WEATHER CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Mohamed Elhoseiny[1] *Sheng Huang*[2] *Ahmed Elgammal*[1]

[1]Rutgers University, Piscataway, NJ, 08854, USA
[2]Chongqing University, Chongqing, 400044, P.R.C

ABSTRACT

In this paper, we study weather classification from images using Convolutional Neural Networks (CNNs). Our approach outperforms the state of the art by a huge margin in the weather classification task. Our approach achieves 82.2% normalized classification accuracy instead of 53.1% for the state of the art (i.e., 54.8% relative improvement). We also studied the behavior of all the layers of the Convolutional Neural Networks, we adopted, and interesting findings are discussed.

Index Terms— Deep Learning, Weather Classification, Image Classification, Convolutional Neural Networks, Image Convolutional Activation Feature

1. INTRODUCTION

The weather conditions not only strongly influences us in our daily lives [1] through the solar energy system and outdoor sporting events as examples, but also affects the functionality of many visual systems including outdoor video surveillance and vehicle assistant driving systems [2, 3] (by heavy rain, haze, etc.). It is no doubt that, judging the weather conditions by a single image, also known as weather classification task, plays a vital role in many visual and weather systems. Nowadays, the weather classification task is commonly accomplished by the human vision or expensive sensors. Since weather condition is local to an area, lack of the required human resources and/or the expensive sensors limits the availability of local measurement of the weather condition. Recently, researchers argued that computer vision techniques could be developed to accurately classify weather conditions through images, which might save expensive human and instrumental resources (i.e., sensors) since economical surveillance cameras are ubiquitous and would be sufficient to accomplish weather classification. In this paper, we refer to weather classification from images as the task of predicting the class of the weather given an image (e.g., cloudy, sunny, etc.).

Only very few works have been proposed to investigate weather classification from a single image [1, 2, 3, 4]. Most of these methods can be divided into three basic steps [2, 3, 4]. The first step is to extract the Regions of Interest (ROIs) in a weather image, e.g., sky region and road region extraction,

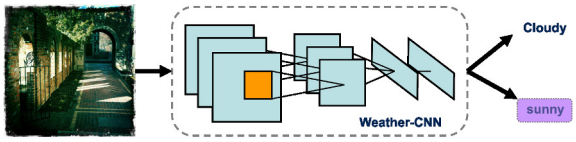

Fig. 1: Weather Prediction CNN-Architecture (Our Neural network follow the CNN-layer specifications by Alex et al [5] in the first seven layers while the output layer (8^{th} layer) is replaced with two nodes, one for Cloudy and one for Sunny)

then use several histogram descriptors to represent the different ROIs and finally some classifiers, e.g., Support Vector Machine (SVM) and Adaboost. These approaches may work well for images captured from the constrained environment but fails for weather classification images, taken in the wild (e.g., classifying an image without any sky region using methods that highly depends on sky features). In order to better address these challenges, Cewu et al recently proposed a complex collaborative learning framework via analyzing multiple weather cues [1]. Specifically, this method involves many pre-processing techniques, such as image segmentation, sky detection, haze detection and boundary detection, which makes the model highly relies on the performance of the aforementioned techniques.

Although the previous works provide interesting solutions for weather classification, the performances of these approaches are unappealing. As far as we know, the best normalized classification accuracy achieved in the challenging weather image dataset, which consists of 10K images, is only 53.1% (the regular accuracy is 76.5%) [1]. Figure 1 (left) shows a challenging image with sunny weather condition. It was also reported in [1] that the histogram of mean lightness of sunny and cloudy images substantially overlap, which makes the dataset very challenging. We attribute the low performance of [1] mainly to the engineered image features adopted in this method. Compared to the typical image classification task, weather classification from images is affected by various factors, e.g., illumination, reflection, scene and shadow. These factors are highly coupled with each other and therefore the categorization manifold is highly nonlinear. Although the previous engineered approaches can satisfy some desirable properties and mitigate some undesirable properties from these factors, they cannot well capture such nonlinearity of the categorization manifold, which makes discrimination

between weather classes a hard problem.

