IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 28, 2019, accepted February 13, 2020, date of publication February 18, 2020, date of current version March 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974785

ResLap: Generating High-Resolution Climate Prediction Through Image Super-Resolution

JIANXIN CHENG[1], QIUMING KUANG[2], CHENKAI SHEN[1], JIN LIU[1], XICHENG TAN[3], AND WANG LIU[1]

[1]School of Computer, Wuhan University, Wuhan 430072, China
[2]Public Meteorological Service Center, China Meteorological Administration, Beijing 100081, China
[3]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China

Corresponding authors: Jin Liu (jinliu@whu.edu.cn) and Xicheng Tan (xctan@whu.edu.cn)

This research was supported by the National Key Research and Development Program of China (No. 2018YFC1507801, No. 2017YFB0504202, No. 2018YFB2100504); the National Science Foundation of China NSFC (No. 61972290, No. 41871312); the Hubei Natural Science Foundation (No. 2017CFB433); the Fundamental Research Funds for the Central Universities (No. 2042019kf0226); the Key Laboratory of Spatial Data Mining & Information Sharing of the Ministry of Education, Fuzhou University (No. 2016LSDMIS06) and the Beijing Key Laboratory of Urban Spatial Information Engineering (No. 2017209).

ABSTRACT In recent years, many models based on the convolutional neural network have achieved high-quality reconstruction for single image super-resolution. Meanwhile, many researches on image super-resolution have been applied to various fields. However, only a few research works have been applied to climate prediction. In this paper, we present ResLap to achieve high-resolution climate prediction. ResLap is a spatial downscaling method that converts low spatial resolution climate data into high-resolution regional climate forecasts. This method mainly introduces a novel residual dense block (RDB) into the Laplacian pyramid super-resolution network (LapSRN). Among them, we use LapSRN to achieve upsampling image reconstruction, and adopt RDB to fully extract the hierarchical features from all the convolutional layers. Extensive experimental results on benchmark climate datasets show that our new proposed model performs better than many super-resolution methods. Besides, the climate data are more complicated than the general image, because of its dynamic and chaotic nature. To facilitate model training, we integrate original climate data provided by the China Meteorological Administration, then convert it into trainable climate images. We also publish some climate image datasets online for research. Finally, we avoid the checkerboard artifacts in the generated high-resolution climate images.

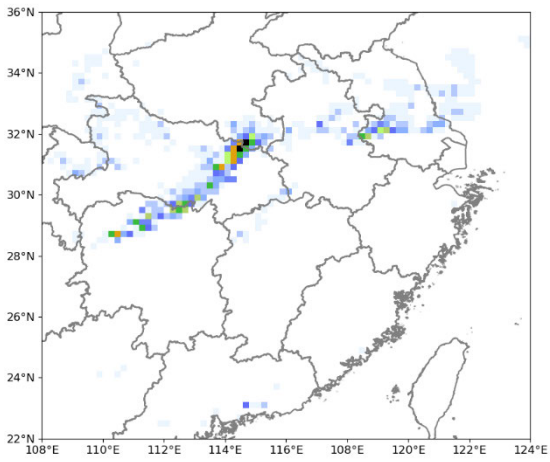
INDEX TERMS Super-resolution, climate image, checkerboard artifacts, convolutional neural network.

I. INTRODUCTION

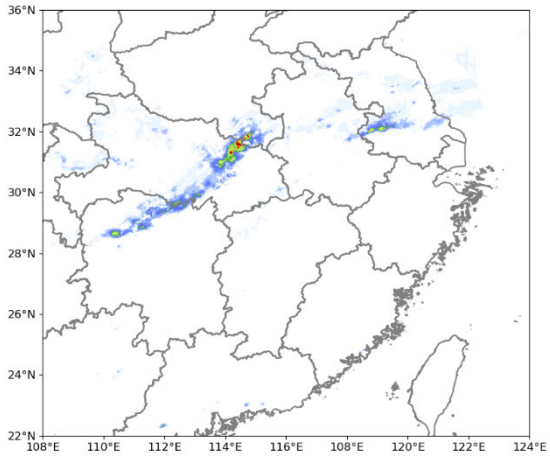
Image super-resolution [1]–[8] has always been a hot topic in computer vision, and it aims to generate a high-resolution image from a low-resolution input image. Many super-resolution methods have been proposed to reconstruct high-resolution images. For example, Early super-resolution methods are mostly example-based, such as the sparse-coding-based method [6], [9]. These algorithms mainly learn feature mapping functions from the low-resolution and high-resolution exemplar pairs. Later, Dong *et al.* [10] firstly introduced the convolutional neural network (CNN) [11] into image super-resolution. Then, they proposed the super-resolution convolutional neural network (SRCNN) and obtained further improvement over traditional methods. Thus, current methods widely apply CNN to achieve image super-resolution reconstruction, such as the Laplacian pyramid super-resolution network (LapSRN) [2], the very deep super-resolution network (VDSR) [1], and the efficient sub-pixel convolutional network (ESPCN) [7]. These methods have gained great performance for image super-resolution.

Many super-resolution models have been applied in various fields, such as medical examination [13] and traffic travel [14]. These methods have all achieved good results. Inspired by these successful cases, we directly consider applying the CNN based network to spatial downscaling methods [15]–[17] and achieve high-quality climate prediction. This method converts large-scale (or low-resolution) climate output information into small-scale (or high-resolution) regional climate prediction. For example, it can downscale the low-scale weather information at a *5km* hourly spatial resolution to *1km* low-scale climate forecasts, as shown in Fig 1. Namely, we can get a smaller range of precipitation

The associate editor coordinating the review of this manuscript and approving it for publication was Oguz Elibol.



(a) The low scale at 5km.



(b) The high scale at 1km.

FIGURE 1. The high-resolution precipitation prediction, and the clearer the generated image shows that we can get the precipitation forecast in a smaller area.

prediction through downsampling technology for the same area. In addition, climate data are more complicated than general images and many climate variables have large effects on weather conditions, such as topography, humidity, and pressure. We need to take full advantage of the potential information of these climate factors. Therefore, we integrate climate data into a single climate image to facilitate model training. Note that we get the original climate data from the public meteorological service center of China Meteorological Administration (CMA), and convert it into trainable climate images openly available to the public.

Downscaling technology can obtain high-resolution climate prediction from low-resolution spatial climate images [18], [19]. Vandal *et al.* [18] firstly introduced a three-layer CNN in the downscaling approach. This method, termed DeepSD, actually applied the SRCNN network. Then, researchers proposed to apply some advanced CNN based networks to the downscaling model, such as VDSR, ESPCN, and LapSRN. These methods had achieved further improvements than DeepSD in high-resolution climate prediction. However, these methods neglected to fully exploit hierarchical features [8] from any convolutional layers, thus obtained relatively-low performance in climate image super-resolution.

To address the drawback, we introduce a novel residual dense block (RDB) [8] to the LapSRN network. This method, termed ResLap, allows the current convolutional layer to directly access all of the previous convolutional layers. Thus, it can fully extract the hierarchical features from all convolutional layers. Furthermore, our RDB structure mainly includes contiguous memory and local feature fusion mechanism, different from RDB in the residual dense network (RDN) [8], as shown in section IV. D.

We adopt the LapSRN network to implement residual image reconstruction and generate high-quality climate prediction. Besides, we observe that many super-resolution models are easy to produce uncoordinated noise in the final generated images. For example, there are a series of strange checkerboard artifacts [20]–[22] emerging in the final output images. To solve this problem, we mainly focus on the operation of deconvolution [21], [23]–[25], and readjust the detail of the deconvolutional layer. Finally, the checkerboard artifacts are eliminated in the final generated images.

