Identifying Clouds with Convolutional Neural Networks

Jeff Mullins ’18, Sean Richardson ’20, Peter Drake

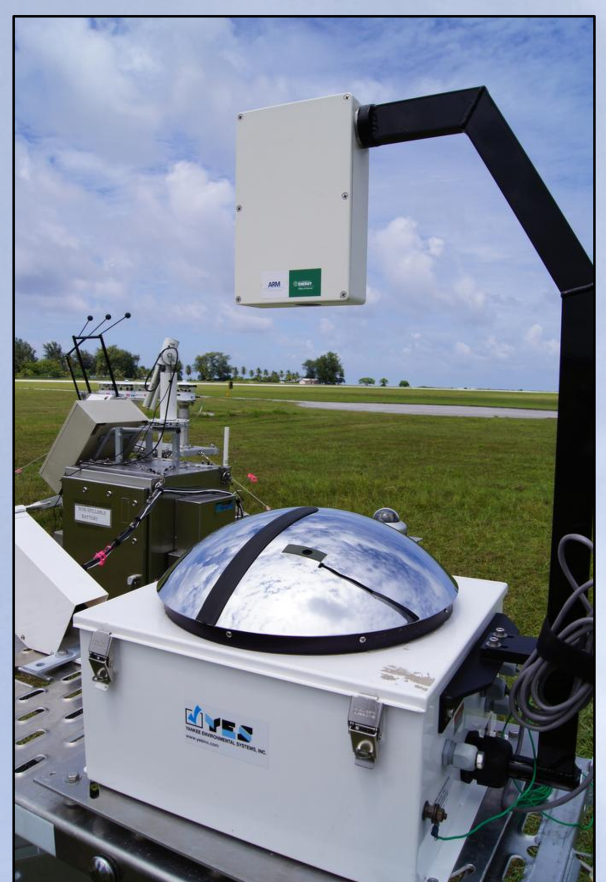
Lewis & Clark College

Abstract

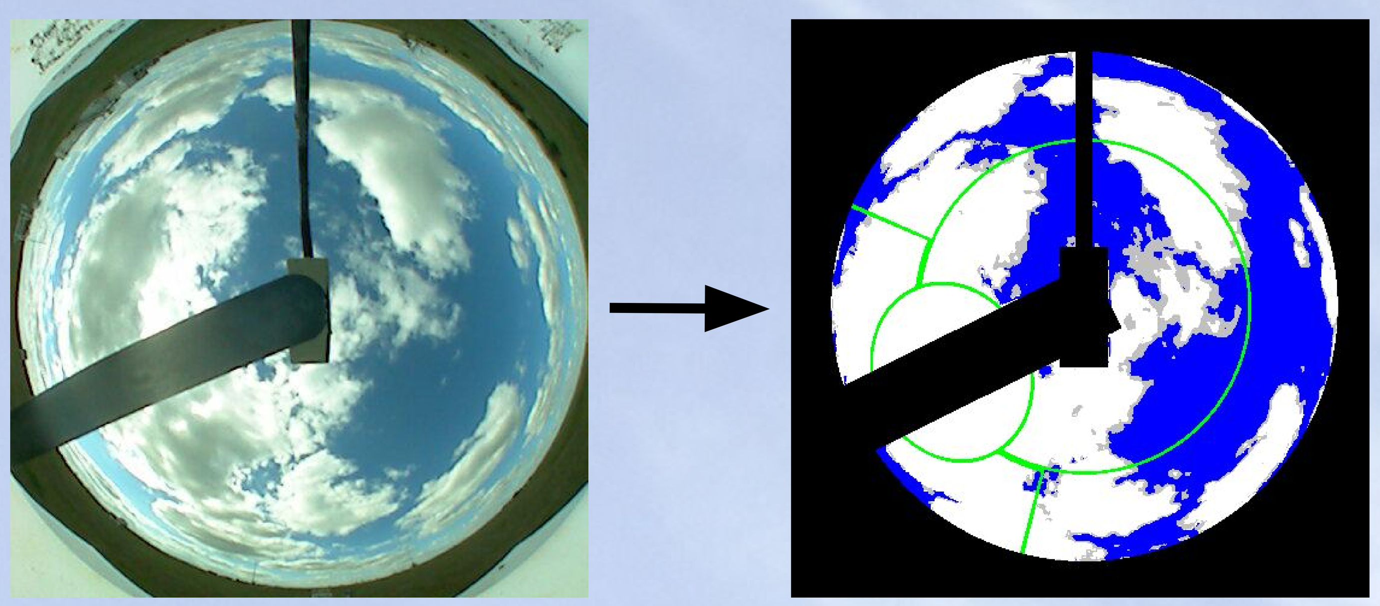
The greatest source of uncertainty in model estimates of projected climate change involve clouds and aerosols [1]. Photographic images of clouds in the sky are simple to acquire and archive, but climate scientists need an automated process for identifying clouds in these images. We bring machine learning to bear on this problem. Specifically, we use convolutional neural networks [2], which to our knowledge have not previously been applied to this task [3]. We trained a network to identify clear sky, thin cloud, thick cloud, and non-sky pixels in photos taken by the Total Sky Imager [4]. The trained network is capable of classifying 91.9% of pixels correctly. An ensemble of several networks increases this to 94.6%.

