Lightweight models for weather identification

Congcong Wang[1,2,3,4], Pengyu Liu[1,2,3,4,*],Kebin Jia[1,2,3,4,*] ,Siwei Chen[1,2,3,4]
[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124 China
[2]Beijing Laboratory of Advanced Information Networks, Beijing 100124 China
[3]Beijing Advanced Innovation Center for Future Internet Technology, Beijing University of Technology, Beijing 100124 China
[4]Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing University of Technology, Beijing 100124 China
[*]E-mail: liupengyu@bjut.edu.cn, kebinj@bjut.edu.cn

Abstract—**At present, the recognition of weather phenomena mainly depends on the weather sensors and the weather radar. However, large-scale deployment of meteorological observation equipment for intensive weather monitoring is difficult because it is expensive and difficult to maintain. Moreover, convolutional neural networks (CNNs) can also be used to identify weather phenomena, but existing methods require high computing power of equipment, making it difficult to deploy in practice. Therefore, designing a lightweight model that can be deployed in a small device with weak computing power is crucial for intensive weather monitoring. In this paper, we study the shortcomings of some existing lightweight models. By comparing the disadvantages of these models, a new lightweight model is proposed. In addition, considering the number of existing weather datasets are too small to meet real monitoring needs, so we produced a dataset with a more complex variety of weather phenomena. Through the experiments, the proposed method can save more than 25 times memory usage with only 1.55% accuracy lost compared with the best CNNs method which achieves state-of-the-art performance among the other lightweight models.**

I. INTRODUCTION

At present, in the field of meteorology, the identification of weather phenomena mainly relies on hardware device-based methods such as the weather sensors [1,2] and the weather radar [3,4]. However, due to the cost, these devices are difficult to deploy in large areas and are difficult to maintain.

Convolutional neural networks (CNNs) have become ubiquitous for superhuman accuracy in challenging image recognition tasks [6]. However, there are two problems in applying convolutional neural network to weather identification: 1. Training a CNNs model requires a lot of data which is not easy to obtain. 2. Modern state-of-the-art networks tend to be deeper and more complicated to achieve higher accuracy which require high computational resources beyond the capabilities of small devices.

This paper reviews some popular structures and proposes a series of lightweight models for weather identification that can be deployed in small devices to achieve intensive weather monitoring. Section 2 reviews prior research on identification of weather phenomena. Section 3 describes the details of our method. Section 4 presents extensive experiment to demonstrate the model's performance. Section 5 contains conclusions and future work.

II. RELATED WORK

Due to the complexity and diversity of images, the identification of weather phenomena has always been a difficult problem in computer vision [5]. Some studies have tried such as atmospheric scattering model and region concurrent selection model, while others have extracted HOG, contrast and other features after image segmentation and combined them with machine learning to classify images [7,8]. However, these methods generally have high requirements for image preprocessing and are difficult to achieve end-to-end prediction.

Ref. [9] used convolutional network in weather classification earlier. This paper proposed a simple eight-layer convolutional network to classify two weather conditions, namely cloudy day and sunny day, which can achieve high accuracy without preprocessing of images. Then using CNNs for weather identification became more and more popular [10,11,12,13].

While as the performance of neural network often improves with the increase of layers, which also means higher requirements on data volume. In order to train with small-scale data in a large network, transfer learning was introduced. This lead to improvements in accuracy, but such a large network is difficult to deploy. In addition, transfer learning limits the freedom of model design which means it's hard to combine modern state-of-the-art structures.

Since CNNs' high demand to computation power. Designing lightweight network has been an active research area in recent years. SqueezeNet [14] extensively uses 1x1 convolutions with squeeze and expand modules primarily focusing on reducing the number of parameters. MobileNetV1 [15], MobileNetV2 [16], MobileNetV3 [29], ShuffleNet [17] use depthwise separable convolutions [18] to decrease the computation cost. Also, pruning [19,20,21,22], quantization [23,24] and knowledge distillation [25,26] are important complementary effort to improve the network efficiency.

Consider the need to deploy on hardware, we choose to design a lightweight network. Some current lightweight models sacrifice a lot of precision for efficiency [14,15,17] while others are designed for specific tasks and require a lot of computing equipment and time to train [29,32]. The proposed model focuses on both precision and efficiency. It is inspired by the above models and combines some of the most advanced structures to achieve great improvements in accuracy and efficiency.

III. METHOD DETAILS

The proposed model is based on a lightweight building blocks. We will the model structure and details in this section.