In summary, the contributions of this work are mainly three-folds:

1) We innovatively introduce the RDB structure into the LapSRN network, which can make full use of the hierarchical features from the original input images. The new proposed network can generate multiple high-quality super-resolution predictions through progressive reconstruction.
2) We propose a new method to eliminate the checkerboard artifacts in the final generated images. This method allows the generated image without losing any climate detail. Therefore, it can improve the accuracy of climate prediction.
3) We propose a new spatial downscaling method, termed ResLap, to generate high-resolution climate prediction. This method can convert various large-scale (or low-resolution) climate output information into small-scale (or high-resolution) regional climate prediction.

II. RELATED WORK

As mentioned previously, climate data are more complicated than the general image, and include predictions for multiple climate patterns, such as precipitation, temperature, and humidity. We mainly focus on precipitation prediction in the paper. Besides, the precipitation information is spatial climate data that are affected by many climate variables, such as topography, pressure, and humid. Thus, our goal is to downgrade low-scale precipitation information to high-scale precipitation prediction in the spatial range by the downsampling method, as shown in Fig. 1.

Early algorithms mainly adopted dynamical and statistical modeling [15], [16] to achieve precipitation prediction, such as linear models (Bilinear) [26] and support vector machines [27]. These traditional methods only learned a func-

TABLE 1. The comparisons of six methods: Bilinear, DeepSD, ESPCN, VDSR, RDN, LapSRN, and the proposed ResLap. The number of layers consists of both convolution and deconvolution. For example, our ResLap network includes 37 convolutional layers and 4 deconvolutional layers.

Model	Upsampling Operation	Layers	Residual learning	Reconstruction	Loss function
Bilinear	Bilinear Interpolation	1	No	Direct	L2
DeepSD	Sub-Pixel Convolution	4	No	Direct	L2
ESPCN	Sub-Pixel Convolution	5	No	Direct	L2
VDSR	Sub-Pixel Convolution	22	Local Residual Learning	Direct	L2
RDN	Up-Sampling Net	25	Residual Dense Block	Direct	L1
LapSRN	Transposed Convolution	27	Local Residual Learning	Progressive	Charbonnier
ResLap (ours)	**Transposed Convolution**	**41**	**Residual Dense Block**	**Progressive**	**MRMSE**

tion or model from many observed low-resolution and high-resolution precipitation data pairs, which were challenged by the difficulties of effectively modeling the data. Furthermore, these approaches did not exploit the spatial structural information of the precipitation data, thus achieved relatively-low performance in the precipitation forecasts.

To solve the drawback, Vandal *et al.* [18] proposed the DeepSD, and first applied the CNN network to the spatial downscaling model. In addition, they integrated complex precipitation data into a single image by a computer vision method, which could make use of the potential information of the precipitation variables. We can regard that each pixel on the climate image represents the precipitation information of the corresponding area. Thus, our task is to increase the number of pixels in the climate image by the downsampling method. Actually, the downscaling process is similar to image super-resolution in computer vision. Besides, the better the quality of the generated climate image is, the better the effect of climate prediction is.

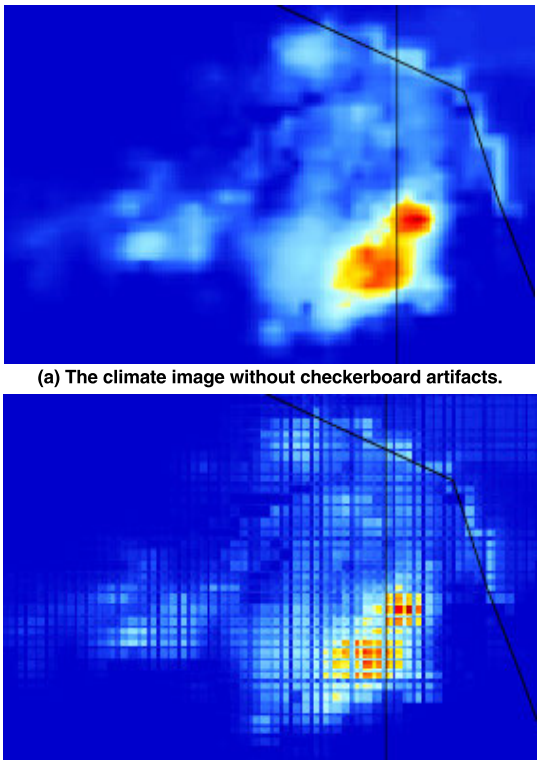
Inspired by the DeepSD algorithm, researchers applied many CNN based models to improve the quality of the generated climate images. For example, the VDSR model increases the network depth from 3 convolutional layers to 20 convolutional layers. It implements a deeper network than SRCNN. Morethan, VDSR predicts the residual of the image, not the actual pixel value. Thereby, it can accelerate the convergence speed to train a deeper model. The ESPCN network extracts feature maps from the low-resolution image with efficient sub-pixel convolution, instead of deconvolution upsampling operation. LapSRN adopts the local residual learning to exploit more hierarchical features from the original input image. It upscales the low-resolution image to the target scale with the transposed convolutional layer, instead of bicubic interpolation. Besides, RDN uses the residual dense learning to extract feature information from input pictures.

We compare all the network structures of Bilinear, DeepSD, VDSR, ESPCN, RDN, LapSRN, and our ResLap, as shown in Table 1. Among them, most methods neglected to fully exploit hierarchical features from each convolutional layer, except for RDN. In contrast, our proposed approach introduces the RDB framework to efficiently extract feature maps from all convolutional layers in the low-resolution images. Although RDN also utilizes RDB to extract image features, our model is still different from RDN. The differences between RDN and ResLap would be summarized in Section IV. Besides, we use the multi-scale root mean square error (MRMSE) to optimize the deep network, instead of the Charbonnier loss function [2].

We have observed that a series of strange checkerboard artifacts appear in the generated climate images, as shown in Fig. 2. Odena *et al.* [21] demonstrated that these artifacts are caused by the deconvolution operation, as shown in Fig. 3. This operation allows the model to draw a series of points in a large image using each point in the small image. However, deconvolution can easily produce uneven overlap, placing more shadows in some places than others. It causes characteristic checkerboard artifacts appearing in the final images. To avoid these artifacts, Odena *et al.* [21] focused on the relationship between the deconvolutional kernel size (*deker*) and the length of the *deker* moving on the image (*stride*). They proposed that if *deker* is divided by stride, the overlap issue can be removed. Although this method is helpful to alleviate the overlap issue, it is still easy for deconvolution to create checkerboard artifacts. To address this drawback, we present a new approach to eliminate the artifacts by adjusting the relationship between *deker* and *stride*.

III. METHODOLOGY

In this paper, we propose ResLap to generate high-resolution climate images from the low-resolution images. ResLap mainly replaces simple residual learning in LapSRN with the RDB structure, and our RDB structure is different from RDB in the RDN. We use the MRMSE loss function to optimize the model. Besides, we also present an approach to avoid the checkerboard artifacts emerging in the final climate images. Finally, we use a neural network to integrate all precipitation variables into a single climate image.



(a) The climate image without checkerboard artifacts.

(b) The climate image with checkerboard artifacts.

FIGURE 2. The two forms of climate image.

A. THE RESIDUAL DENSE BLOCK

Currently, we mainly improve the performance of the super-resolution models by increasing the depth of the network layers. However, there is a shortcoming when the network layers become very deep, the model will be difficult to train. Fortunately, the residual network can accelerate the convergence speed to train a deeper model, such as the residual block [28], the dense block [29], and the residual dense block [8]. Compared with the residual block and the dense block, the residual dense block has better performance in extracting hierarchical features from the input images. Therefore, we apply the RDB structure to our model and make some adjustments to it, as shown in Fig. 4. Furthermore, our RDB includes two parts: contiguous memory and local feature fusion.

