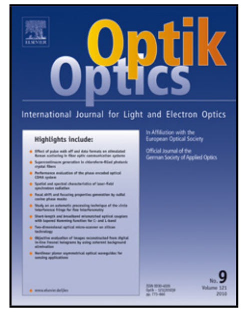
Journal Pre-proof




# Infrared Scene-based Non-Uniformity Correction Based on Deep Learning Model

Yitong Li, Ning Liu, Ji Xu

PII: S0030-4026(20)31716-2

DOI: https://doi.org/10.1016/j.ijleo.2020.165899

Reference: IJLEO 165899

To appear in: *Optik*

Received Date: 27 March 2019

Accepted Date: 26 October 2020

# Infrared Scene-based Non-Uniformity Correction Based on Deep Learning Model

Yitong Li, Ning Liu*, Ji Xu

*Nanjing University of Posts and Telecommunications, College of Electronic ans Optical Engineering & College of Microelectronics, 9 Wenyuan Ave. Nanjing, Jiangsu Province, 210023.*

**Corresponding author: coolboy006@sohu.com*

**Abstract:** In this paper, an infrared scene-based non-uniformity correction method based on deep learning technology has been proposed. This method combines the scene-based infrared non-uniformity correction with the state-of-the-art deep learning technology. The traditional scene-based non-uniformity correction technologies generally face the problem that when the radiation of the scene is changed, the correction parameters may not converge anymore, and the correction process will rise again. Multiple times of correction will potentially increase the risk of the correction failure. The deep learning can help setting up a systematic correction parameter which self-adaptive to the thermal imager, which means that once the parameters are computed, the non-uniformity will be corrected according to the deep learning network by itself. Although the pre-calculation period of setting up the parameters take much time, the upcoming correction process is much easier than the traditional technologies. We use the feed-forward denoising convolutional network as the fundamental structure, and deploy the modified residual learning process as well as the batch normalization process onto it. Figures and charts show the priority of our method.
**Keywords:** infrared non-uniformity correction; scene-based correction; deep learning; residual learning

## 1 Introduction

The fixed pattern noise(FPN) caused by non-uniformity has been a major degradation of the image quality of an infrared thermal imager[1]. In most of the cases, the traditional 2-point non-uniformity correction method can only be used as a factory setup of the parameter of a thermal imager [2,3]. When the thermal imager is used in the industrial or military facilities, the newly generated non-uniformity has to be corrected only by means of the scene-based non-uniformity correction(SBNUC) methods in order to remain the continuous of the video sequence. Thus, the research of the scene-based non-uniformity correction methods has been focused by the researchers all over the world.

The traditional SBNUC methods can be classified as the following categories: 1. Constant statistics based NUC(CS-NUC) algorithm [4]. This algorithm assumes that the first and the second order statistics of each detector's output should be the same over a sufficient number of frames. 2.Temporal high-pass filtering based NUC (THP-NUC) algorithm [5]. This algorithm sets a high-pass filtering in the temporal domain and the FPN will be removed due to its low-frequency characteristic. 3. Least mean square error based NUC (LMS-NUC) algorithm [6]. This algorithm uses a least mean square algorithm to adaptively determine the non-uniformity model parameter to achieve well correction performance. 4. The registration-based NUC (RG-NUC) algorithms [7]. This algorithm corrects the FPN under the assumption that different detectors between two adjacent frames should have identical response when observing the same scene.

However, the SBNUC methods have their own setbacks, such as the so called “ghost effect” and the choose of the parameters. The “ghost effect” occurs in almost every SBNUC methods. During the correction process, the scene information are sometimes being corrected into the image, which appears as some ghost image covered on the real image. Meanwhile, the choose of the parameters is another important aspect of the SBNUC methods. Take the RG-NUC as example, the set of the convergent speed is a main concern of the RG-NUC. If the speed is set too fast, the ghost effect will occur, if the speed is set too slow, the non-uniformity will hardly be corrected. Up to now, the choose of the parameters can only be set as a fixed value before the correction process, the goal of self-adaptive of the parameters is not achieved yet, which somehow restrict the development of the SBNUC methods. Meanwhile, the SBNUC face an unavoidable problem that, the non-uniformity will rise again when the infrared imager is facing different scenario with big change of the temperature. Under this situation, the SBNUC will start correcting again, however, multiple times of correction increases the odds to destroy the image quality.

In this paper, we propose an innovative method to deal with the non-uniformity of the infrared thermal imager. We use the method of deep learning to corrected the non-uniformity. The deep learning has been a worldwide hot point in recent years, it is a particular learning structure and computational model which contains multiple hidden layers. It can discover the distributed characteristics of the original data by combining the characteristic of the lower layers to acquire the more abstract characteristic of the higher layers [8]. With the advantage of semi-supervised or non-supervised learning and the algorithm of stratified feature extraction, deep learning uses the back propagation algorithm to let the machine know how to modify the internal parameter and to get the feature. There are many classical model structure, which can be classified into two main categories ---- supervised learning and non-supervised learning. In supervised learning, the most commonly used models are: 1. recurrent neural network (RNN). It has the advantage of processing the sequence related data [9,10]. Thus, the RNN has been widely used in voice recognition and machine translation; 2. convolutional neural network (CNN). It has the advantage of processing the network data [11,12]. Thus, the CNN has been widely used in image and video recognition, target segmentation and tracking. 3. back propagation. It can be used to process large amount of data [13]. While in non-supervised learning, the most dazzling aspects are deep generative model and autoencoder, for example, the boltzmann machine [14], deep belief network [15], generative adversarial network [16] are all belong to the deep generative model.

We use the feed-forward denoising convolutional neural networks(DnCNN) as the basic model to deal with the infrared non-uniformity. The main strategy of our work is that, unlike the common use of DnCNN, we do not use the 2 point corrected images as the background image which are generally being set as the learning target and the output of the model. Instead, we analyze the non-uniformity of an infrared thermal imager, and set the FPN, background image with the FPN and the computational loss while during the network iteration as the learning target to conduct the loss learning of the non-uniformity. The loss learning can get more stable optimum solution during the model training, and faster speed of mapping. It can overcome the gradient disappear problem during the machine training. With the deployment of deep learning, we can train a whole set of self-adaptive parameter for the infrared thermal imager, and keep it work under the environment which the non-uniformity will hardly appear on the image.