Motivated by the remarkable successes of Convolutional Neural Networks (CNNs) in computer vision and machine learning [5, 6, 7, 8, 9], we adopt CNNs to solve the weather classification task. There are three reasons for us to choose this technique: The CNN is a neural network model which captures nonlinear mapping between the different spaces, e.g.,. feature space and label space; Deep CNN has demonstrated the powerful discriminating power in extensive image representation and classification tasks; CNNs are simple and explicit end-to-end convolutional architectures, which can simplify the weather classification, without the need for engineered features (e.g. HOG [10], GIST [11]).

Most CNN works are designed for addressing the object recognition and detection tasks [5, 6, 8, 9]. However, weather classification is quite different to these issues. It is more sensitive to factors, such as lighting condition and the status of sky and shadows, rather than object-related information, such as shape and texture. This paper focuses on studying the feature spaces introduced by the different layers of CNN in the weather classification task. There are three main questions that we aim to answer:

1. How good is the representation at different layers of a pre-trained CNN for addressing the weather classification problem?
2. How fine-tuning of a pre-trained CNN optimized for weather classification dataset will affect the representation at each layer of the network?
3. How spatial coherence is important in CNN-based weather classification?

We conducted several experiments to address all these questions over different layers of the CNNs for the weather problem. Adopting CNNs, we concluded our work by significantly outperforming the state-of-the-art by 54.8%.

2. METHODOLOGY

Our work is inspired by the recent success of deep Convolutional Neural Network (CNN) proposed by Krizhevsky et al. [5], which is the winner of LSVRC-2012 ImageNet challenge [12]. Krizhevsky's CNN is based on Yann LeCun's architecture, which is used for digit recognition [13]. However, it is composed of 8 layers (deeper than Yann LeCun's CNN) and includes 1000 neurons in the output layer corresponding to 1000 classes). We follow a training paradigm similar to [5] for studying and learning deep representations for the weather classification task.

Figure 1 sketches the CNN-architecture that we adopt to tackle the two-class weather classification task. The first seven layers of this network is defined with the same configuration of [5]. They mainly consists of five convolutional/pooling layers followed by three fully connected layers. In contrast to [5], the eighth (output) layer in the weather classification CNN has two nodes corresponding to sunny and cloudy classes.

2.1. Loss function

The weather classification loss is defined by the multinomial logistic regression objective. We optimized the CNN parameters by maximizing the average of the log-probability of the correct label over the training examples. The multinomial logistic loss is sometimes denoted as softmax loss. We denote the softmax loss of a training example image x with label $l \in \{Sunny, Cloudy\}$ as $loss(x, l)$.

2.2. Training and Testing

The weather classification CNN-models were trained by the back-propagation algorithm with batch stochastic gradient descend (batch size = 50 images) such that the softmax loss is minimized. Since training is done per batch, the losses contributed by each example are summed up for the given batch which is fed to the back-propagation algorithm. The update of the CNN parameters is mainly conducted by propagating the gradient of the $loss(x_i, l_i)$ for all (x_i, l_i) in the training batch. The base learning rate is assigned to $0.5 \cdot 10^{-3}$. The learning rate of the output layer's parameters are assigned to be ten times higher than the learning rate of the parameters of the remaining layers which is assigned to the base learning rate. This is since the parameters of the output layer is initialized randomly, while the parameters of the remaining layers were initialized from the pre-trained CNN on ImageNet [5]. This methodology that started from a pre-trained network and adapt it to a new task, like weather classification in our case, is called fine-tuning. Fine-tuning has been shown to be successful in other tasks like Object Recognition [6, 14].

The decay of the learning rate γ is assigned to 0.1. The policy of the learning rate is step for each 5000 iterations. The momentum and the weight decay were assigned to 0.9 and 0.0001 respectively. Training images are randomly shuffled before feeding the CNN for training. Following the the state-of-art methods for training CNNs, Dropout technique [15] is used for better generalization. We use Caffe Deep Neural Network tool [7] for training and testing the models in this paper.

During training, we augment the training dataset in this study by reflecting the images and randomly sampling 227×227 patches from the downscaled images of size 256×256. At test time the center 227×227 patches are taken. The data and the models are publicly available here [16].