1) CONTIGUOUS MEMORY

we use the dense connected convolutional layer to extraction features and allow each layer to be directly connected to all subsequent layers. Thus, the input of the local convolution layer is the output of all the previous convolutional layers. RD_{d-1} and RD_d refers to the input and output of the *d-th* RDB, respectively, and both have m feature maps, we empirically set m to between 5 and 20. Let $RD_{d,c}$ be the output of the *c-th* convolutional layer of the *d-th* RDB, and it can be represented as:

$$RD_{d,c} = relu(W_{d,c}\left[RD_{d-1}, RD_{d,1}, \cdots, RD_{d,c-1}\right]),$$

where *relu* and $W_{d,c}$ are the ReLU activation function and the weights of the *c-th* convolutional layer, respectively.

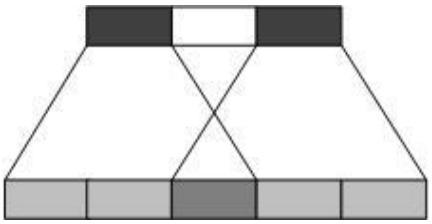


FIGURE 3. The movement process of deconvolution kernel, and the uneven overlap can cause checkerboard artifacts in the final images.

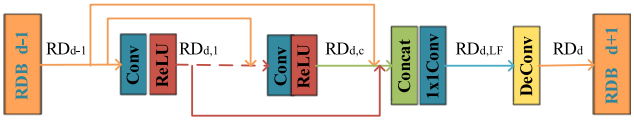


FIGURE 4. The structure of our RDB. Each RDB contains 16 3 × 3 convolutional layers, 1 1 × 1 convolutional layer, and 1 4 × 4 deconvolutional layer.

$\left[RD_{d-1}, RD_{d,1}, \cdots, RD_{d,c-1}\right]$ is the concatenation of the feature maps generated by all subsequent RDB, which can efficiently extract local dense feature.

2) LOCAL FEATURE FUSION

As mentioned above, the feature maps of the convolutional layers 1, …, (*c-1*) in the (*d-1*)*-th* RDB are directly used as input to the *d-th* RDB in a concatenation way. Thereby, we need to decrease the number of feature maps to make the model training faster. We apply a 1 × 1 convolutional layer after the *c-th* convolutional layer to control the output information. This operation, termed local feature fusion (LFF), can be expressed as:

$$RD_{d,LF} = F^d_{LFF}\left(\left[RD_{d-1}, RD_{d,1}, \cdots, RD_{d,c-1}, RD_{d,c}\right]\right),$$

where F^d_{LFF} refers to the function of the 1 × 1 convolutional layer in the *d-th* RDB.

The main difference between our RDB and the original residual dense block is that our RDB does not carry the local residual learning, and we implement the local residual learning in the image reconstruction. Thus, we can attain different scales of high-resolution images at the corresponding level.

B. NETWORK STRUCTURE

We build the ResLap model by combining the LapSRN and RDB, as shown in Fig. 5. Our ResLap mainly consists of two parts: feature extraction and image reconstruction. Among them, we use RDB to exploit hierarchical features from each convolutional layer in the feature extraction. We adopt local residual learning to complete the fusion of the upsampling image at each level in the image reconstruction. Besides, let I_{LR} and I_{SR} be the input and output of ResLap, respectively. The $conv(\cdot)$ and $trans(\cdot)$ are the operations of the convolution and the deconvolution, respectively.

In the experiments, our ResLap model is a two-level structure network, due to limited computing resources. For example, a *2km* low-resolution climate picture will generate *5km* and *10km* high-resolution pictures in turn, which generate

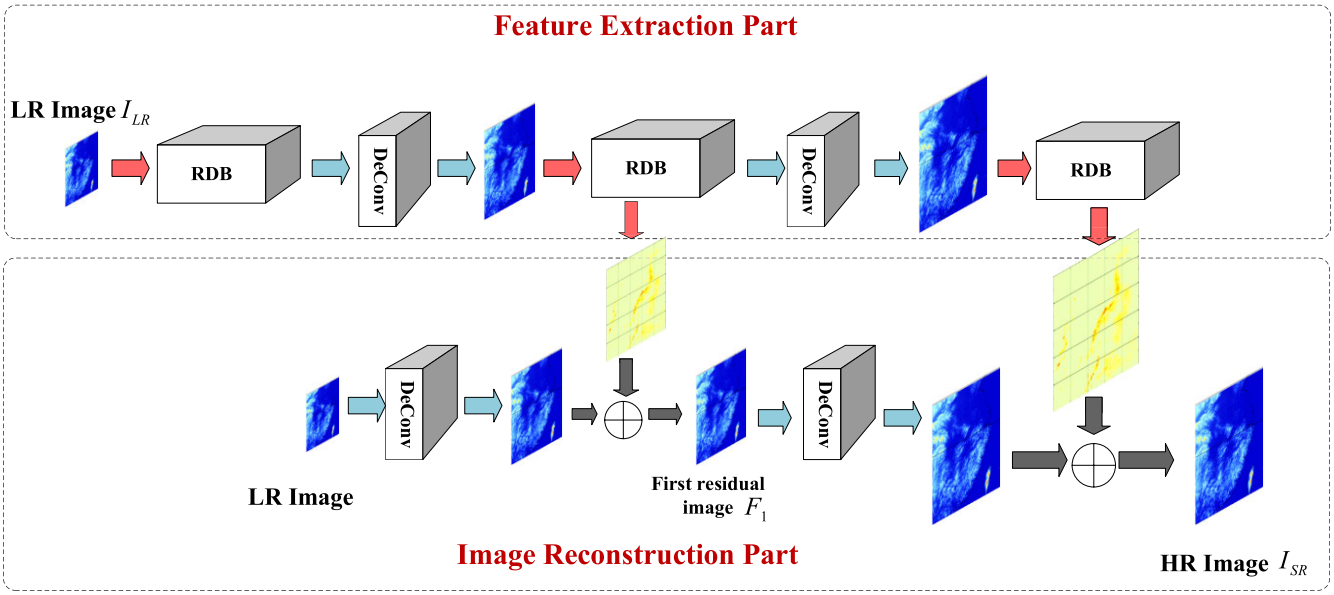


FIGURE 5. The structure of our ResLap network. ResLap is a two-level structure network, and each level contains a feature extraction and an image reconstruction. Besides, ResLap has two upsampling operations.

the final climate forecast. Furthermore, each level contains a feature extraction and an image reconstruction, and the corresponding scale of the reconstructed image is generated at each level.

Level 1 is expressed as $F_1 = F_{0,T} + conv(RD_{1,T})$, where $F_{0,T}$ refers to the upsampling image of the input low-resolution image passing through a transposed convolutional layer [30], and F_1 can be defined as $F_{0,T} = trans(I_{LR})$. $RD_{1,T}$ is the upsampling image of the output of the first RDB passing through a transposed convolutional layer, and $RD_{1,T}$ is expressed as $RD_{1,T} = trans(RD_{1,LF})$. Morethan, RD_1 is the output of the first RDB and is denoted as $RD_1 = rdb(RD_0)$, where $rdb(\cdot d)$ refers to the operation of the RDB. The output of the upsampling image $RD_{1,T}$ is not only the input of the second RDB, but also the input of a convolutional layer for extracting feature from $RD_{1,T}$. Then we take $RD_{1,T}$ through a convolutional layer to get the feature maps $conv(RD_{1,T})$. Finally, we achieve the upsampling image reconstruction, and combine these high-resolution images $F_{0,T}$ and $conv(RD_{1,T})$ to get the first residual image F_1. Furthermore, we emphasize that the size of the convolution kernel and the deconvolution kernel are the same.