A. Building blocks

Depthwise Separable Convolutions are key building blocks for many efficient neural architectures [15,16,17] and we use them in the present work as well.

The difference between standard convolution and depthwise convolution can be seen in Fig.1.

In standard convolution a feature map will convolution with each filter while in depthwise convolution each feature map only convolution with one filter.

Network performance can be greatly improved by multiplexing image features [26,27]. To take advantage of this, we design our building blocks inspired by the inverted residuals structure in MobileNetV2 [16], which is shown in Fig.2.

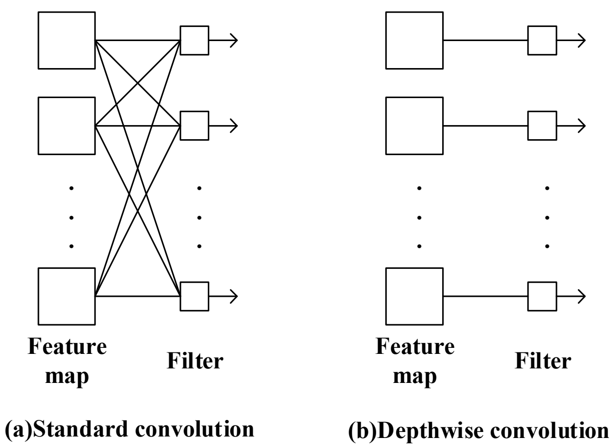

Fig.1 Standard convolution and Depthwise convolution

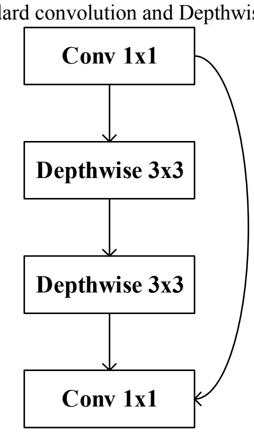

Fig.2 Building blocks

B. Nonlinearities

A better nonlinearity called swish was introduced [28] that when used as a drop-in replacement for ReLU. The nonlinearity is defined as:

$$swish\ x = x \cdot \sigma(x) \quad (1)$$

While sigmoid function spend too much computation cost, MobileNetV3 [29] replace sigmoid function with its piece-wise linear hard analog, the hard version of swish becomes:

$$hswish = x(ReLU(x+3))/6 \quad (2)$$

Use (2) to replace the original swish function can save much computation cost.

C. Squeeze and Excitation

At present, convolution is carried out in 2D space. In essence, it only models the spatial information of images and does not model the information between channels. Ref. [30] proposed a Sequeeze and Excitation block to explicitly model the information between channels. The structure is shown in Fig.3.

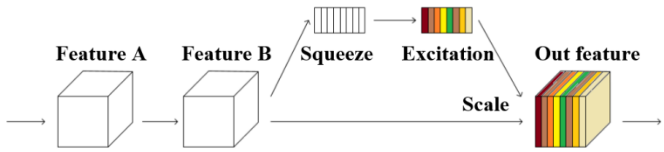

Fig.3 Squeeze and Exciation

The feature map A through the network layers become feature B, feature B then through a Sequeeze and Excitation block. The Sequeeze structure doing global average pooling and get 1x1xC feature map which has a global receptive field. Then the excitation structure uses a fully connection network to take a nonlinear transformation on the new feature map. Finally, the scale block uses the Excitation results as weight, take to the input features.

D. Model define

With these structure, we redefine our building blocks in Fig.4. Each block has two 1x1 filters as pointwise convolution, two 3x3 filters to capture the feature maps as well as batchnorm and hswish after each convolution layer. The Squeeze and Excition block is at the bottom of the block. Also, if stride equals to one, there is a skip-connection structure.

Our blocks use two fixed 3 by 3 filters instead of a 5 by 5 filter. This is because two three-by-three filters obtain the same receptive field as five-by-five but with fewer parameters which can be calculated in (3):

$$RF_{l+1} = RF_l + (ker\_size_{l+1} - 1) * feature\_stride_l \quad (3)$$

RF stands for reception field, the subscript l is the number of layers, ker_size stands the size of kernel, feature_stride means the kernel stride.

In addition, more convolution layers will bring better nonlinear properties to the model

We define our model by stacking these blocks, also with a feature capture block in the top and a classification block in the bottom. Our model is shown in table I, we compare the performance with other model in experiment section.

Input denotes the input feature map's size. Operator denotes the function unit. Exp size denotes the expand size in building blocks. #out denotes the number of output channels. Stride denotes the step length of convolution.

