	**0.2678**
3	5-10		0.3007	0.3070	0.2989	0.2979	0.2579	0.3004	**0.4103**
2	0-5		0.2687	**0.2741**	0.2667	0.2650	0.2035	0.2603	0.2662
1	-20-0		0.3603	0.3682	0.3589	0.3545	0.2835	0.3343	**0.4385**
0	NULL		0.7668	0.7766	0.7659	0.7668	0.7662	0.8350	**0.8481**
13	55-60	FAR	0.0645	0.0428	0.0363	0.0447	0.0842	**0.0262**	0.1959
12	50-55		0.2544	**0.2094**	0.1712	0.1923	0.3807	0.2822	0.3383
11	45-50		0.5916	0.4913	**0.4679**	0.4966	0.6306	0.5698	0.6507
10	40-45		0.6850	0.5847	**0.5685**	0.6051	0.6841	0.6382	0.6996
9	35-40		0.6967	**0.5951**	0.6028	0.6335	0.7299	0.7737	0.7347
8	30-35		0.6956	**0.6050**	0.6118	0.6322	0.7318	0.7862	0.7170
7	25-30		0.6691	**0.5994**	0.6060	0.6268	0.7127	0.7805	0.7493
6	20-25		0.6769	**0.6121**	0.6172	0.6356	0.7016	0.7631	0.7502
5	15-20		0.6601	**0.6031**	0.6140	0.6200	0.6874	0.7448	0.7462
4	10-15		0.6055	**0.5582**	0.5723	0.5857	0.6616	0.6959	0.5656
3	5-10		0.5328	0.5029	0.5061	0.5204	0.5665	0.5829	**0.4628**
2	0-5		0.4831	**0.4655**	0.4699	0.4857	0.5117	0.5583	0.4910
1	-20-0		0.3415	0.3248	0.2900	**0.2809**	0.3254	0.4042	0.3707
0	NULL		0.1761	0.1751	0.1842	0.1684	0.1758	0.1159	**0.0983**
13	55-60	POD	0.0162	**0.0300**	0.0191	0.0170	0.0082	0.0162	0.0149
12	50-55		0.0829	**0.1278**	0.1032	0.0995	0.0631	0.0208	0.0885
11	45-50		0.2028	**0.3248**	0.3059	0.2933	0.1947	0.1162	0.1972
10	40-45		0.2387	**0.3285**	0.2999	0.2968	0.2191	0.1609	0.1868
9	35-40		0.2059	**0.3067**	0.2820	0.2769	0.2121	0.1898	0.2201
8	30-35		0.3238	0.3943	**0.4129**	0.3875	0.2709	0.3055	0.2823
7	25-30		0.2440	0.3326	**0.3335**	0.3054	0.2496	0.1758	0.2020
6	20-25		0.3288	0.3793	**0.3800**	0.3564	0.2732	0.2190	0.2437
5	15-20		0.2966	0.3260	**0.3417**	0.3012	0.2586	0.1311	0.1628
4	10-15		0.3282	0.3494	0.3574	0.3439	0.3400	0.1688	**0.3634**
3	5-10		0.4422	0.4366	0.4185	0.4277	0.3831	0.4637	**0.6027**
2	0-5		0.3756	0.3877	0.3736	0.3814	0.2835	**0.3879**	0.3672
1	-20-0		0.4671	0.4901	0.4553	0.4482	0.3600	0.4382	**0.5938**
0	NULL		0.9051	0.9171	0.9147	0.8967	0.9036	**0.9361**	0.9355
Accuracy			0.6460	0.6617	0.6551	0.6502	0.6277	0.6765	**0.7091**
Correlation			0.9111	0.9206	0.9186	0.9191	0.9129	0.9044	**0.9311**
SSIM			0.5883	**0.6335**	0.6290	0.6224	0.5826	0.6026	0.6324

7 Conclusion

In this study, a new edge-assisted cloud computing framework using RC-FCN is proposed. Its main task is to correct the beam blocking areas of radar images in the environment of Internet of Weather Radars. The framework is based on an edge-assisted cloud computing architecture, which can achieve large-scale weather radar beam blocking correction under the premise of high performance and elastic. A large number of comparative experiments show that the RC-FCN proposed in this paper can effectively correct the radar beam blockage areas and reconstruct the radar image. In addition, the most innovative part of the network is the use of semantic segmentation idea to achieve the correction of radar images. In order to meet the needs of semantic segmentation, a lot of preprocessing work has been done in this study, such as pixel value normalization and coordinate transformation. All processes are integrated into the edge-assisted cloud system which can be invoked remotely in a flexible and efficient way.