The organization of this paper is that, in section II, the basic model of DnCNN will be introduced as well as its deployment in correcting the infrared non-uniformity; in section III, the experimental setup and results will be shown to illustrate the effect of our new method; in section IV, we give conclusion of this paper.

## 2 Basic model of DnCNN

In this section, we will give a brief discussion about the DnCNN we used in this paper. The DnCNN is first proposed by Kai Zhang and Wangmeng Zu et al, which is based on the modification of the very deep convolutional networks raised by Simon Yan and the VGG networks [17]. The DnCNN abandon the pooling layers and the final fully connected layers, it is widely used in reducing the Gaussian white noise. We make significant modification of the original DnCNN in purpose of correcting the infrared non-uniformity. Next, we will give our discussion about the DnCNN we used, it contains three parts which are the total architecture of DnCNN, residual learning, batch normalization(BN).

### *2.1 the architecture of DnCNN*

The model we used is composed by 17 convolutional layers, every layer can be described by three parts of the structure and the function as shown in fig.1.

In fig.1, the first layer is defined as the first part of the model, the convolutional function and the activation function are included. The convolutional function is composed by 64 convolutional kernel which dimension is $3\times3$, and the moving step is 3 pixels. The activation function is the commonly used ReLU function. The 64 convolutional kernel work as 64 filters in order to generate 64 feature maps, and the ReLU function can then use to utilized for nonlinearity, which finally achieve the goal of machine learning [18]. The second part of the model is composed by 2-16 layers, every layer has 64 filters familiar to the first layer, along with the activation functions and the batch normalization process. The purpose of the second part is either calculate the nonlinearity feature generated from the first layer, which in order to compute higher and more abstract features, or to normalize the input data into a distribution which the average is 0, and the deviation is 1. The purpose of the BN will be discussed in 2.3. The last part of the model is the finally layer, the $17^{th}$ layer. It only contains 64 filters, which can use to regenerate the output map of the processed data. The distribution of the second part can be rearranged into 8-bit gray scale for the observation by means of the map.

*2.2 Residual learning*

When deeper networks are able to start converging, a degradation problem has been exposed. With the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training errors [19].

To this situation, Kaiming He et al. [24] raised a method called “Residual learning”, which is shown in fig.2. Fig.2 shows a residual unit, with the identity mapping of y=x added onto the normal gradient transmission, it solves the network degradation problem which caused by the fact that multiple nonlinear network cannot approach identity mapping network. The dot line on the left represents the original strategy of residual learning raised by Kaiming He [24], while the line on the left represents our modified strategy of residual learning.

In this paper, we make a modification of the traditional residual learning [20,21]. We assume the entire model as a complete residual unit instead of so many small residual units shown in fig.2. Thus, we set the input of the model as F(x), the non-uniformity as residual. Here we have to explain that, the way of acquiring the non-uniformity is quite tricky. Since the non-uniformity of every infrared thermal imager is not the same. We have to get the non-uniformity separately. We use the black body to correct the imager with the 2-point calibration in a certain temperature region. Then we adjust the radiation ratio of the black body in order to set the temperature unfit of the 2-point calibration parameters. Thus, the shape of the non-uniformity can be acquired.

*2.3 Batch normalization*

When the network depth is large enough, not only the network degradation problem will occur, but also the problem of gradient disappear or overfitting. Here we introduce a method of batch normalization to overcome these problems [22]. The principle of the BN we used can be described as the following equation:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}} \tag{1}$$

Where, $x^{(k)}$ represents the dimension of the data, $E[x^{k}]$ represents the average of a certain dimension, $\sqrt{Var[x^{k}]}$ represents the standard deviation. By doing BN, the data of each layer can be fixed in a certain scale. However, it will destroy the original data distribution a little bit, we then add two self-adaptive variables of $\gamma$ and $\beta$ into the calculate process. Eq.2 shows how these two parameters work. Through eq.3, the original unfixed data distribution can be turning into fixed data distribution, the new distribution accelerates the training speed of the entire network, and overcome the problem of gradient disappear and overfitting.

$$y^{(k)} = \gamma^k \hat{x}^{(k)} + \beta^k \tag{2}$$

## 3 Experiment setup and result

### *3.1 Computational platform and evaluation indexing*

Deep learning usually needs to be deployed on GPUs, in our experiment, the computational platform we used has an intel I7-6800K CPU, two NVidia GeForce 1080ti GPUs with total display memory of 22G, and 32G DDR4 memory. In our experiment, we first add non-uniformity onto the clean infrared images and then conduct the deep learning procedure, and finally get the result. By doing this, we can illustrate the detail of how this method works, and then we give results of this method which is working on the original raw infrared images.

Here we have to point out, in order to understand how this method works, we use two indexes as criterion----the loss function and the peak signal to noise ratio(PSNR). Where the loss function represents the averaged mean squared error between the ground images and corrected images, shows in eq.3 as follows:

$$\ell(\Theta) = \frac{1}{2N}\sum_{i=1}^{N}\|\mathcal{R}_{(y_i;\Theta)} - (y_i - x_i)\|_F^2 \tag{3}$$

Where N represents the number of the training samples, $\mathcal{R}_{(y_i;\Theta)}$ represents the loss after one step of training, y represents the images with added non-uniformity, x represents the clean images. The training aim is to modify the value of $\Theta$ as possible as it can in order to lower the loss function $\ell(\Theta)$. $\{(y_i, x_i)\}_{i=1}^{N}$ represents the comparing sets of infrared images with/without non-uniformity in the training set.