E. Scaling

Scaling up the model is widely used to achieve better accuracy [26,31] while shrinking the model save more computing resources. We define a width multiplier α to scale the model. We show it in experiment section.

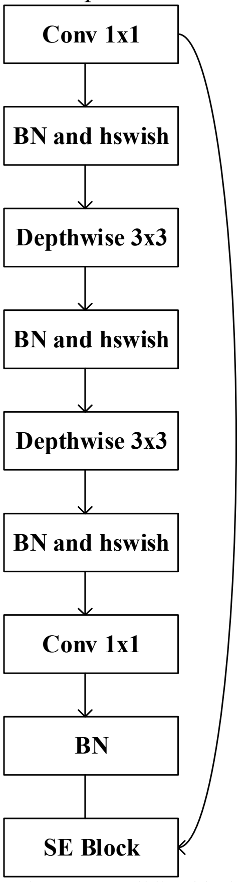

Fig.4 New Building blocks

Table I. Specification for the proposed model

Input	Operator	exp size	#out	stride
$224^2 * 3$	conv2d	-	8	2
$112^2 * 8$	block	16	12	2
$56^2 * 12$	block	24	18	1
$56^2 * 18$	block	36	24	2
$28^2 * 24$	block	48	32	1
$28^2 * 32$	block	64	48	2
$14^2 * 48$	block	96	96	1
$14^2 * 96$	conv2d	-	364	1
$14^2 * 364$	pool	-	-	1
$1^2 * 364$	fc	-	6	1

IV. EXPERIMENT

we compared the performance between transfer learning and our method. In addition, some lightweight models are in the comparison range. Finally, we scaling the model to exploit the best performance. The dataset and code are at: https://github.com/guhuozhengling/lightweight-model-for-weather.

A. Datasets

We made a dataset containing six weather phenomena including dew, frost, haze, rain, sand and snow, the number of picture is 12100, a simple demonstration is shown in Fig 5. Our access to images is through the Internet, physical photography and academic exchanges [5]. In addition to our own dataset, we conducted experiments on a publicly available dataset with four types of weather phenomena as we called it dataset-four.

B. Experiment settings

We split our dataset as training set, validing set and testing set by 3:1:1 with a fixed random seeds, this ensures that our experiments are run on the same data. We use data augmentation in training set to prevent overfitting. The testing set is only using once at last to evaluate the performance of model.

Fig.5 The image demonstration

We compare our model with some transfer learning model including Vgg16, Vgg19, Resnet152, Densenet201, Inception_V3 and some lightweight model including Squeezenet, Shufflenet, Efficientnet, MobilenetV1-V3. Evaluation criteria include accuracy and memory usage.

C. Experiment results

Table II. Comparison with transfer learning

Model name	Acc1	Acc2	Memory usage(MB)
InceptionV3	82.97	90.95	733.33
Resnet152	**92.98**	**96.55**	829.00
Densenet201	92.61	95.26	510.66
Vgg16	89.45	95.26	735.52
Vgg19	89.66	93.07	775.68
Proposed model	91.43	**96.55**	**32.97**
Performance gap	-1.55	0.00	25.14 times

Table III. Comparison with other lightweight models

Model name	Acc1	Acc2	Memory usage(MB)
Squeezenet	72.97	89.66	92.62
Shufflenet	89.53	95.19	60.71
Efficientnet-0	87.74	93.53	123.20
Efficientnet-1	87.00	90.95	175.33
Efficientnet-2	86.84	92.24	187.06
Efficientnet-3	87.08	93.97	250.01
MobilenetV3-large	90.89	94.40	118.06
MobilenetV3-small	90.47	93.53	40.35
Proposed model	**91.43**	**96.55**	**32.97**

Table II shows the performance between the proposed model and transfer learning in different models. Acc1 denotes the accuracy in our dataset and Acc2 denotes the accuracy in dataset-four. Performance gap denotes the performance gap between the best models in accuracy which is resnet152 in this table. Table III shows the performance between the proposed model and other lightweight models.

Through the experiment, transfer learning with Resnet152 achieves the best accuracy in both two datasets while it also takes up the most memory usage. The proposed model can save more than 25 times memory usage with only 1.55% accuracy lost. Compared to the rest of transfer learning model except

Densenet201, the proposed model demonstrates the advantages of both precision and efficient.