Correspondingly, level 2 is the same operation as layer 1, and is defined as $I_{SR} = F_{1,T} + conv\left(RD_{2,T}\right)$, where $F_{1,T}$ refers to the upsampling image of the first residual image F_1 passing through a transposed convolutional layer, and $F_{1,T}$ can be defined as $F_{1,T} = trans(F_1)$. $RD_{2,T}$ is the upsampling image of the output of the second RDB passing through a transposed convolutional layer. $RD_{2,T}$ is expressed as $RD_{2,T} = trans(RD_{2,LF})$. RD_2 is the output of the second RDB, it is denoted as $RD_2 = rdb(RD_1)$. Then we use a convolutional layer to extract feature from $RD_{2,T}$. Finally, we achieve the upsampling image reconstruction, and combine $F_{1,T}$ and $conv(RD_{2,T})$ to get the final output image I_{SR}.

C. THE LOSS FUNCTION

The ResLap algorithm generates a corresponding-size climate prediction at each level. Therefore, we need to compute the loss between the generated high-resolution image and the truth-ground image at all levels, and the proposed MRMSE function is to calculate the sum of the mean square error [31] of each level.

Let x and θ be the input low-resolution image and the set of model parameters to be optimized, respectively. x_s and r_s are the upscaled low-resolution image and the residual image at level s, respectively. $\hat{y}_s$ refers to the generated output high-resolution image at each level s, and is represented as $\hat{y}_s = x_s + r_s$. We denote the corresponding truth high-resolution image at each level s by y_s, and adopt the scipy ndimage function to downsample the ground truth HR image y to y_s at each level. Furthermore, our target is to make the generated output image $\hat{y}_s$ close to the corresponding truth image y_s at each level through the proposed loss function. The MRMSE loss function is defined as:

$$\varphi\left(\hat{y}, y; \theta\right) = \frac{1}{L}\sum_{s=1}^{L}\left(\left(\hat{y}_s - y_s\right)^2 + \delta\right),$$

where L refers to the number of the network levels, and we set L to 2 in this work. δ is an influential factor constant, and we empirically set δ to 0.001. In addition, we adopt the corresponding truth high-resolution image y_s at each level as supervision. The supervision allows our model to predict the residual images and generate corresponding-size output images at each level, which is helpful for us to learn the internal mechanism of the model.

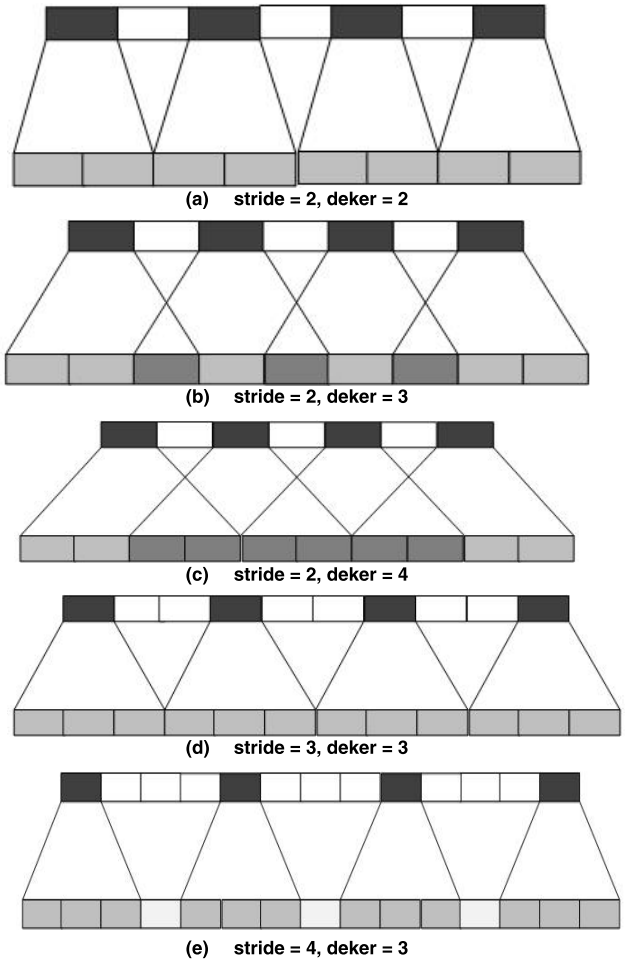


FIGURE 6. Different sizes of the stride and deker.

D. THE WAY TO AVOID CHECKERBOARD ARTIFACTS

As mentioned previously, to avoid checkerboard artifacts appearing in the final climate images, the model should solve the problem of uneven overlap. Thus, we mainly focus on the relationship between the sizes of the *deker* and the *stride*, and this relationship directly depends on the quality of the final generated climate image, as shown in Fig. 6. Besides, we observe that different sizes of *deker* and *stride* will cause different effects on the final image. For example, when *stride* is smaller than *deker*, uneven overlap will appear in the final generated image, as shown in Fig. 6(b) and Fig. 6(c). When *stride* is greater than *deker*, the sparse stripes will appear in the final generated image, as shown in Fig. 6(e). Only when *stride* is equal to *deker*, no uneven overlap and sparse stripes appear in the generated image.

In this paper, we propose that if the condition $stride = deker$ is satisfied, the problems of uneven overlap and sparse stripes can be perfectly solved, as shown in Fig. 6(a) and Fig. 6(d). This is equivalent to sub-pixel convolution [7], [20], [25], a technique that also has a great performance in image super-resolution. Furthermore, we also test two thousand samples of different sizes of the *deker* and the *stride*, the experiment results show that if the *stride* is equal to the *deker*, the corresponding checkerboard pattern will not appear in the picture. These results further prove that our proposed method can solve the problem of uneven overlap.

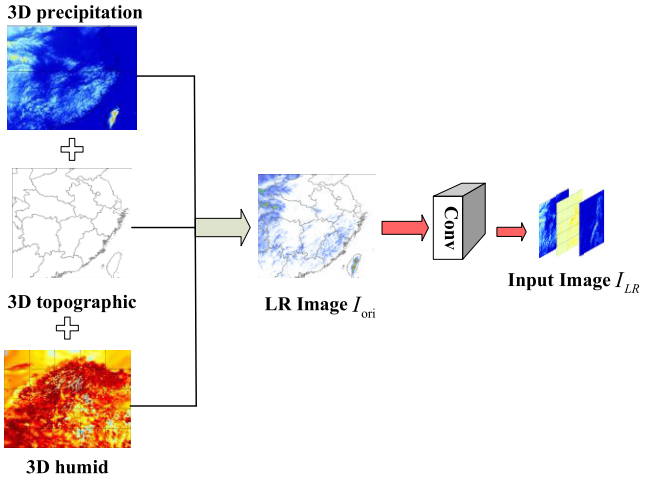


FIGURE 7. The fusion process of the input image. The input is a 4D low-resolution image, and we get input by combining 3D precipitation, 3D topographic, and 3D humid data.

E. PRE-PROCESSING CLIMATE DATA

The precipitation data are more complex than the general images, and the high-resolution precipitation prediction is affected by many climate factors, such as topography, humid, and temperature. To generate the high-resolution precipitation prediction, we should take advantage of valuable information on these climate variables. It is well known that the color images contain channels such as red, green, and blue. Inspired by it, the precipitation data can be represented analogously, and we regard each precipitation variable as an image channel [32]. Finally, we attain the multi-channel climate images for training.