Acknowledgements This work has received funding from the Key Laboratory Foundation of National Defence Technology under Grant 61424010208, National Natural Science Foundation of China (No. 41911530242 and 41975142), 5150 Spring Specialists (05492018012 and 05762018039), Major Program of the National Social Science Fund of China (Grant No. 17ZDA092), 333 High-Level Talent Cultivation Project of Jiangsu Province (BRA2018332), Royal Society of Edinburgh, UK and China Natural Science Foundation Council (RSE Reference: 62967_Liu_2018_2) under their Joint International Projects funding scheme and basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20191398 and BK20180794).

References

1. Bi, R., Liu, Q., Ren, J., Tan, G.: Utility aware offloading for mobile-edge computing. Tsinghua Science and Technology **26**(2), 239–250 (2020)
2. Cai, W., Wei, Z.: Piigan: Generative adversarial networks for pluralistic image inpainting. IEEE Access **8**, 48451–48463 (2020)
3. Chen, H., Zhang, Y., Cao, Y., Xie, J.: Security issues and defensive approaches in deep learning frameworks. Tsinghua Science and Technology **26**(6), 894–905 (2021)
4. Chen, Y., Liu, L., Tao, J., Xia, R., Zhang, Q., Yang, K., Xiong, J., Chen, X.: The improved image inpainting algorithm via encoder and similarity constraint. The Visual Computer pp. 1–15 (2020)
5. Duc, T.L., Leiva, R.G., Casari, P., Östberg, P.O.: Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. ACM Computing Surveys (CSUR) **52**(5), 1–39 (2019)
6. Ham, Y.G., Kim, J.H., Luo, J.J.: Deep learning for multi-year enso forecasts. Nature **573**(7775), 568–572 (2019)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
8. He, Q., Cui, G., Zhang, X., Chen, F., Yang, Y.: A game-theoretical approach for user allocation in edge computing environment. IEEE Transactions on Parallel and Distributed Systems **PP**(99), 1–1 (2019)
9. Hongyan, W., Liping, L., Liping, H.: Beam blockage studies of doppler weather radar in mountainous region of zhejiang. Plateau Meteorology **33**(6), 1737–1747 (2014)
10. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708 (2017)
11. Kim, S., Hong, S., Joh, M., Song, S.k.: Deeprain: Convlstm network for precipitation prediction using multichannel radar data. arXiv preprint arXiv:1711.02316 (2017)

12. Lai, P., Qiang, H., Abdelrazek, M., Chen, F., Hosking, J., Grundy, J., Yun, Y.: Optimal edge user allocation in edge computing with variable sized vector bin packing. Springer, Cham (2018)
13. Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7760–7768 (2020)
14. Li, N., Wang, Z., Sun, K., Chu, Z., Leng, L., Lv, X.: A quality control method of ground-based weather radar data based on statistics. IEEE Transactions on Geoscience and Remote Sensing **56**(4), 2211–2219 (2017)
15. Li, P.C., Yu, T.T.: Landslide early warning with rainfall data from correcting weather radar reflectivity using machine learning. In: EGU General Assembly Conference Abstracts, p. 19265 (2020)
16. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 85–100 (2018)
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440 (2015)
18. Mabrouki, J., Azrour, M., Dhiba, D., Farhaoui, Y., El Hajjaji, S.: Iot-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. Big Data Mining and Analytics **4**(1), 25–32 (2021)
19. Malek, Y.N., Najib, M., Bakhouya, M., Essaaidi, M.: Multivariate deep learning approach for electric vehicle speed forecasting. Big Data Mining and Analytics **4**(1), 56–64 (2021)
20. McRoberts, D.B., Nielsen-Gammon, J.W.: Detecting beam blockage in radar-based precipitation estimates. Journal of Atmospheric and Oceanic Technology **34**(7), 1407–1422 (2017)
21. Miao, Y., Wu, G., Li, M., Ghoneim, A., Al-Rakhami, M., Hossain, M.S.: Intelligent task prediction and computation offloading based on mobile-edge cloud computing. Future Generation Computer Systems **102**, 925–931 (2020)
22. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)
23. Rafique, W., Qi, L., Yaqoob, I., Imran, M., Rasool, R.U., Dou, W.: Complementing iot services through software defined networking and edge computing: A comprehensive survey. IEEE Communications Surveys & Tutorials **22**(3), 1761–1804 (2020)
24. Ren, Z., Liu, Y., Shi, T., Xie, L., Zhou, Y., Zhai, J., Zhang, Y., Zhang, Y., Chen, W.: Aiperf: Automated machine learning as an ai-hpc benchmark. Big Data Mining and Analytics **4**(3), 208–220 (2021)
25. Shi, E., Li, Q., Gu, D., Zhao, Z.: A method of weather radar echo extrapolation based on convolutional neural networks. In: International Conference on Multimedia Modeling, pp. 16–28. Springer (2018)
26. Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.: Deep learning for precipitation nowcasting: A benchmark and a new model. In: Advances in neural information processing systems, pp. 5617–5627 (2017)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
28. Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., Jay Kuo, C.C.: Contextual-based image inpainting: Infer, match, and translate. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
29. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
30. Tian, H., Xu, X., Lin, T., Cheng, Y., Qian, C., Ren, L., Bilal, M.: Dima: Distributed cooperative microservice caching for internet of things in edge computing by deep reinforcement learning. World Wide Web pp. 1–24 (2021)
31. Tong, Z., Ye, F., Yan, M., Liu, H., Basodi, S.: A survey on algorithms for intelligent computing and smart city applications. Big Data Mining and Analytics **4**(3), 155–172 (2021)