Segmenting TSI Images

The Total Sky Imager (TSI) consists of a downward facing digital camera directed at an upward facing convex mirror to produce photographs of the sky [4]. The TSI takes a picture every 30 seconds, so millions of images are available [5]. Several TSI devices are in use around the world; we draw our data from a climate research facility in Oklahoma. We use 467,906 pictures taken from January to November of 2006.



Cloud segmentation is the task of labeling each pixel in an image as clear sky, thin cloud, thick cloud, or non-sky. Non-sky pixels include the corners of the image, the reflection of the camera arm, and the moving shadowband that blocks sun glare.

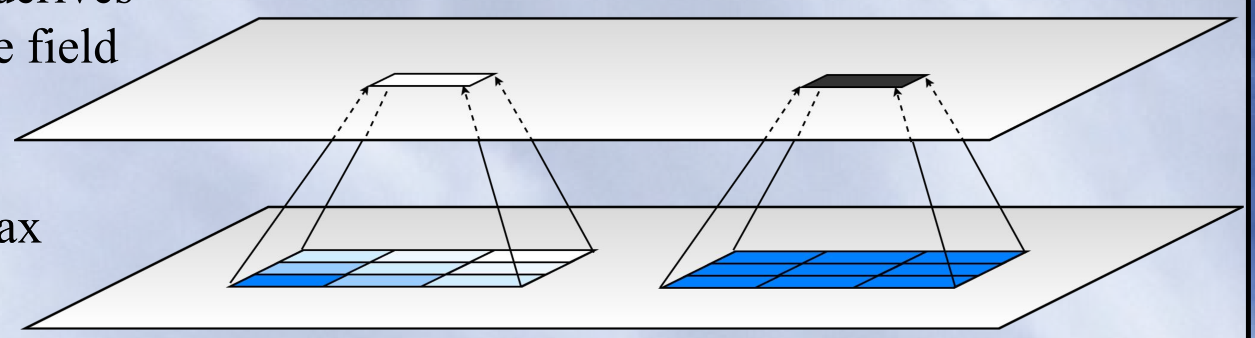


Sky image *Segmentation*

TSI incorporates a simple cloud segmentation algorithm that compares pixel values to thresholds. This can be prone to errors and requires tedious hand tuning to account for the specific device location and lighting conditions.

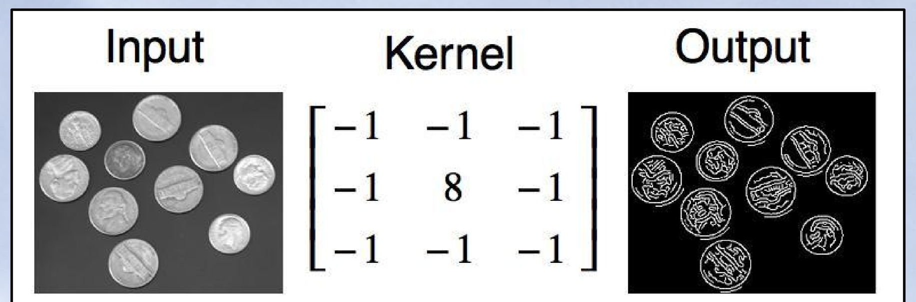
Convolutional Neural Networks

A convolutional neural network learns to recognize patterns by generalizing from many examples. Such networks have been very successful in artificial vision tasks [2]. The network consists of a series of layers, each of which computes a modified version of the output of the previous layer. The first layer takes the image as input. Each pixel in a layer derives from a small receptive field in the previous layer, using functions such as convolution and max pooling.