In the experiments, we assume that the topography and humid variables have large effects on the precipitation prediction. To get the model input climate images, we first obtain low-resolution 3D precipitation data, 3D humid data, and 2D topographic data from the same area through CMA. Then, we will integrate these two variable data into the low-resolution precipitation data. However, the dimensions of these data are not the same, and we should use the reshape function to upgrade the 2D topographic data into 3D topographic data. Next, we treat each type of climate data as a channel for climate images, and get the 4D three-channel climate images I_{ori}, as shown in Fig. 7. Finally, we use a convolutional layer to extract shallow hierarchical features from the three-channel images I_{ori} to get the model input images I_{LR}. The model input images can be obtained by $I_{LR} = conv(I_{ori})$, where $conv(\cdot)$ denotes convolution operation. Therefore, our goal is to generate high-resolution climate images I_{SR} from low-resolution images I_{LR} using the ResLap network.

IV. EXPERIMENTS

We get the original climate data from the CMA and integrate it into a training image set of models, this is our unique

contribution. Besides, our model has three downscaling modes, such as EC (*2km*), NMC (*5km*), and GRAPE_MESO (*10km*). Finally, we compare the proposed ResLap network with six common super-resolution algorithms: Bilinear, DeepSD, VDSR, ESPCN, RDN, and LapSRN. We also set five corresponding meteorological indicators to evaluate the performance of these models.

A. DATASETS FOR TRAINING AND TESTING

We get all the original climate data from the CMA. Among them, the precipitation data record 1 hour of precipitation information over a latitude and longitude region. The humid data record 1 hour of humid change information in the same latitude and longitude. The terrain data note the latitude and longitude coordinates corresponding to the precipitation and humid. Then we integrate these three data into climate images, and these climate images will be used as training and test sets for the model.

We carry out extensive experiments using two region datasets: North China (NCN) and South China (SCN). These climate datasets record precipitation information from different parts of the country. Note that the area of each divided region is different, thus the size of the climate image of each data set is different. For example, the size of a high-resolution climate image in the NCN is 1400×1600, while the size of a high-resolution image in the SCN is 1200×2000.

We randomly select 1200 climate images from the annual precipitation observation data of each region. Each climate image records 1 hour of precipitation information in a certain area. Among them, we train all of our networks with 1000 training images and use 100 validation images in the training process. We evaluate the performance of each model with 100 testing images.

B. TRAINING DETAILS

The ResLap model is a two-level structure network, as shown in Fig. 5. Each level has 17 convolutional layers and 2 deconvolutional layers. Among them, each convolutional layer consists of 32 filters with a size of 3×3 in the RDB. Another convolutional layer consists of one filter with a size of 1×1, it is used to control the number of feature maps of the image. Besides, the size of the all deconvolution filter is 4×4, and all the convolutional and the deconvolutional layers are followed by a rectified linear unit. We implement the ResLap network with the TensorFlow framework [33]. Furthermore, we optimize ResLap with the MRMSE loss function and update it with Adam optimizer [34]. The learning rate is initialized to 0.0001 for all layers, and the batch size is set to 32. Finally, we train all models with a Titan Xp GPU for 100 epochs and it roughly takes one day to complete the training process.

Our model has three downscaling modes: EC (*2km*), NMC (*5km*), and GRAPE_MESO (*10km*). We produce high-resolution images in the corresponding mode and compare the results generated by different models. For example, NMC is that we downscale the hourly precipitation data at a *5km* spatial resolution to the precipitation data at a 1km spatial resolution. To train the model, we first get the original high-resolution climate images with the size of 1400×1600 as the label. Then, we downscale the original label data by five times into low-resolution climate images. The low-resolution images with the size of 280×320 are used as the input of the model. Finally, we use the input data to train all models. The final generated results are evaluated with five metrics.

To eliminate the checkerboard artifacts in the final generated images, we apply the newly proposed method to two models: LapSRN and ResLap. Our new method only discusses the relationship between the *deker* and the *stride.* Thus, we take three different sizes of the *deker* and the *stride*. Among them, the first case is that *deker* is equal to *stride*. The second case is that *deker* is greater than *stride*, which can be expressed as $deker = stride + 1$. The last case is that *deker* is less than *stride*, which is defined as $dekel = stride - 1$. For testing, we only adopt the NMC mode and use two benchmark datasets: NCN and SCN.

C. THE EVALUATION METRICS

We evaluate the performance of all models using five metrics: root mean squared error (RMSE), prediction correction (PC), prediction omission (PO), false alarm ratio (FAR), and threat score (TS) [35]. The first metric is the common image evaluation indicator, and the last four metrics are climate data evaluation indicators. These four indicators are calculated as follows:

$$PC = \frac{NA + ND}{NA + NB + NC + ND},$$
$$PO = \frac{NC}{NA + NC},$$
$$FAR = \frac{NB}{NA + NB},$$
$$\mathrm{TS} = \frac{\sum_{day=1}^{day=n} NA}{\sum_{day=1}^{day=n} (NA + NB + NC)}.$$

As shown in Table 2, *NA* represents the number of stations where the prediction is the same as the actual precipitation in the test area. *NB* indicates the number of stations where the prediction does not appear in reality. *NC* indicates the number of stations that are not predicted and appear in reality. *ND* represents the number of stations that predicted to be the same as the actual sunny days in the test area, and *n* refers to the number of days of scoring. Actually, these indicators are somewhat similar to the ROC curve values in machine learning.

We divide 4 precipitation levels according to the amount of rainfall: light rain ($\geqslant$0.1mm), moderate rain ($\geqslant$10mm), heavy rain ($\geqslant$25mm), and torrential rain ($\geqslant$50mm). Therefore, our TS score has four corresponding values. Furthermore, It is known that the higher value of RMSE denotes the better quality of the generated high-resolution image. The higher value of the TS score denotes the more accurate the precipitation level prediction of the model. The larger value of the PC score

TABLE 2. Precipitation test indicator.

Confusion matrix		Prediction	
		Yes	No
Reality	Yes	NA	NC
	No	NB	ND

denotes the more precise the forecast for sunny and rainy days. The smaller values of the PO and FAR score denote the better the effectiveness of the climate prediction.

D. DIFFERENCES FROM PREVIOUS MODELS

1) DIFFERENCE TO LAPSRN

Inspired by LapSRN, our newly designed model is also divided into two parts: feature extraction and image reconstruction. Among them, LapSRN uses a series of continuous convolutional layers to achieve feature extraction, and this way neglected to fully exploit hierarchical features from each convolutional layer. However, ResLap introduces RDB structure to extract image features, rather than a series of continuous convolutional layers. Our network allows the current convolutional layer to directly access all of the previous convolutional layers. Thus, it can fully extract the hierarchical features from all convolutional layers and it can add more convolutional layers to extract features. In addition, LapSRN utilizes the Charbonnier loss function to optimize the deep network, which is to calculate the L_1 loss for each level structure. In contrast, ResLap uses the MRMSE function to calculate the total loss of the entire model, which is to calculate the L_2 loss for each level. In fact, extensive experiments have proved that L_2 loss function is more powerful for performance and convergence in climate prediction.

2) DIFFERENCE TO RDN

There are two main differences between RDN and our ResLap network. The first one is that our model can generate multiple high-resolution images through progressive reconstruction using the Laplacian pyramid. However, RDN only gets the final high-resolution image through one upsampling operation. It is well known that climate prediction is a complex task and we need multi-scale climate prediction in daily life. Fortunately, our model can greatly simplify this operation. Besides, we calculate the total loss of high-resolution images and truth images at different scales, which can make the model converge faster. The second one is that our RDB does not achieve the local residual learning, and we implement the local residual learning in the image reconstruction. Thus, we can attain different scales of high-resolution images at the corresponding level. For example, we get the output of the *d-th* RDB by combining the input of the *d-th* RDB and the fusion image $RD_{d,LF}$ in the original RDB. However, we will perform upsampling operations on the input of the *d-th* RDB and $RD_{d,LF}$ in the new RDB, respectively. Then, we implement residual image reconstruction at the corresponding level.