Compared with other lightweight models, the proposed model also shows great performance. To our best knowledge, MobilenetV3 is the best lightweight models for Imagenet task. Our model design was greatly inspired by this while it is not suitable for training on small datasets. That can be seen by the enormous accuracy drop in the dataset-four which is smaller than our dataset. Nevertheless, the accuracy of the proposed model is better than that of the best transfer learning model.

Fig.6 and Fig.7 visually illustrates the performance comparison between the two datasets. It can be seen that the proposed model in both two figures are located in the upper left corner of the figure, which indicates that the models built in this paper can maintain high classification accuracy and meet the original design intention while takes up less memory resources, thus reducing the demand for computing resources and facilitating deployment and application.

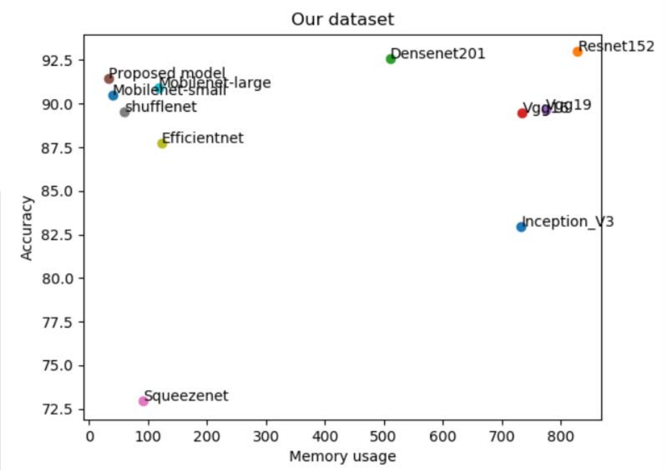

Fig.6 Comparison in our dataset

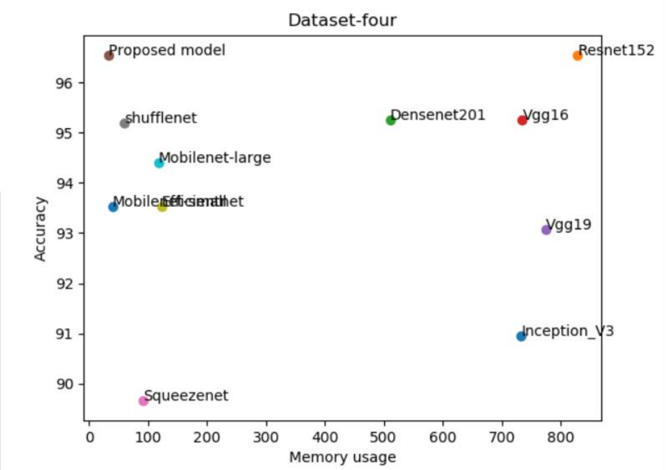

Fig.7 Comparison in dataset-four

D. Scaling

Sometimes we need to extend our model to other tasks or devices. Instead of re-designing the models, we can conduct the scaling of our models. In order to balance accuracy and efficiency, we scaling our model by a width multiplier α to change the internal channels of model.

Through our experiments, when α is equal to 2.5 which means expanding the models by 2.5 times can reach a more precise result in the same task. Fig.8 shows the loss curve. If the computing power of the device allows, the accuracy can be improved by deploying a 2.5 times larger model.

For different tasks and hardware limitations we can simply change the width multiplier α converting to a new model, which is computationally friendly.

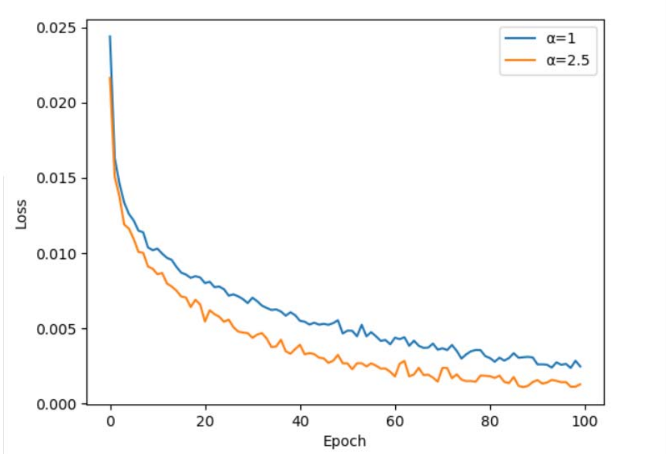

Fig.8 Loss curve

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a lightweight model for weather classification and built a weather dataset. We have reviewed efficient structures and used them in our model. Through the experiment by our weather dataset. The proposed model can save 25 times memory usage comparing with the best transfer learning model with only 1.55% accuracy lost. Compared with other lightweight models, the proposed model achieves the optimal efficiency and accuracy. Also, our model performs equally well on other datasets. In addition, we have studied the scaling method that can easily expand the model to other tasks.

