

Machine Learning for Retrieving Cloud Optical Thickness from Observed Reflectance: 3D Effects

CyberTraining: Big Data + High-Performance Computing + Atmospheric Sciences

Kallista Angeloff¹, Kirana Bergstrom², Tianhao Le³, Chengtao Xu⁴

Research assistants: Chamara Rajapakshe⁵, Jianyu Zheng⁵

Faculty mentor: Zhibo Zhang⁵

¹Department of Atmospheric Sciences, University of Washington

²Department of Mathematical and Statistical Sciences, University of Colorado Denver

³Division of Geological and Planetary Sciences, California Institute of Technology

⁴Department of Electrical Engineering, Embry-Riddle Aeronautical University, Daytona Beach

⁵Department of Physics, University of Maryland, Baltimore County

Technical Report HPCF-2020-15, hpcf.umbc.edu > Publications

Abstract

Clouds are inherently three dimensional (3D), and simulating radiative transfer (RT) properties accurately requires models that take their 3D effects into account. Because 3D models are complex and computationally expensive, RT models often use simplified 1D models to retrieve cloud properties, which suffer from retrieval uncertainty and sometimes significant biases due to 3D effects. Recent advancements in machine learning may lead to a retrieval algorithm that is capable of taking these effects into account. We will develop a machine-learning based cloud property retrieval algorithm that is able to reconstruct the 3D structure of clouds based on observed cloud radiative signatures.

Key words. Machine learning, radiative transfer, cloud optical depth, 3D cloud effects

1 Introduction

Clouds are the most important, most variable, and one of the most difficult component to model in global climate models. The radiative impact of clouds is controlled by their optical properties. One of the most important of these is cloud optical thickness (COT), the vertical optical depth of the cloud along the path of its thickness. COT quantifies the degree to which a cloud scatters and reflects visible light (virtually none of which is absorbed) rather than allowing it to pass through to the Earth's surface.

All clouds reflect some incoming shortwave radiation and absorb some outgoing longwave radiation. However, optically thick clouds (high COT) generally reflect sufficient shortwave radiation such that their retention of surface-emitted longwave radiation is offset, producing an overall cooling effect in the climate system. Optically thin clouds (low COT) do not reflect sufficient shortwave to offset their longwave retention and so have an overall warming effect. It is therefore necessary to consider COT on a global rather than local basis to understand the net planetary effect. While it is possible to estimate COT values from surface measurements [1], an accurate means of retrieving COT from satellite observations is essential to understanding the influence of clouds across a global scale.

The established method for calculating COT [6] uses radiance data from multiple wavelengths to estimate COT and cloud drop effective radius (CER), a measure of the size distribution of water

droplets within a cloud. The radiance is a function of the cloud’s COT and CER, and so finding those properties presents an inverse problem. This is achieved by measuring the radiance at pairs of non-absorbing and absorbing wavelengths (e.g. as in Nakajima and King’s original 0.75 and 2.16 μm , or as in the most recent MODIS products 0.66, 0.86, and 1.24 μm and 1.6, 2.1, and 3.7 μm), and comparing the results with a theoretical forward model calculation using a lookup table.

There are many simplifying assumptions inherent in this calculation. Our main concern is to address two of these assumptions. One is the plane-parallel assumption: all clouds are treated as a flat plane parallel to a flat ground surface. The second is the independent pixel assumption: each point within the cloud plane is assumed to be independent from every other point, rather than being subject to shadowing or reflected light. These assumptions are closest to accuracy when the real cloud resembles the model cloud, e.g. stratus clouds, but most real clouds have significant horizontal and vertical variability that cannot be fully represented by a 1D radiative transfer model; that is, they have peaks, valleys, and gaps that generate horizontal photon transport. Differing locations within a cloud receive different amounts of light based on their positions relative to each other.

An array of 3D radiative transfer models are currently in use and/or development. However, 3D radiative transfer models have so far been prohibitively computationally expensive to conduct at a global scale over the timescales necessary for climate modelling, and 1D models remain more widely used. The differences between the results of 1D and 3D radiative transfer models are called the 3D radiative transfer effects. These differences can be significant, although they do not inevitably introduce large errors [7]. This project expands upon a novel method for determining COT from synthetic reflectance data using machine learning.

2 Atmospheric Model

The use of neural networks to improve COT retrievals is an ongoing effort. Okamura et al. (2017) [4] used a deep neural network (DNN), examining multiple pixels simultaneously, to retrieve the COT and CER from synthetic cloud reflectances. Masuda et al. (2019) [3] similarly used a convolutional neural network (CNN) to improve 3D retrievals of COT from sky-view camera images. In this project, we expanded upon the work of Rajapakshe and Zhang (2020) [8] to experiment with using both a CNN and DNN to correct for 3D effects.