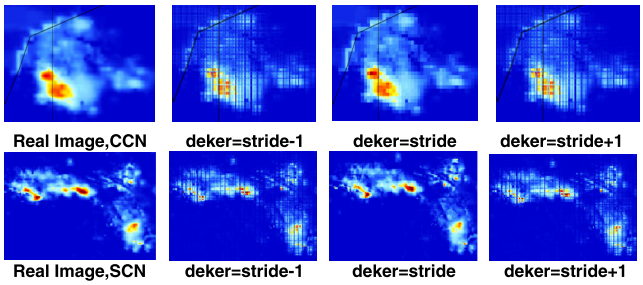


FIGURE 8. Experimental results of LapSRN, NMC.

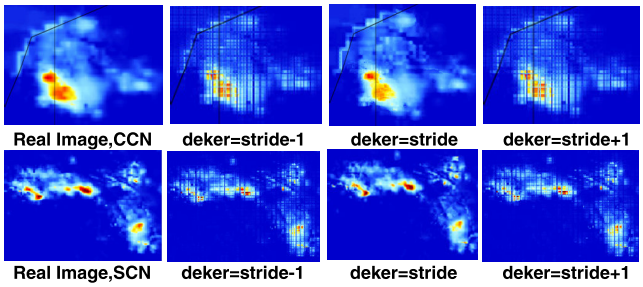


FIGURE 9. Experimental results of ResLap, NMC.

V. RESULTS AND EVALUATIONS

We perform two experiments, one to avoid the presence of checkerboard artifacts in the generated climate images, and the other to achieve high precision precipitation prediction. For the first experiment, we can directly observe the final output images to evaluate the effectiveness of the proposed method. For the second experiment, we adopt five corresponding metrics to evaluate the accuracy of precipitation prediction.

A. THE RESULTS OF THE CHECKERBOARD ARTIFACTS

In the experiment, we apply the proposed method to the LapSRN and ResLap models, and these two networks adopt the deconvolution layer to implement the upsampling operation. Fig. 8 shows different high-resolution climate images generated by LapSRN in the CCN and SCN dataset. Fig. 9 illustrates three high-resolution climate images produced by our ResLap network in the CCN and SCN dataset. In addition, we only adopt the NMC downscaling mode.

As we can see, there are a series of apparent checkerboard patterns appearing in all final output images, when the *deker* is not equal to the *stride*. However, when the *deker* is equal to the *stride*, we notice that there are no checkerboard artifacts emerging in the generated climate images. Besides, the generated images are clear and smooth. Note that the problem of checkerboard artifacts may occur when our super-resolution models use deconvolution operations. This problem does not emerge when the model uses other upsampling operations, such as the sub-pixel convolution layer and bilinear interpolation. Finally, we can deduce that our proposed method be applied to the super-resolution models and eliminate the artifacts that appear in the final generated images.

TABLE 3. Quantitative climate evaluation of six downscaling algorithms, for scale factors 2km, 5km, and 10km. Black bold text indicates the best performance, and our ResLap model has significant advantages over other models in TS indicators. Besides, DS refers to downscaling scale.

Area	DS	Algorithm	RMSE	PC	PO	FAR	TS			
	2km	Bilinear	0.091	0.970	0.016	0.023	0.945	0.903	0.836	0.861
	2km	DeepSD	0.039	0.984	0.010	0.014	0.968	0.964	0.936	0.932
	2km	ESPCN	0.131	0.976	0.015	0.039	0.929	0.917	0.833	0.816
NCN	*2km*	VDSR	0.034	0.986	0.007	0.009	0.979	0.972	0.951	0.950
	2km	RDN	0.035	0.988	0.006	0.011	0.980	0.974	0.952	0.945
	2km	LapSRN	0.034	0.987	**0.004**	**0.007**	**0.984**	0.975	0.951	0.947
	2km	**ResLap (ours)**	**0.030**	**0.997**	0.005	0.007	0.983	**0.976**	**0.956**	**0.959**
	5km	Bilinear	0.304	0.945	0.053	0.081	0.822	0.695	0.559	0.660
	5km	DeepSD	0.102	0.975	0.020	0.038	0.923	0.903	0.825	0.866
	5km	ESPCN	0.201	0.939	0.015	0.159	0.792	0.842	0.738	0.772
NCN	*5km*	VDSR	0.105	0.978	0.018	0.029	0.936	0.906	0.828	0.857
	5km	RDN	0.092	0.985	0.019	0.020	0.950	0.919	0.847	0.879
	5km	LapSRN	0.095	0.980	0.017	0.021	0.948	0.917	0.842	0.873
	5km	**ResLap (ours)**	**0.086**	**0.991**	**0.017**	**0.017**	**0.955**	**0.933**	**0.856**	**0.899**
	10km	Bilinear	0.546	0.924	0.090	0.155	0.698	0.502	0.356	0.475
	10km	DeepSD	0.262	0.953	0.052	0.097	0.821	0.757	0.650	0.736
	10km	ESPCN	0.331	0.939	0.087	0.108	0.766	0.705	0.583	0.657
NCN	*10km*	VDSR	0.252	0.959	0.045	0.076	0.843	0.763	0.663	0.740
	10km	RDN	0.250	0.965	0.050	**0.057**	0.085	0.764	0.665	0.739
	10km	LapSRN	0.252	0.962	0.051	0.058	0.854	0.765	0.661	0.737
	10km	**ResLap (ours)**	**0.244**	**0.975**	**0.042**	0.061	**0.858**	**0.773**	**0.672**	**0.747**
	2km	Bilinear	0.128	0.953	0.010	0.016	0.963	0.922	0.866	0.876
	2km	DeepSD	0.071	0.984	0.007	0.014	0.972	0.963	0.940	0.945
	2km	ESPCN	0.268	0.913	0.003	0.249	0.711	0.909	0.869	0.883
SCN	*2km*	VDSR	0.056	0.987	0.004	0.007	0.985	0.975	0.958	0.961
	2km	RDN	0.053	0.989	0.006	0.005	0.988	0.977	0.965	0.970
	2km	LapSRN	0.058	0.987	0.004	0.006	0.987	0.978	0.962	0.966
	2km	**ResLap (ours)**	**0.042**	**0.998**	**0.003**	**0.003**	**0.991**	**0.986**	**0.974**	**0.978**
	5km	Bilinear	0.409	0.956	0.034	0.052	0.882	0.758	0.615	0.637
	5km	DeepSD	0.171	0.973	0.017	0.042	0.923	0.904	0.857	0.858
	5km	ESPCN	0.428	0.729	0.007	0.492	0.444	0.734	0.696	0.739
SCN	*5km*	VDSR	0.179	0.978	0.026	0.052	0.903	0.896	0.850	0.862
	5km	RDN	0.151	0.983	0.153	**0.018**	0.953	0.920	0.875	0.880
	5km	LapSRN	0.160	0.980	0.016	0.019	0.952	0.917	0.869	0.873
	5km	**ResLap (ours)**	**0.137**	**0.992**	**0.012**	0.019	**0.959**	**0.927**	**0.888**	**0.896**
	10km	Bilinear	0.715	0.935	0.059	0.101	0.795	0.608	0.430	0.427
	10km	DeepSD	0.385	0.959	0.038	0.072	0.861	0.799	0.702	0.701
	10km	ESPCN	0.561	0.433	0.012	0.656	0.291	0.694	0.601	0.607
SCN	*10km*	VDSR	0.377	0.956	0.048	0.069	0.854	0.798	0.701	0.706
	10km	RDN	0.374	0.965	0.041	0.042	0.891	0.808	0.705	0.704
	10km	LapSRN	0.372	0.968	0.040	**0.037**	**0.897**	0.807	0.708	0.708
	10km	**ResLap (ours)**	**0.360**	**0.978**	**0.031**	0.043	0.896	**0.815**	**0.719**	**0.720**