Convolution

A convolutional layer contains a matrix of numbers called a kernel. The kernel is combined with the receptive field (using a dot product) to produce each output pixel. This operation is effective for detecting edges and other features. Learning in the network involves adjusting these kernels to produce the desired output.



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Edge detection using convolution

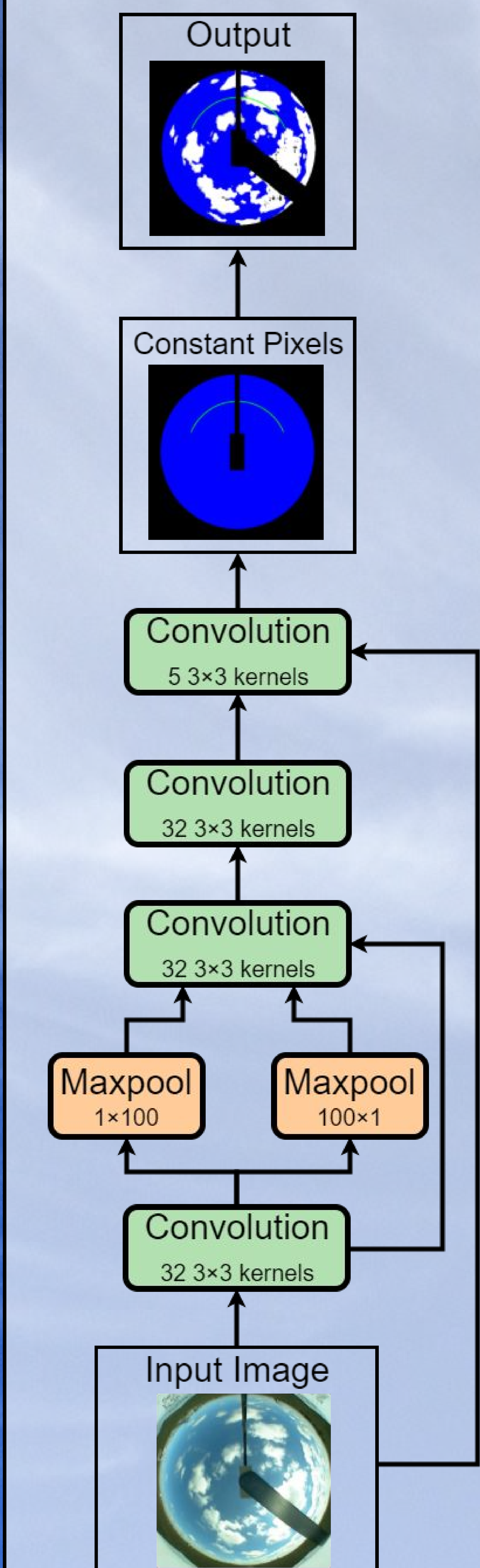
Images: https://commons.wikimedia.org/wiki/File:Edge_detection.png

Max Pooling

A max pooling layer simply outputs, for each pixel, the largest value in its receptive field. This allows the layer to determine if some feature (detected by a previous convolutional layer) is present anywhere in the receptive field.

Our pilot experiments had trouble identifying the shadowband because their receptive fields were too small. Very large receptive fields in max pooling layers (e.g., 100×100) were too computationally expensive. We solved this by including two long, skinny max pooling layers, with 1×100 and 100×1 receptive fields.

Architecture



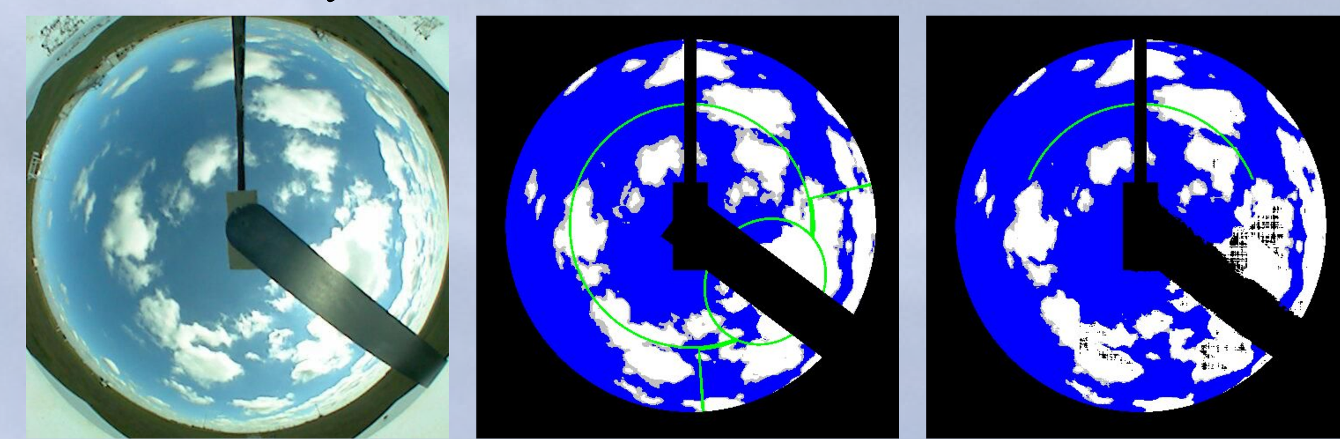
Source code for the network is available at *https://github.com/VISTAS-IVES/sky*