To provide training data for our model, we first generated 4000 profiles of “fractal clouds” - 1D horizontal lines of cloud that have been randomly generated [5] to provide data analogous to a transect across a large area of broken clouds. The fractal clouds have a semi-randomly varying liquid water path (LWP) - a measure of the weight of liquid water between two points in the atmosphere - and a randomly assigned, fixed-per-profile CER. From these two variables, COT can be calculated for each point along the horizontal line.

The synthetic fractal clouds’ features were then applied into 1-D and 3-D radiative transfer models to calculate the reflection with wavelength smaller than 1.8 μm . A visualization of SHDOM 3-D radiative transfer and MSCART 1D and MSCART 3D results are provided in the following figures.

Figure 2.1 shows the case of reflection with fractal cloud generated on 0.865 μm , $maxiter = 1000$, 10^{-4} solution accuracy. The general similarity of the pattern from three RT models reveals the illumination effects brought by the fractal cloud. In the strongest illumination points with largest tau value jumps, the SHDOM model follows the MSCART 3D results better than the MSCART

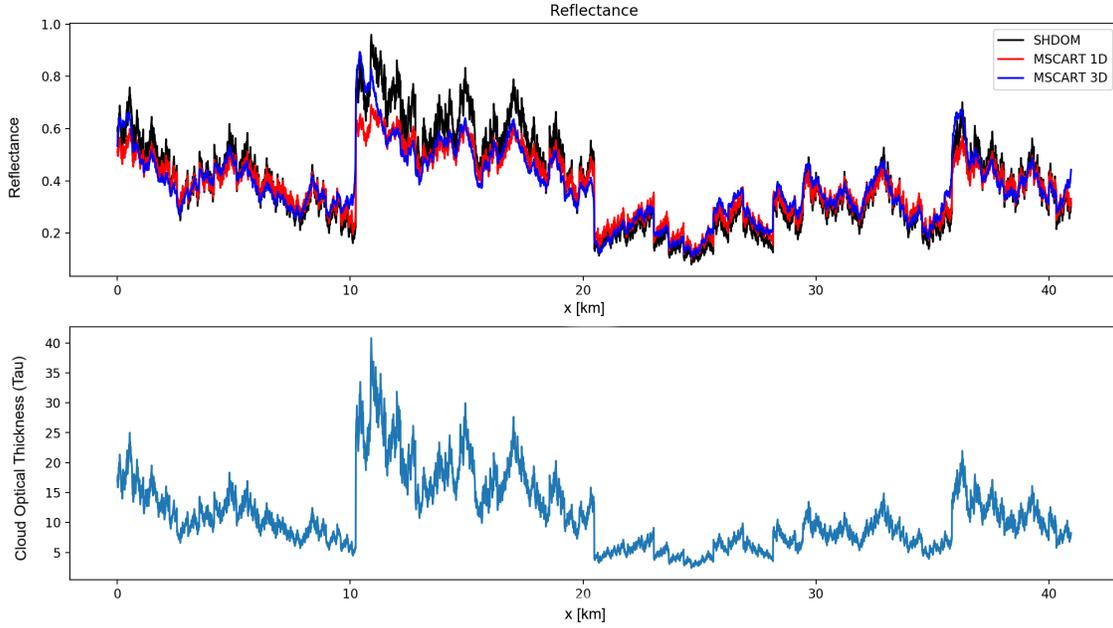


Figure 2.1: Radiative Transfer (SHDOM) Results from Fractal Model. Given a COT (τ) profile, represented in the lower panel, the reflectance results produced by SHDOM (black curve), MSCART 1D (red curve) and MSCART 3D (blue curve) are represented in the top panel.

1D results. However, the predictions of the SHDOM model has similar reflection feature patterns compared with the MSCART 1D RT model. We considered that this could be caused by the Mie scattering and bulk averaging routines within the SHDOM model.

To verify the 3D radiative transfer effects from the SHDOM model, a step cloud case is useful. As shown in Figure 2.2, the edges of the step cloud, which are unrealistically sharp when using a 1D radiative transfer model, are smoothed to account for the effects of the nearby pixels when using the 3D radiative transfer model SHDOM. Using this simple test case is a good “sanity check” to ensure that the models are working as expected.