B. THE PERFORMANCE OF THE GENERATED CLIMATE IMAGES

Table 3 shows quantitative comparisons between our ResLap network and the other five super-resolution models, and we adopt the EC (*2km*), NMC (*5km*), and GRAPE_MESO (*10km*) mode, respectively. The values of precipitation data are generally small, therefore most models have similar values on various indicators. It is obvious that our ResLap network performs better compared with existing methods on the CCN and SCN datasets. Especially, the ResLap method

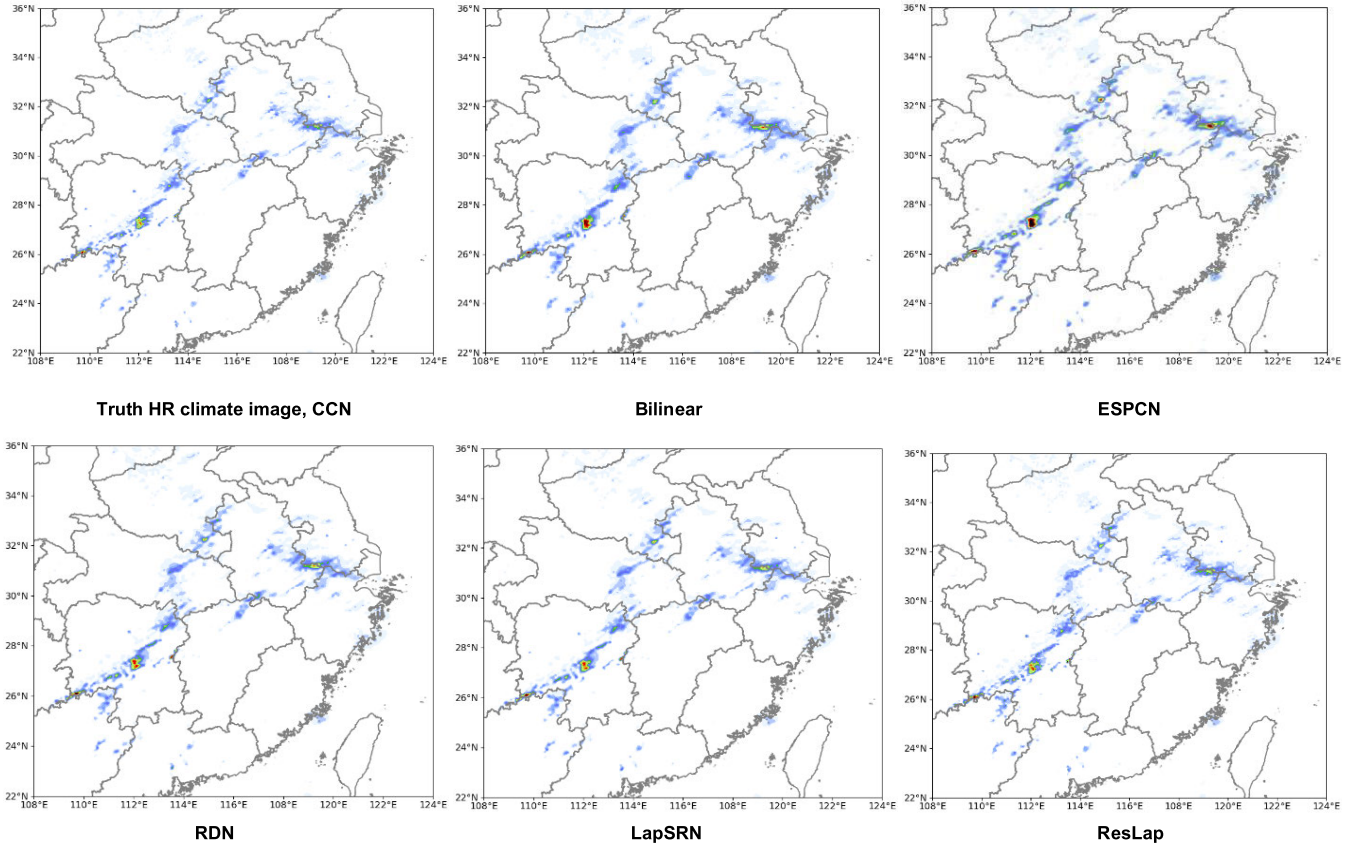


FIGURE 10. Visual comparison of five models in precipitation prediction. The downscaling scale is 10km.

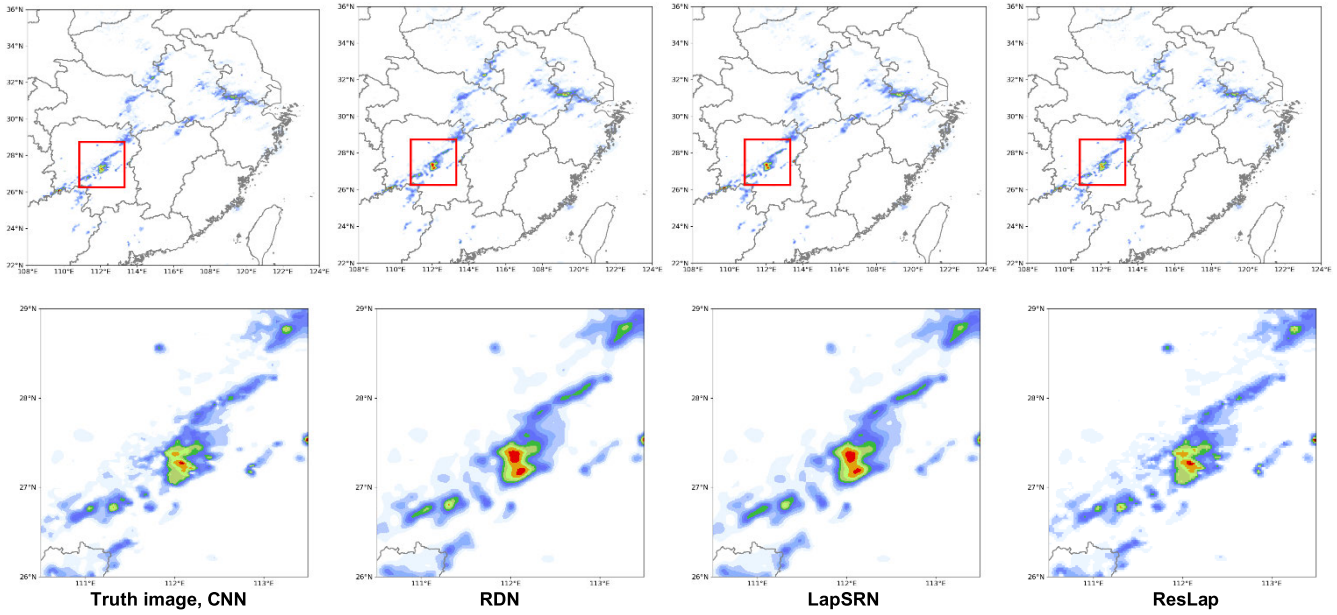


FIGURE 11. Visual comparison of precipitation prediction for RDN, LapSRN, and ResLap in darker areas. The downscaling scale is 10km. Obviously, the image generated by ResLap is closer to the real image.

can attain higher TS scores. The higher the value of the TS score denotes the more accurate the prediction of the precipitation level of the model. Moreover, we have achieved the most significant performance in the prediction of heavy rain and torrential rain. When compared with traditional super-resolution models (Bilinear), the ResLap network performs the best in every meteorological indicator, and we get the same result in all datasets with all downscaling modes.