### *3.2 experiment setup*

In our experiment, we add non-uniformity onto 39228 infrared images, 39216 ones are set as training set, 12 ones are set as test set. There are three parameters need to be manually adjusted in the DnCNN model, which are the learning rate, the training steps and the batch size. We set the batch size as 24 times, which means that 24 infrared images are participated during every training procedure. We adopt the method of Cyclical learning rates raised by Leslie N. Smith [25] to set our learning rate, the strategy is shown below:

By using this strategy, the fluctuation of the learning rate is descended from 1E-3 to 1E-4, this guarantees that the learning rate will not have the problem of gradient exploding or vanishing, which accelerates the iteration speed as possible as it can.

Next, let us take a look at how our method works on the non-uniformity correction of the infrared images. Very much unlike the traditional scene-based non-uniformity correction methods, our method is using the DnCNN to predict the possible non-uniformity and compare it to the real non-uniformity, once the prediction is almost the same as the real one, the non-uniformity can be reduced from the infrared image. This method can only be done by the deep learning technology and give stable correction result to the infrared image. Looking deep into our modified DnCNN model, it is constructed by massive number of neurons shown in fig.4:

In fig.4, the input capital X are x1, x2 and the bias "+1", the output is described in eq.4:

$$h_{w,b}(X) = f(W^T X) = f(\sum_{i=1}^{2} W_i x_i + b) \tag{4}$$

Where, $W_i$ represents the weighting vector, it indicates that whether a neuron is activated. If the value in $W_i$ is 1, then the related neuron is activated, if the value in $W_i$ is 0, the related neuron is not activated. Only the activated neurons will participate in the current forward propagation and back propagation process. $b$ represents the bias, it is a constant. The function $f$ is the so called activated function, in our model we use the "ReLU" function as the activation function because of its good linearity feature. The red dot square box represents a filter, 64 parallel filters form a convolutional layer, which shows in blue rectangle in fig.1. When the learning process starts, the images in the training set will be input into the filters in each layer, the parameters $W_i$ and $b$ will be initialized at the beginning. Each layer will randomly compute one feature of the image, that is to say, there are 64 random features will be computed in first layer, the parameters $W_i$ and $b$ will be updated at the same time. These features are the prediction of the non-uniformity, after the first layer, the prediction will be entering the following layers for further calculation. Once all the 17 layers have been passed through, we can get two computed parameters $W_i$ and $b$. The final output of this round of iteration can be acquired with eq.4. Then the model will calculate the loss function between the output and the original input, since we use the added non-uniformity to illustrated the computation step in the experiment, here the input is the added non-uniformity and the output is the predicted non-uniformity. The loss function will calculate the differences of the real non-uniformity and the predicted non-uniformity, if the loss is strong, the model will conduct the back propagation onto the parameters w and b from the very first layer to the last layer in order to adjust them with the gradient descent method like eq.5:

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W,b)$$
$$b_{i}^{(l)} = b_{i}^{(l)} - \alpha \frac{\partial}{\partial b_{i}^{(l)}} J(W,b) \tag{5}$$

Where $\alpha$ represents the learning rate. When the parameters $W_i$ and $b$ are updating, the back propagation algorithm eq.5 is applied to do so. The terms $\alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W,b)$ and $\alpha \frac{\partial}{\partial b_{i}^{(l)}} J(W,b)$ are following the next eq.6:

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W,b) = \left[ \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial W_{ij}^{(l)}} J(W,b;x^{(i)},y^{(i)}) \right] + \lambda W_{ij}^{(l)}$$
$$\frac{\partial}{\partial b_{i}^{(l)}} J(W,b) = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial b_{i}^{(l)}} J(W,b;x^{(i)},y^{(i)}) \tag{6}$$

For a given training set like the infrared images with non-uniformity, the forward propagation will compute the output of each neuron. When all the output of the whole model has been acquired, the loss function calculates the “loss” of these neurons, the “loss” can be used to estimate the trend of the parameters of the neurons, and the minimum loss and best PSNR can then be reached.

*3.3 Experimental results*

In this part, we will give some test results to illustrate the performance of our method. First, we will analyze the influence of the integrity of our method, and then compare the performance of our method with the traditional scene-based non-uniformity correction algorithm.

As we mentioned in above, our model is constructed by several parts. The activation function “ReLU” and the BN affect most to the correction performance. We run our model in three ways which are the whole model way, the model without ReLU but with BN, and the model without BN but with ReLU to test the computational effectiveness by the trend of the loss descent. The results are shown in fig. 5:

Fig.5(b) is represents the red dot line region in fig.5(a). We can see through fig.5 that, the model performs much butter in correcting the non-uniformity when it maintains the integrity. The blue line representing the trend of the whole model has much better stable results. This means that the prediction of the non-uniformity achieves high similarity with the added non-uniformity. According to our calculation, in fig.5, the original loss of these three forms of model are almost the same, the value is about 80, and the PSNR is about 5db. When the iteration starts, the loss of the integral model goes down fast after 2500 times iteration and reaches under 5, while the PSNR goes up to 35db. Also, the loss and PSNR do not obviously fluctuate from then on. The final loss is about 0.22, and the PSNR is 37db with the integral model. In the model with BN but without ReLU, the loss goes down fastest in the early 100 times iteration, the PSNR goes up fastest as well. But after that, a serious fluctuation shows up, the loss goes up to 400, and the PSNR goes down to 20db. After 4000 times iteration, the loss and PSNR are stabled at 14 and 35db separately, which means that the learning ability drops seriously when the model casts the ReLU away. In the model with ReLU but without BN, the iteration speed of the model is much slower than the other two forms of model. This form of model cannot reach the optimum solution after 50000 times iteration, the loss is 0.6, and the PSNR is 36db. Meanwhile, the fluctuation is stronger than the other two forms of model, in this situation, the model may take the locally optimum solution as the final result and miss the true optimum solution.

The reason why the learning ability drops when the ReLU is not included in the model can be answered that under this circumstance, the model is turning to a linear regression model, which means that the weighting and bias coefficients can only change linearly. Although the model response fast to the input data, the computational ability is limited. While if the BN is not included in the model, the distribution of the deep learning network will be shifting away when the network is conducting non-linear calculation. The deeper the network reaches, the shifting will become more obvious. This will affect the back propagation process and cause the gradient exploding problem, thus the iteration speed will be very slow.