32. Trebing, K., Stanczyk, T., Mehrkanoon, S.: Smaat-unet: Precipitation nowcasting using a small attention-unet architecture. Pattern Recognition Letters (2021)
33. Wang, N., Ma, S., Li, J., Zhang, Y., Zhang, L.: Multistage attention network for image inpainting. Pattern Recognition **106**, 107448 (2020)
34. Wang, T., Zhang, G., Liu, A., Bhuiyan, M.Z.A., Jin, Q.: A secure iot service architecture with an efficient balance dynamics based on cloud and edge computing. IEEE Internet of Things Journal **6**(3), 4831–4843 (2018)
35. Wei, D., Ning, H., Shi, F., Wan, Y., Xu, J., Yang, S., Zhu, L.: Dataflow management in the internet of things: Sensing, control, and security. Tsinghua Science and Technology **26**(6), 918–930 (2021)
36. Wu, H., Zhang, Z., Guan, C., Wolter, K., Xu, M.: Collaborate edge and cloud computing with distributed deep learning for smart city internet of things. IEEE Internet of Things Journal **7**(9), 8099–8110 (2020)
37. Wu, K., Shen, Y., Wang, S.: 3d convolutional neural network for regional precipitation nowcasting. Journal of Image and Signal Processing **7**(4), 200–212 (2018)
38. Xia, X., Chen, F., He, Q., Grundy, J.C., Abdelrazek, M., Jin, H.: Cost-effective app data distribution in edge computing. IEEE Transactions on Parallel and Distributed Systems **32**(1), 31–44 (2021). DOI 10.1109/TPDS.2020.3010521
39. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp. 802–810 (2015)
40. Xu, X., Fang, Z., Qi, L., Zhang, X., He, Q., Zhou, X.: Tripres: Traffic flow prediction driven resource reservation for multimedia iov with edge computing. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **17**(2), 1–21 (2021)
41. Xu, X., Fang, Z., Zhang, J., He, Q., Yu, D., Qi, L., Dou, W.: Edge content caching with deep spatiotemporal residual network for iov in smart city. ACM Transactions on Sensor Networks (TOSN) **17**(3), 1–33 (2021)
42. Xu, X., Huang, Q., Zhu, H., Sharma, S., Zhang, X., Qi, L., Bhuiyan, M.Z.A.: Secure service offloading for internet of vehicles in sdn-enabled mobile edge computing. IEEE Transactions on Intelligent Transportation Systems **22**(6), 3720–3729 (2020)
43. Yao, G., Liu, Z., Guo, X., Wei, C., Li, X., Chen, Z.: Prediction of weather radar images via a deep lstm for nowcasting. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2020)
44. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5505–5514 (2018)
45. Zamora-Izquierdo, M.A., Santa, J., Martínez, J.A., Martínez, V., Skarmeta, A.F.: Smart farming iot platform based on edge and cloud computing. Biosystems engineering **177**, 4–17 (2019)
46. Zhang, W., Chen, X., Jiang, J.: A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems. Tsinghua Science and Technology **26**(1), 95–111 (2020)