2.3. Studied Layers

We study the performance of all the layers on each of pre-trained CNN [5], and the weather CNN in figure 1. We denote these layers in order as: Pool1, Pool2, Conv3, Conv4, Conv5, Pool5, FC6, FC7, FC8 (output layer) where Pool indicates Max-Pooling layers, Conv indicates layers performing convolution on the previous layer and FC indicates fully connected layer. The last fully connected layer (FC7) is fed to an N-way softmax which produces a distribution over the category labels of the dataset, $N = 2$ for weather classification

oriented CNN and $N = 1000$ for pre-trained CNN on ImageNet, which has 1000 object classes.

In order to study these layers, we extract the activation of each of these layers as a learning representation given an image (a feature vector for each layer). In order to evaluate Pool1 to FC7 in both the ImageNet pre-trained CNN and the weather classification oriented CNN, we apply SVM on these features. Since, the weather-CNN FC8 layer can directly predict the weather class, there is no need to perform SVM to predict the weather class. On the other hand, we learn an SVM classifier on ImageNet-CNN FC8 layer that consists of 1000 dimensions corresponding to ImageNet visual objects which have a semantic meaning. The purpose of this experiment to see whether these semantic dimensions could help do weather classification.

3. EXPERIMENTS

3.1. Experimental Setting

We evaluate the feature representations and the CNNs on the most recent and largest weather image dataset, which has just been released in [1][1]. This dataset has two classes(total of 10,000 images). For the sake of comparison, we adopt the same evaluation metric in [1] which is the normalized accuracy, defined as follows

$$\varrho = \max(\frac{\rho - 0.5}{1 - 0.5}, 0),$$

where ϱ and ρ denote the normalized and regular accuracies respectively.

Following the experimental setting in [1], we randomly select 80% images from each class as training and learning dataset and the remain 20% images are used for testing. We create five random training-test splits by the same percentage and we report the mean of the normalized accuracies over the five splits. For Weather-CNN, we further randomly divide training part for each split into two subsets. The first subset (85% of the training) is used for optimizing/fitting Weather-CNN model and the second subset (15% of the training set) is used for validation. In order to distinguish the CNN model pre-trained by ImageNet, we name our weather classification oriented CNN model as *Weather-CNN* and name the ImageNet pre-trained CNN as *ImageNet-CNN*.

3.2. CNN Models Layer Analysis

We conduct several experiments to perform layer by layer analysis of Weather-CNN model and ImageNet-CNN model on the weather classification task. Figure 2 plots the results and Figure 3 shows the relative improvement of Weather-CNN over ImageNet-CNN in each layer[1]. It is not hard to see that the Weather-CNN features consistently outperforms

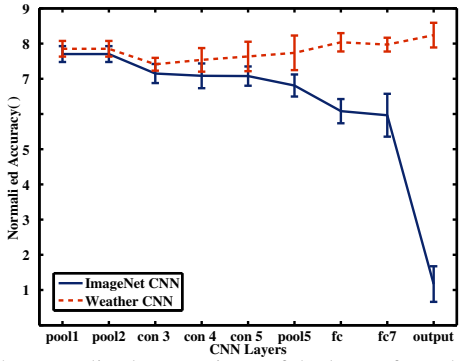

Fig. 2: The normalized accuracies ϱ of the layers from the Weather-CNN and the ImageNet-CNN.

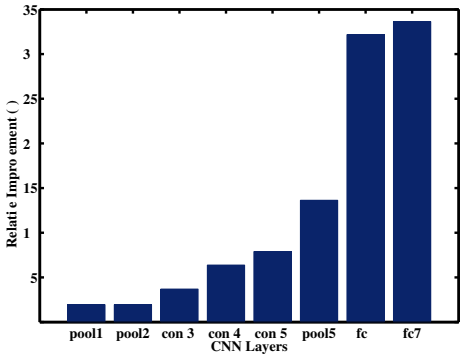

Fig. 3: The relative improvement of Weather-CNN features over ImageNet-CNN features by layer by layer (in normalized accuracy). Note, the improvement of the last layer, which is the label layer, has not been reported.