Next, we will give some figures to show the correction performance of our method, meanwhile, we will compare the performance of our method with the traditional scene-based non-uniformity correction method in these figures. The method we choose to compare is the algorithm our team published in 2014[7]. The thermal imager we use for testing has a 384×288 VOx un-cooled microbolometer. According to the computational process mentioned above, we cut the image into 256×256 dimension ahead of the time. First, let us take a look at fig.6:

It can be seen from fig.6, our method has the best correction result. The corrected infrared image appears much cleaner in fig.6(b). Fig.6(c) has almost the same corrected result, but if we look carefully into it, we can spot some slightly “ghost effect” within the figure. During the test, we manually adjust the convergence step to 0.07 according to ref.7 and get this results. The comparison shows that, the traditional correction method is riskier than the proposed method. While according to the proposed method, once the correction parameter has been determined, the upcoming infrared stream will adapt to the parameter and finish the correction automatically. Fig.6(d) and fig.6(e) are two results when the ReLU or the BN is casting away from the model, unsatisfying results are then showed up.

We use the same microbolometer to test the correction result. We also make some comparisons between our new method and the method in ref.7. The results are shown in fig.7:

Fig.7 gives four comparisons of the new method and the traditional method. In every comparison pair, the figure on the left is corrected by the new method, the one in the middle is the image with non-uniformity, and the one on the right is corrected with ref.7. In fig.7 we can see that, our new method surely has better correction performance. When the scene has much detail, the difference between these two methods is tiny, while the detail is not that sufficient, the correction difference becomes clearer. Evidence shows the superiority of our new method.

## 4 Conclusion

Deep learning plays very important role in nowadays, it gives significant advantage in almost every field of research. The modern industry, civilian and military all need the assistance of deep learning. In this paper, our goal is to find a way to combine the scene-based non-uniformity correction with the deep learning. Through the understanding and modification of the DnCNN deep learning model, we finally figure out a way to do so. With the help of deep learning, we can control the correction parameters in a much stable way than the traditional correction method. We do not have to worry about the convergence problem which happens in the traditional correction method. We will further our research in this field and raise the field applicable technology in the future.

## Acknowledgments

This work was supported in part by the National Science Foundation of China under Grant 61505083 and the Scientific Research Foundation of Nanjing University of Posts and Telecommunications (NO. NY215043).

## References

1. D. R. Pipa, E. A. B. da Silva, C. L. Pagliari, and P. S. R. Diniz, "Recursive Algorithms for Bias and Gain Nonuniformity Correction in Infrared Videos," IEEE T. Image Process 21, 4758-4769 (2012).
2. A. Friedenberg and I. Goldblatt, "Nonuniformity two-point linear correction errors in infrared focal plane arrays," Opt. Eng. 37, 1251-1253 (1998).
3. Y. P. Cao and C. L. Tisse, "Shutterless solution for simultaneous focal plane array temperature estimation and nonuniformity correction in uncooled long-wave infrared camera," Appl. Opt. 52, 6266-6271 (2013).
4. J. G. Harris and Y. M. Chiang, "Nonuniformity correction of infrared image sequences using the constant-statistics constraint," IEEE T Image Process 8, 1148-1151 (1999).
5. D. A. Scribner, K. A. Sarkady, J. T. Caulfield, M. R. Kruer, G. Katz, C. J. Gridley, and C. Herman, "Nonuniformity correction for staring IR focal plane arrays using scene-based techniques," in (International Society for Optics and Photonics, 1990), 224-233.
6. D. A. Scribner, K. A. Sarkady, M. R. Kruer, J. T. Caulfield, J. D. Hunt, and C. Herman, "Adaptive nonuniformity correction for IR focal-plane arrays using neural networks," in (International Society for Optics and Photonics, 1991), 100-109.
7. N. Liu and H. Qiu, "A time-domain projection-based registration-scene-based nonuniformity correction technology and its detailed hardware realization," Opt. Rev. 21, 17-26 (2014).
8. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. “Deep learning.” Nature

521.7553 (2015): 436-444.
9. Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. “Speech recognition with deep recurrent neural networks.” 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.
10. Graves, Alex. “Generating sequences with recurrent neural networks.” arXiv preprint arXiv:1308.0850 (2013).
11. W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, G. Zweig “Achieving Human Parity in Conversational Speech Recognition.” arXiv preprint arXiv:1610.05256 (2016).
12. Yann LeCun, leon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. IEEE, November, 1998.
13. Szegedy, Christian, Alexander Toshev, and Dumitru Erhan. “Deep neural networks for object detection.” Advances in Neural Information Processing Systems. 2013.
14. Hopfield J J. “Neural networks and physical systems with emergent collective computational abilities.” Proceedings of the National Academy of Sciences of the United States of America, 1982, 79 (8): 2554-2558.
15. Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets.” Neural computation 18.7 (2006): 1527-1554.
16. Goodfellow, Ian, et al. “Generative adversarial nets.” Advances in Neural Information Processing Systems. 2014.
17. Alexis Conneau, Holger Schwenk, Loïc Barrault, Yann Lecun, “Very Deep Convolutional Networks for Text Classification”, Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1107–1116.
18. A Krizhevsky, I Sutskever, GE Hinton, “ImageNet classification with deep convolutional neural networks”, International Conference on Neural Information Processing Systems, 2012, 60(2) :1097-1105.
19. Sergey Zagoruyko, Nikos Komodakis, ”Wide Residual Networks”, arXiv preprint arXiv:1605.07146, 2016.
20. A.Emin Orhan, Xaq Pitkow, “ SKIP CONNECTIONS ELIMINATE SINGULARITIES “, Published as a conference paper at ICLR 2018.
21. D.Kiku, Y.Monno, M.Tanaka, and M.Okutomi, “Residual interpolation for color image demosaicking,” in 2013 IEEE International Conference on Image Processing. IEEE, 2013, pp. 2304–2308.
22. Sergey Ioffe, Christian Szegedy, ” Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift ”, arXiv preprint arXiv:1502.03167, 2015.
23. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in International Conference on Machine Learning, 2015, pp. 448–456.
24. K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on image net classification,” in IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
25. Leslie N. Smith, “Cyclical Learning Rates for Training Neural Networks”,

Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on.