This highlights the huge effectiveness of our algorithm in precipitation prediction. When compared with the remaining popular models (DeepSD, VDSR, ESPCN, RDN, and LapSRN), our model still has a further improvement compared with other networks. Furthermore, when the downsampling scale is small, the ResLap will not have a significant advantage over other models. For example, for the downsampling scale *2km*, most methods get almost the same results on all datasets. When the downsampling scale becomes larger, ResLap will show significant advantages over other models.

In Fig. 10, we present visual comparisons on the NMC dataset with the downscaling scale of *10km*, and we compare the ResLap network with Bilinear, ESPCN, VSDR, RDN, and LapSRN. For the selected low-resolution image, the darker the color is, the greater the precipitation in the area is, and we find that most models easily produce excessive deep color patterns in the final climate images. Among them, ESPCN generates the most image noise compared to other models. However, our model generates almost the same climate image as RDN and LapSRN. In order to compare the performance of these three models more clearly, we focus on the darker regions in the generated climate image, as shown in Fig 11. It is obvious that the images generated by our model are closer to the real images, our method can effectively suppress noise in dark areas in the generated images. It means that our approach achieves the best performance in predicting heavy rain and torrential rain.

VI. CONCLUSION

In this paper, our proposed method, termed ResLap, innovatively integrates the RDB with LapSRN. Among them, we use RDB to exploit hierarchical features from all convolutional layers, and adopt LapSRN to achieve high-resolution image reconstruction. Besides, our multi-scale root mean square error is simplified on the basis of the Charbonnier loss function, which can effectively reduce the error between the predicted images and the original truth images. The comparison experiments with some common super-resolution models also illustrate that the proposed ResLap network can produce high-quality climate images, and perform well in climate prediction. In addition, we infer that the deconvolution operation caused the checkerboard artifacts to appear in the final output images. Then, we propose a method to adjust the size relationship between *deker* and the *stride*. The extensive experimental results also show that these super-resolution models can generate climate images without any checkerboard artifacts in this method.

REFERENCES

[1] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[2] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 624–632.

[3] J. Allebach and P. W. Wong, "Edge-directed interpolation," in *Proc. 3rd IEEE Int. Conf. Image Process.*, Sep. 1996, pp. 707–710.

[4] M. S. Sajjadi, B. Scholkopf, and M. Hirsch, "EnhanceNet: Single image super-resolution through automated texture synthesis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4491–4500.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[6] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[7] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.

[8] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[9] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Sep. 2014, pp. 184–199.

[11] Y. LeCun, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[12] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 391–407.

[13] H. Greenspan, "Super-resolution in medical imaging," *Comput. J.*, vol. 52, no. 1, pp. 43–63, 2008.

[14] K. V. Suresh, G. M. Kumar, and A. N. Rajagopalan, "Superresolution of license plates in real traffic videos," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 321–331, Jun. 2007.

[15] J. C.-F. Lo, Z.-L. Yang, and R. A. Pielke, "Assessment of three dynamical climate downscaling methods using the weather research and forecasting (WRF) model," *J. Geophys. Res.*, vol. 113, p. 9, May 2008.

[16] A. W. Wood, L. R. Leung, V. Sridhar, and D. P. Lettenmaier, "Hydrologic implications of dynamical and statistical approaches to downscaling climate model outputs," *Climatic Change*, vol. 62, nos. 1–3, pp. 189–216, Jan. 2004.

[17] J. Chen, F. P. Brissette, and R. Leconte, "Uncertainty of downscaling method in quantifying the impact of climate change on hydrology," *J. Hydrol.*, vol. 401, nos. 3–4, pp. 190–202, May 2011.

[18] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, and A. R. Ganguly, "DeepSD: Generating high resolution climate change projections through single image super-resolution," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1663–1672.

[19] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, and A. R. Ganguly, "Generating high resolution climate change projections through single image super-resolution: An abridged version," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 5389–5393.

[20] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," 2017, *arXiv:1707.02937*. [Online]. Available: http://arxiv.org/abs/1707.02937

[21] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, Oct. 2016.

[22] Y. Sugawara, S. Shiota, and H. Kiya, "Super-resolution using convolutional neural networks without any checkerboard artifacts," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 63–70.

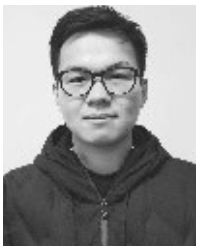
[23] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2018–2025.

[24] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, and Z. Wang, "Is the deconvolution layer the same as a convolutional layer?" 2016, *arXiv:1609.07009*. [Online]. Available: http://arxiv.org/abs/1609.07009

[25] H. Gao, H. Yuan, Z. Wang, and S. Ji, "Pixel deconvolutional networks," 2017, *arXiv:1705.06820*. [Online]. Available: http://arxiv.org/abs/1705.06820

[26] M. Hessami, P. Gachon, T. B. M. J. Ouarda, and A. St-Hilaire, "Automated regression-based statistical downscaling tool," *Environ. Model. Softw.*, vol. 23, no. 6, pp. 813–834, Jun. 2008.

[27] S. Ghosh, "SVM-PGSL coupled approach for statistical downscaling to predict rainfall from GCM output," *J. Geophys. Res.*, vol. 115, no. D22, May 2011.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[29] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4799–4807.
[30] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, *arXiv:1603.07285*. [Online]. Available: http://arxiv.org/abs/1603.07285
[31] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, pp. 79–82, Dec. 2005.
[32] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "DeepSat: A learning framework for satellite imagery," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, 2015, pp. 1–10.
[33] M. Abadi, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[35] F. Mesinger, "Bias adjusted precipitation threat scores," *Adv. Geosci.*, vol. 16, pp. 137–142, Apr. 2010.

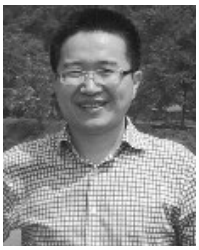


JIANXIN CHENG received the B.S. degree from the School of Computer, Hubei University, Wuhan, in 2018, where he is currently pursuing the master's degree with the School of Computer. He is also involved in computer vision and images super resolution. His research interests include natural language processing and computer vision.

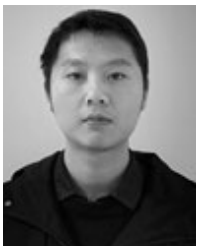
QIUMING KUANG received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2017. He joined the Public Weather Services Center, China Meteorological Administration, in 2017. He is a Team Leader in AI and weather integration with the Public Weather Services Center, CMA. His research interests include meteorological hazards forecasting and smart weather service.

CHENKAI SHEN received the B.S. degree from the School of Computer, Wuhan University, in 2018, where he is currently pursuing the M.S. degree with the Software Engineering Laboratory, School of Computer. He is involved in computer vision. His research interests include semantic segmentation and image enhancement.

JIN LIU received the Ph.D. degree in computer science from the State Key Laboratory of Software Engineering, Wuhan University, China, in 2005. He is currently a Professor with the School of Computer Science, Wuhan University. He has authored or coauthored a number of research articles in international journals. His research interests include mining software repositories and intelligent information processing.



XICHENG TAN received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2007. He was a Visiting Scholar with the Center for Spatial Information Science and Systems, George Mason University, Fairfax, VA, USA, from 2013 to 2014. He is currently an Associate Professor with the School of Remote Sensing and Information Engineering, Wuhan University. His current research interests include intelligent geocomputing, geospatial web service, high-performance geocomputing, cloud computing, sensor networks, artificial intelligence, virtual geographic environment (VGE), and natural disaster emergency response.



WANG LIU received the B.S. degree from the School of Computer, Chongqing University, in 2018. He is currently pursuing the M.S. degree with the Software Engineering Laboratory, School of Computer, Wuhan University. He is involved in computer vision and image super resolution. His research interests include image enhancement and machine translation.