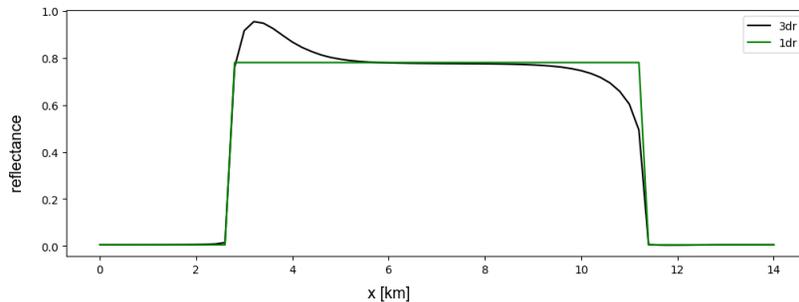


Figure 2.2: Visible reflectance of step cloud from SHDOM. The green curve is based on 1D radiative transfer simulation and the black curve is based on 3D radiative transfer simulation.

2.1 Problem Domain

The scope of this project is to use the machine learning model to solve the inverse problem on a 1D domain. In practical application, we would use the same method for a 2D domain from a satellite image. Envisioning the Earth’s surface, seen from a satellite, as a flat 2D area, the 1D domain is a single slice of this image. Arbitrarily, we use a slice in the y direction, so the x coordinate determines spatial position. The z coordinate is height. The domain of interest starts arbitrarily at $x = 0$ and is $4096 \times dx$ in spatial length.

The step cloud is generated with the following parameters: $\lambda = 0.865 \mu\text{m}$, $Nz = 2$, $dz = 1\text{km}$ (height in vertical direction is 1km), $Nx = 101$, $dx = 0.5\text{km}$, $Ny = 1$, $CEV = 0.05$. These are parameters necessary for the fractal cloud and SHDOM models.

The reflectance is calculated as

$$\text{reflectance} = (\pi \times \text{rad}) / \cos(\text{SZA}) \quad (2.1)$$

where rad denotes the SHDOM radiance output, and SZA represents the solar zenith angle. We generated 3 channels of light in different visible wavelengths $0.645 \mu\text{m}$ (red light) $0.555 \mu\text{m}$ (green light) $0.469 \mu\text{m}$ (blue light).

3 Machine Learning Model

Determining the COT from reflectance is a classical inverse problem. We are given a model, Q , that takes COT as input and produces reflectance as output, which we write as

$$Q(\text{COT}) = \text{reflectance}, \quad (3.1)$$

and we desire, given reflectance, to find the COT that produced that reflectance, or to be able to reproduce the mapping Q^{-1} , where

$$Q^{-1}(\text{reflectance}) = \text{COT}. \quad (3.2)$$

Inverse problems are typically solved deterministically using optimization methods; however, in recent years less traditional methods such as machine learning have become popular for problems of this type [10]. The standard optimization solution for the map in Eq. (3.2) utilizes either the lookup table method, which has limitations for even the step clouds that we have seen in 2, or a much more complex 3D inversion method that is prohibitively expensive for practical applications. We attempt to solve the problem in a more efficient way by approximating the function Q^{-1} with a surrogate model formed using machine learning methods.

3.1 Machine Learning Model

The general procedure of the machine learning model for both the CNN and DNN is shown in Figure 3.1. First, we generate realistic COT profiles using the fractal cloud model. We then apply a 3D radiative transfer model to these cloud fields, which produces cloud reflectances for three separate wavelengths. The CNN model interprets the wavelengths as color channels.

We use the COT profiles generated from the fractal cloud model, and propagate these profiles through SHDOM in a Monte Carlo fashion to generate a set of reflectance profiles. We then used 95% of the generated data to train our machine learning model, and 5% as the testing data set to verify the result.

The deep learning models are built using the Keras deep learning package [2] for Python. In the following, we introduce the tested model structures for the DNN and CNN methods attempted, discuss data pre-processing with each model, and discuss some of the chosen parameters.