Journal Pre-proof

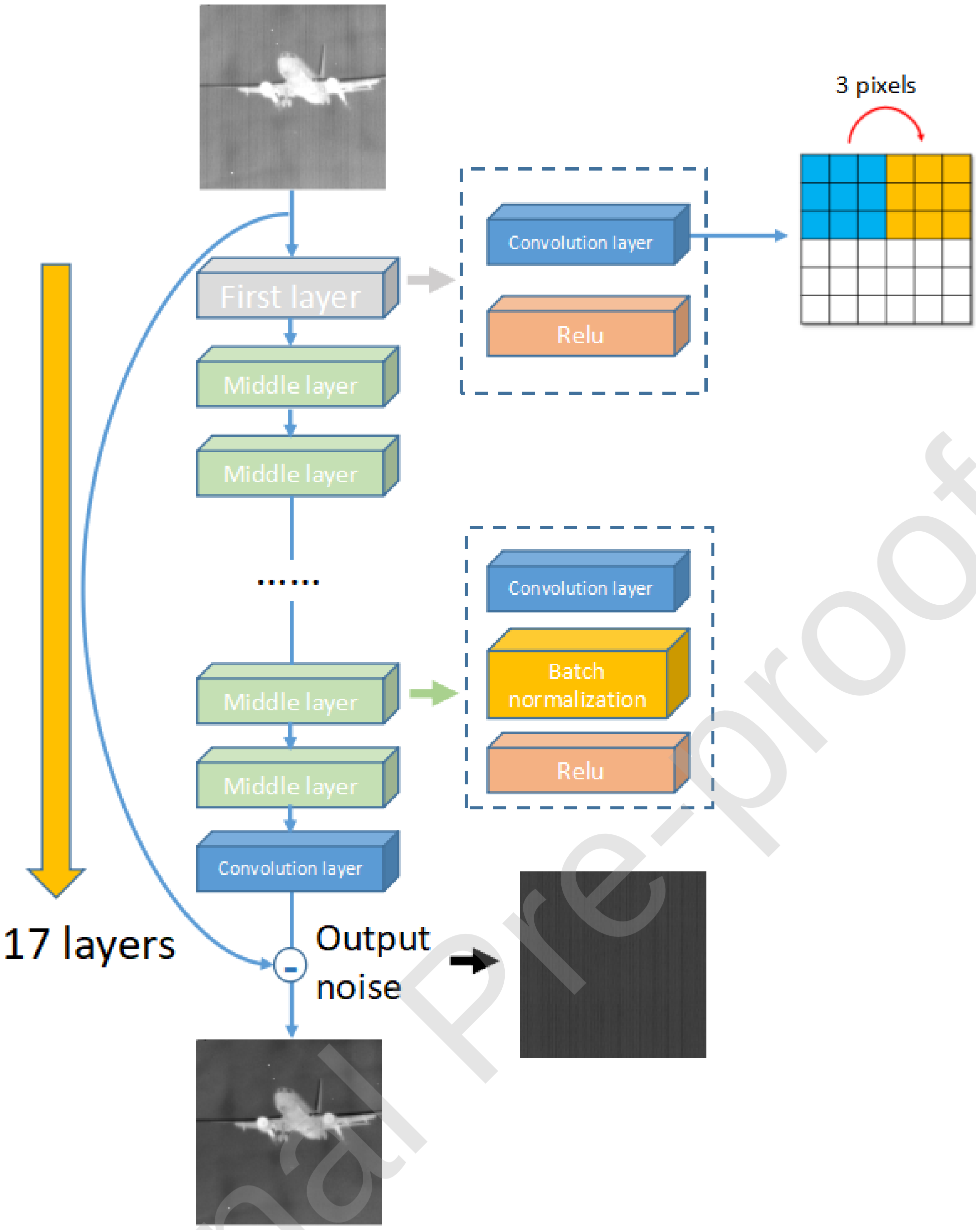



Fig.1 total structure of our deep learning strategy

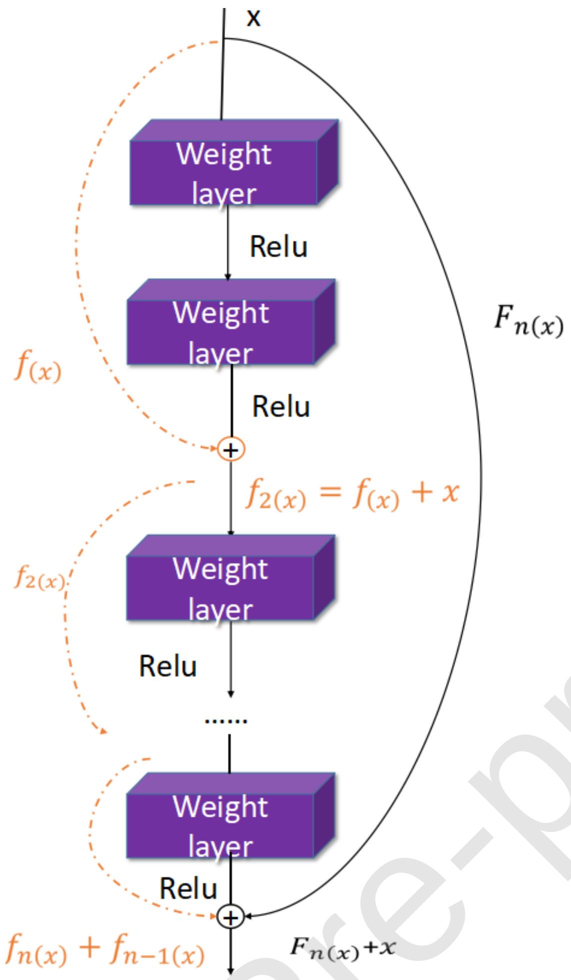



Fig.2 modified strategy of residual learning