the ImageNet-CNN features in all the layers. This is since ImageNet-CNN is trained to discriminate between objects rather than discriminating between weather conditions which is captured by Weather-CNN model. Another interesting observation is that the improvement of the Weather-CNN over ImageNet-CNNs significantly increases in high level layers. This is since ImageNet-CNN performance decreases while going up to higher CNN layers. On the other hand, the performance of Weather-CNN goes up in higher layers. We attribute this behavior to following reasons: low-level layers are good at depicting the detailed local characteristics, such as edge, corner point and primitive shape patterns, which are shared by most of the images, while the higher level layers pay more attentions on capturing the abstract and task-specific information, such as object parts, and objects for ImageNet-CNN. Since Weather-CNN is trained for weather classification, this justifies the big gap in the performance between it and ImageNet-CNN in higher levels since Weather-CNN learns weather-related high level information.

3.3. Spatial Analysis of ImageNet CNN Layers

In this experiment, we study the effect of spatial distortion of the images on ImageNet-CNN. The reason why we study ImageNet-CNN is that the experiments in the previous section shows a clear correlation between the level of the layer

[1]http://www.cse.cuhk.edu.hk/ leojia/projects/weatherclassify/index.htm

[1]SVM is the classifier adopted in all experiments except the last layer of the Weather-CNN model, since the output of this layer is exactly the predicted label

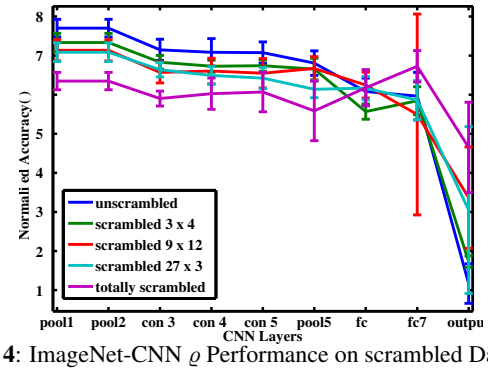

Fig. 4: ImageNet-CNN ϱ Performance on scrambled Datasets

and the performance, where the best performing layer is Pool1 and the performance decreases in the higher layers. One possible reason why Pool1 performs the best for ImageNet CNN is that Pool1 has the most spatial information among all the layers. In order to understand the role of spatial arrangement on weather classification, we spatially scramble all the images in the dataset in 3×4, 9×12, 27×36 grids, and totally scrambled (at pixel level). We scramble an image by dividing it into $m \times n$ equal-size rectangular blocks (i.e., m rows and n columns) and then randomly shuffling them. While, it is obvious that the spatial structure is significantly distorted for big m and n, spatial coherence is preserved within each block when m and n are smaller (e.g., when $m, n = 1$, scrambling has no effect on the image). Then, we evaluated the performance layer by layer of ImageNet-CNN for each of these scrambling settings and also the unscrambled setting.

Figure 4 shows ϱ performance layer by layer for the unscrambled, 3×4, 9×12, 27×36, and totally scrambled versions of the dataset. The results shows a drop of Pool1 features from 77.02% (unscrambled) to 63.48% (totally scrambled). This behavior is consistent until FC6, where the performance starts to get unstable with high variance and gets more similar. This is since FC6 contains high level cues which might be equally effective as cues trained for object recognition rather than weather classification. Our conclusion for these experiments is two-fold. First, spatial distortion of the images highly degrades the performance in the lower layers (e.g., Pool1) compared to higher layers (e.g., FC6). Second, even with high spatial distortion, the Image-Net CNN at lower layers (e..g, Pool1) still performs well at the rate 63.48%, which indicates that the low-level cues are still good to discriminate between the two weather classes (63.48% is $\approx 10\%$ better state-of-the-art performance, discussed next).

3.4. Comparison with Engineered Representations and the State of The Art

We start by comparing the CNN features with engineered features including the combined feature in [1], and two other engineered features, namely HOG [10], GIST [11]. For classification, linear SVM is applied for all the engineered features; see table 1 top 3 rows. The results shows a big gap between these engineered features (top 3 rows in table 1) compared to

Table 1: The performances (mean±std, in percents) of different weather image representations where NormAcc = Normalized Accuracy and Acc = Regular Accuracy.