We plan to work on how to combine our model with pruning to achieve a higher performance and use neural structural search to find the most appropriate parameter α.

ACKNOWLEDGMENT

This paper is supported by the following funds: Development of major scientific instruments and equipment (2018YFF01010100), National natural science foundation of China(61672064), Beijing natural science foundation project(4172001) and Advanced information network Beijing laboratory(PXM2019_014204_500029).

REFERENCES

[1] H. Üçgün and Z. K. Kaplan, "Arduino based weather forecasting station," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 972-977.

[2] M. I. Haque, A. H. MD. Shatil, A. N. Tusar, M. Hossain and M. H. Rahman, "Renewable Powered Portable Weather Update Station," 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 2019, pp. 374-377.

[3] S. Turso, T. Bertuch, S. Brüggenwirth, A. Danklmayer and P. Knott, "Electronically steered cognitive weather radar - a technology perspective," 2017 IEEE Radar Conference (RadarConf), Seattle, WA, 2017, pp. 0563-0566.

[4] A. Rudiakova, A. Lipinskii and F. Yanovsky, "Optoelectronic Aided Polarimetric Weather Radar Signal Processing," 2018 IEEE 17th International Conference on Mathematical Methods in Electromagnetic Theory (MMET), Kiev, 2018, pp. 111-114.

[5] Z. Zhang and H. Ma, "Multi-class weather classification on single images," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 4396-4400.

[6] Z. Zhang and H. Ma, "Multi-class weather classification on single images," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 4396-4400.

[7] Y. Liu, H. Li and M. Wang, "Single Image Dehazing via Large Sky Region Segmentation and Multiscale Opening Dark Channel Model," in IEEE Access, vol. 5, pp. 8890-8903, 2017.

[8] Navneet Dalal, Bill Triggs, "Histograms of oriented gradients for human detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886-893, 2005.

[9] M. Elhoseiny, S. Huang and A. Elgammal, "Weather classification with deep convolutional neural networks," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 3349-3353.

[10] J. An, Y. Chen and H. Shin, "Weather Classification using Convolutional Neural Networks," 2018 International SoC Design Conference (ISOCC), Daegu, Korea (South), 2018, pp. 245-246.

[11] C. Lu, D. Lin, J. Jia and C. Tang, "Two-Class Weather Classification," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2510-2524, 1 Dec. 2017.

[12] Y. Shi, Y. Li, J. Liu, X. Liu and Y. L. Murphey, "Weather Recognition Based on Edge Deterioration and Convolutional Neural Networks," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, 2018, pp. 2438-2443.

[13] Z. Li, Y. Jin, Y. Li, Z. Lin and S. Wang, "Imbalanced Adversarial Learning for Weather Image Generation and Classification," 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 2018, pp. 1093-1097.

[14] Iandola, Forrest N., et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." arXiv preprint arXiv:1602.07360 (2016).

[15] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

[16] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[17] Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[18] Sifre, Laurent, and Stéphane Mallat. "Rigid-motion scattering for image classification." PhD thesis, Ph. D. thesis 1 (2014): 3.

[19] Han, Song, et al. "Learning both weights and connections for efficient neural network." Advances in neural information processing systems. 2015.

[20] Li, Hao, et al. "Pruning filters for efficient convnets." arXiv preprint arXiv:1608.08710 (2016).

[21] Ye, Jianbo, et al. "Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers." arXiv preprint arXiv:1802.00124 (2018).

[22] He, Yang, et al. "Soft filter pruning for accelerating deep convolutional neural networks." arXiv preprint arXiv:1808.06866 (2018).

[23] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[24] Wu, Jiaxiang, et al. "Quantized convolutional neural networks for mobile devices." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[25] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

[26] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[27] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[28] Ramachandran, Prajit, Barret Zoph, and Quoc V. Le. "Searching for activation functions." arXiv preprint arXiv:1710.05941 (2017).

[29] Howard, Andrew, et al. "Searching for mobilenetv3." arXiv preprint arXiv:1905.02244 (2019).

[30] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[31] Huang, Yanping, et al. "Gpipe: Efficient training of giant neural networks using pipeline parallelism." arXiv preprint arXiv:1811.06965 (2018).