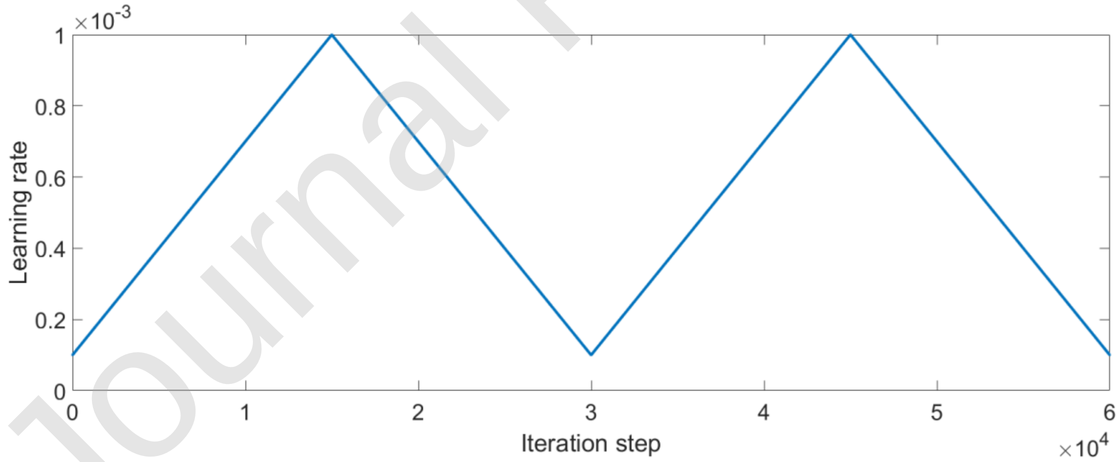



Fig.3 learning rate strategy

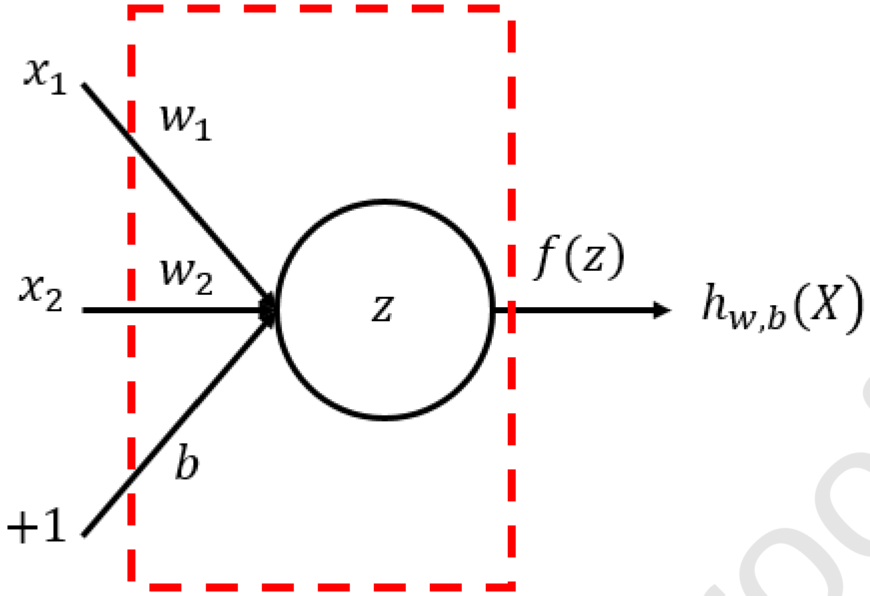
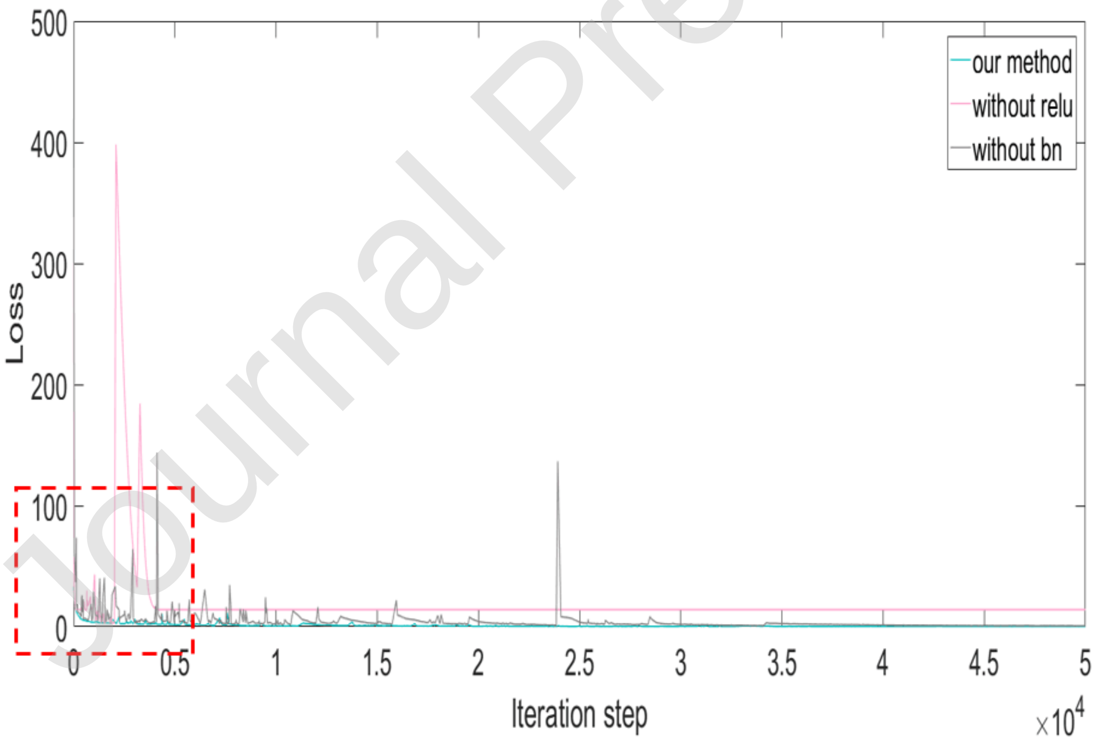
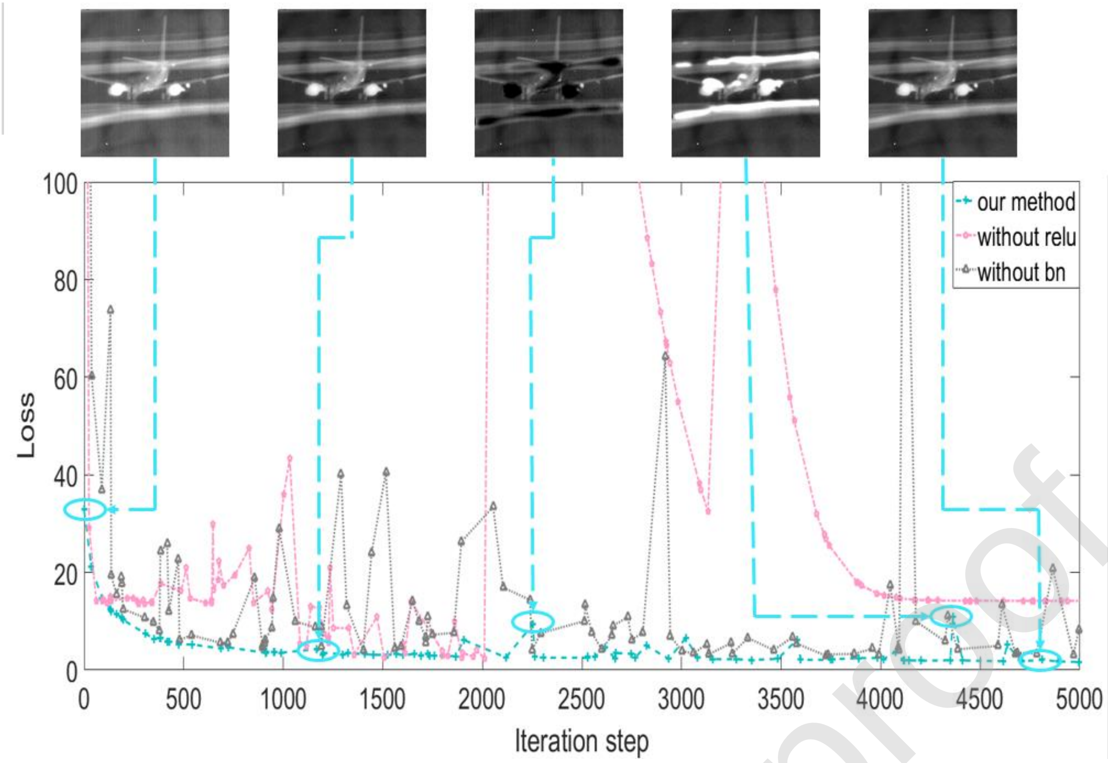