Training

Fifteen copies of the network were trained for several days on Portland State University’s Coeus computing cluster. We trained the network using gradient descent, minimizing cross entropy between the network output and the target “correct answer” (TSI’s segmentation image). In each training step, we selected (without replacement) 50 random images from our data set. The network was not burdened with classifying pixels that were the same in all of the target outputs.

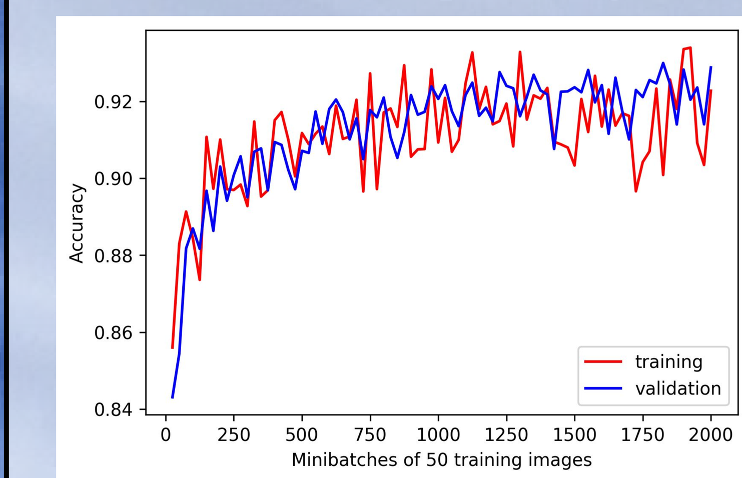
Results

Each network achieved an accuracy (on images that it hadn’t seen before) of 91.9%. An ensemble of all fifteen networks, voting on each pixel, had an accuracy of 94.6%. On the specific image below, accuracy was 94.8%.



TSI image *TSI segmentation* *Network output*

Intriguingly, even though our networks were only trained to match the default TSI segmentation, the ensemble outperforms the TSI in some respects, e.g., labeling birds as non-sky.



The learning curve at left shows our best network’s increase in accuracy as training proceeds; other runs were similar. It does not appear that additional training would improve performance.

Future Work

We hope to move on from *segmentation* to *classification*, identifying which types of clouds (cumulus, etc.) are present in an image. Other data sources of data (radar, satellite images, manual observations by meteorologists, etc.) may help in this more challenging task.

Acknowledgments

- Jessica Kleiss
- Portland State University
- Lewis & Clark College
- TensorFlow
- Atmospheric Radiation Measurement (ARM) Climate Research Facility
- VISTAS team including Jenny Orr
- John S. Rogers Science Program
- M. J. Murdock Charitable Trust

Works Cited

[1] O. Boucher, *et al*., “Clouds and Aerosols,” in *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, Stocker *et al*., Cambridge, UK and New York, NY, USA: Cambridge University Press, 2013, pp. 571-657.

[2] I. Goodfellow, *et al*., *Deep Learning*. Cambridge, MA: MIT Press, 2016

[3] R. Tapakis and A. G. Charalambides, “Equipment and methodologies for cloud detection and classification: A review,” *Solar Energy*, vol. 95, pp. 392–430, Sep. 2013.

[4] C. N. Long, J. M. Sabburg, J. Calbó, and D. Pagès, “Retrieving Cloud Characteristics from Ground-Based Daytime Color All-Sky Images,” *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 5, pp. 633–652, May 2006.

[5] https://www.arm.gov