Methods	NormAcc (ϱ)	Acc (ρ)
GIST [11]+SVM	11.3±2.0	55.7±1.0
HOG [10]+SVM	38.5±2.4	69.3±1.2
Combined Feature [1]+SVM[2]	41.2±2.2	70.6±1.1
Yan et al. [3][2]	24.6±2.6	62.3±1.3
Roser et al. [2][2]	26.2±2.3	63.1±1.2
Lu et al. [1][2]	53.1±2.2	76.6±1.1
ImageNet-CNN(Pool1)+SVM	77.0±2.3	88.5±1.2
Weather-CNN(FC7)+SVM	80.0±2.9	90.0±1.5
Weather-CNN (Output)	**82.2±3.5**	**91.1±1.8**

CNN approaches (bottom 3 rows). We argue that there are two main cues behind the remarkable success of CNN-based weather classification in contrast to engineered learning representation. The first one is that CNN is a powerful Neural Network model which is good at finding the nonlinear mapping. The second one is that CNN is pretrained on one million images (using ImageNet data) which takes extensive factors into the consideration including low-level abstraction.

We also compare Weather-CNN with ImageNet-CNN and other state-of-the-art weather classification approaches; see table 1 (bottom six rows). It is not hard to see that all the CNN-based approaches significantly outperform the conventional state-of-the-art methods. The normalized accuracy gain of ImageNet-CNN over Lu's work [1] is 23.9%, which is the most recent and the best performing weather classification approach. Note, in the experiment of ImageNet-CNN, the most discriminative layer is selected as the image representation (i.e., Pool1; see section 3.2 for the discussion of layer activation) and the linear SVM is employed for classification. Similarly, Weather-CNN achieves 3% gain in normalized accuracy over ImageNet-CNN by applying SVM on FC7. The last layer of Weather-CNN has two-dimensional output layer (FC8), which directly outcomes the predicted weather label of an input image. Directly from output (FC8) layer, the Weather-CNN achieves normalized accuracy of 82.2% (91.1% in regular classification accuracy), which is 5.2% better than ImageNet-CNN (77.0%). Furthermore, this concludes that the relative improvement of the Weather-CNN approach over the best performing state-of-the-art method [1] is 54.8%. We think that the good performance of the CNNs emerges mainly from its impressive abilities to capture the non-linearities of the manifold of weather conditions.

4. CONCLUSION

In this work, we analyzed the recognition performance for the layers of both pretrained imageNet-CNN and Weather-trained CNN. We also studied the performance drop under spatial distortion for the layers. We concluded our work by Weather classification results outperforming the state-of-the-art by a huge margin (82.2% compared with 53.1%).

[2]The results are referenced from [1] under the same experiment settings.

Acknowledgment: This research was supported in partially funded by NSF awards IIS-0905647. The GPUs used in this research were generously donated by the NVIDIA Corporation.

5. REFERENCES

[1] Cewu Lu, Di Lin, Jiaya Jia, and Chi-Keung Tang, "Two-class weather classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3718–3725.

[2] Martin Roser and Frank Moosmann, "Classification of weather situations on single color images," in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 798–803.

[3] Xunshi Yan, Yupin Luo, and Xiaoming Zheng, "Weather recognition based on images captured by vision system in vehicle," in *Advances in Neural Networks*, pp. 390–398. 2009.

[4] Zichong Chen, Feng Yang, Albrecht Lindner, Guillermo Barrenetxea, and Martin Vetterli, "How is the weather: Automatic inference from images," in *IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 1853–1856.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems (NIPS)*, 2012, pp. 1097–1105.

[6] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *ACM International Conference on Machine Learning (ICML)*, 2014.

[7] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM International Conference on Multimedia*, 2014, pp. 675–678.

[8] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *International Conference on Learning Representations (ICLR)*, 2014.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

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

[11] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, 1998.

[14] Georgia Gkioxari, Bharath Hariharan, Ross B. Girshick, and Jitendra Malik, "R-cnns for pose estimation and action detection," *CoRR*, vol. abs/1406.5212, 2014.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, 2014.

[16] Mohamed Elhoseiny, "Weather cnns data and models," `https://sites.google.com/site/mhelhoseiny/computer-vision-projects/weather-classification-cnn`, Accessed May 21, 2015.