Fig.4 neurons within our modified DnCNN model




(a)




(b)

Fig.5 the computational effectiveness shows by the trend of the loss descent. (a)the total trend of the loss descent after 50000 times iteration; (b) the trend of the loss descent after 5000 times iteration

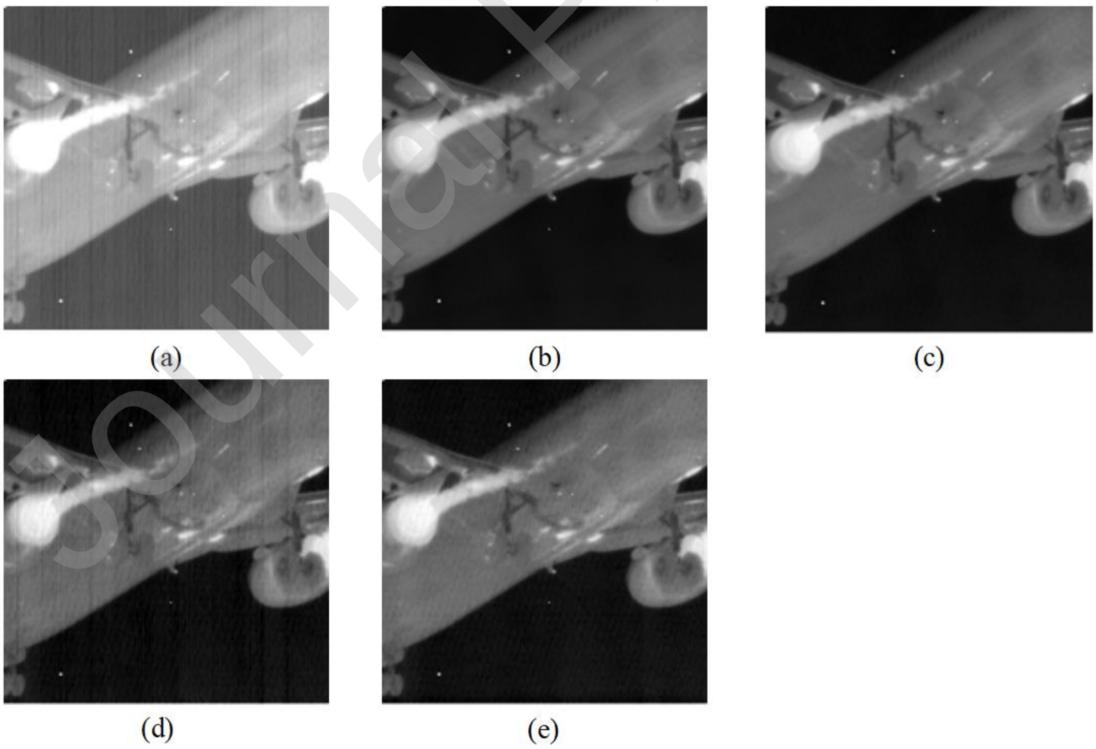
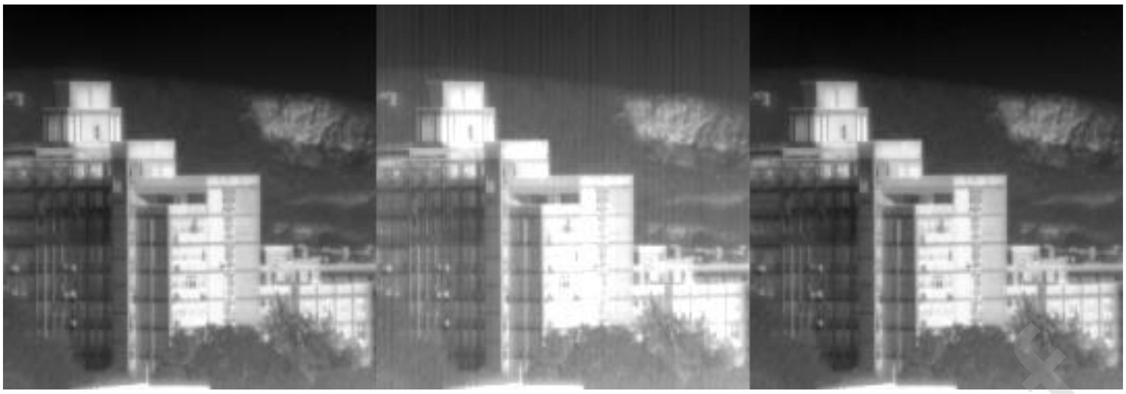
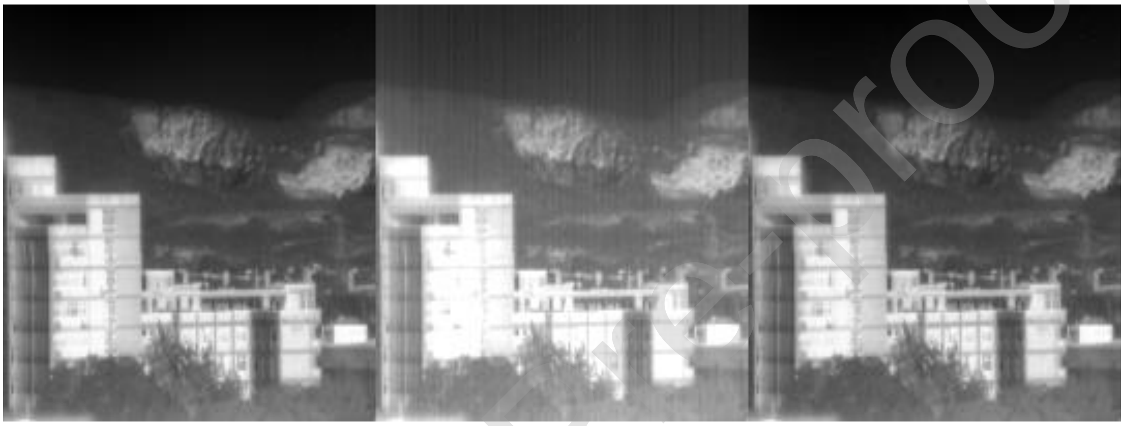



Fig.6 the correction results comparison. (a)the infrared image with non-uniformity;

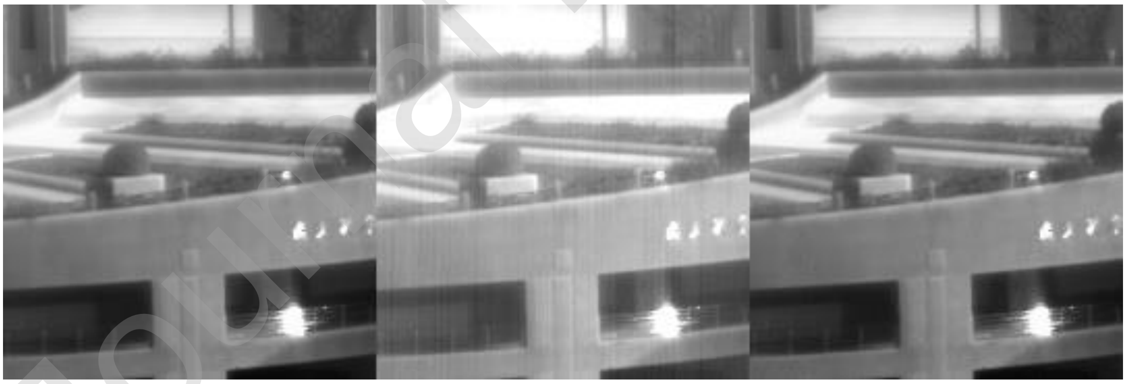
(b)the correction result of our method; (c)the correction result in ref.7; (d)the correction result without BN; (e)the correction result without ReLU



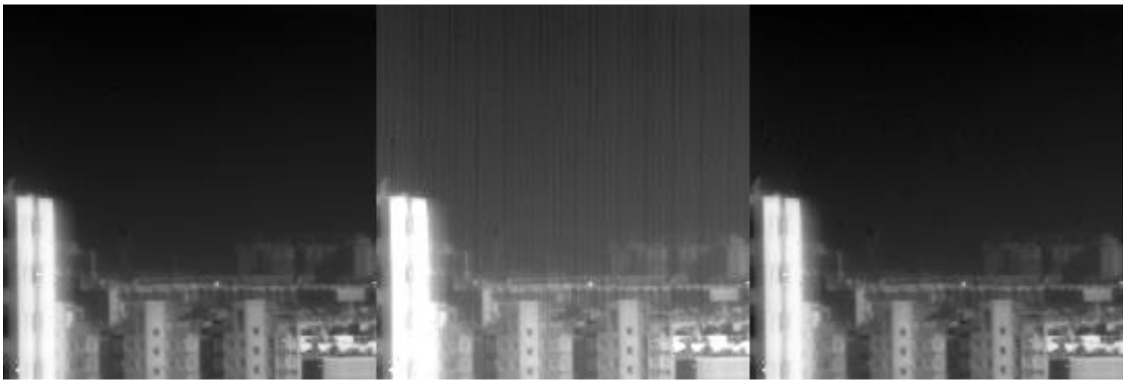
(a)



(b)



(c)



(d)

Fig.7 more comparisons between our new method and the traditional scene-based non-uniformity correction algorithm.