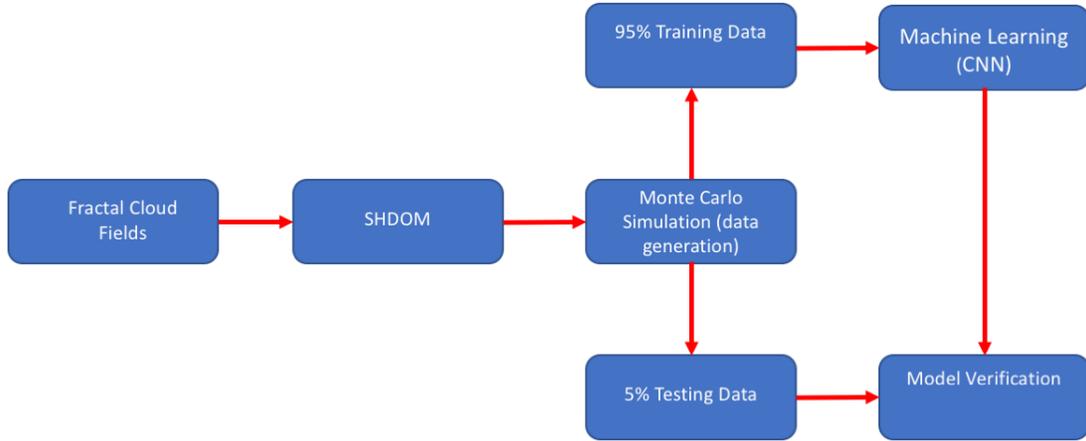


Figure 3.1: Machine Learning Model with 3D Radiative Transferring Data

3.2 DNN Structure

Figure 3.2 represents the general process of how the input and output data are processed in the DNN structure. The input vector is of length $3 \times Nx$, three channels over the whole spatial domain.

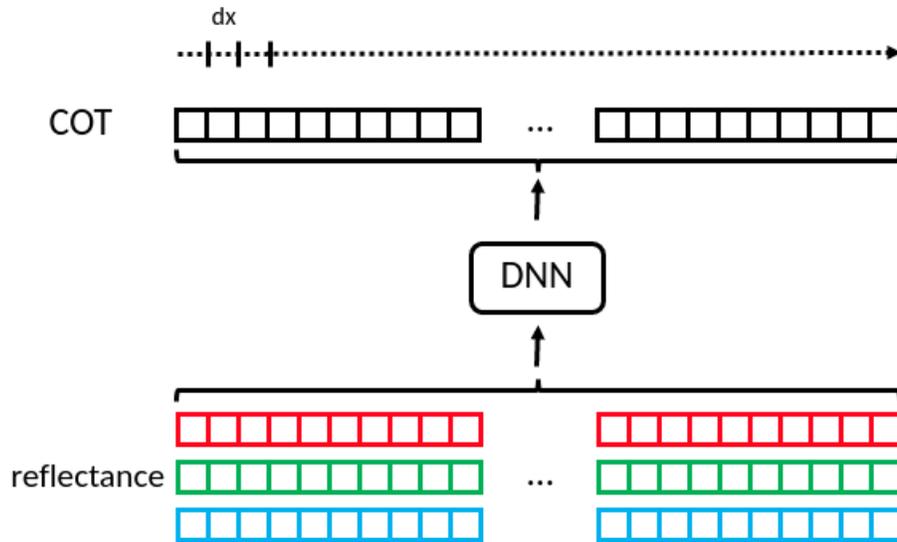


Figure 3.2: DNN Machine learning Model

The DNN structure listed below is based on the DNN-2r structure in [4]. The optimizer and loss function are chosen as adam and mean square error, respectively. To extract an efficient amount of data from the fully connected layers, we adopted a deep and fully connected layer structure. The first layer, which has an 8-neuron density, was chosen to extract the initial features from the input reflectance data with 4096×4000 spatial cloud reflectance information and 3 dimensional features,

which is activated with the relu function. The less complex neuron density was chosen due to the (assumed) less complex input data feature dimensions. A linear activation function is chosen in the final layer with 1 neuron density to generate the 1D cloud property that is optical thickness. The batch size is chosen as 4 under this structure. Normally, we choose the batch size as a power of two to save the computational resource of the model.

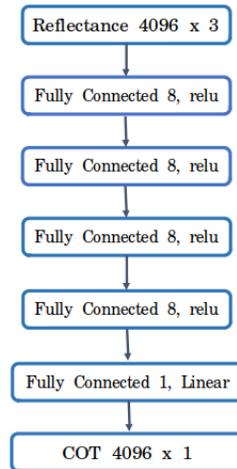


Figure 3.3: DNN Model Structure

As we will see in the results, the network structure did not contain enough features, leading to inaccurate model estimation of cloud optical thickness despite the MSE converging to a relatively low value.

3.3 CNN Structure

3.3.1 Spatial Slicing

The aim of the project is to estimate the COT over large spatial areas. The naive approach to building the machine learning model takes reflectance input over all three channels and the entire spatial domain, and outputs COT for over the whole domain. While this approach would most likely be effective when massive quantities of data are available, as it takes into account the reflectances across the entire domain, the amount of data required to make accurate predictions for the whole domain is larger than we were able to generate during the span of this project.

While developing the CNN model, we implemented a more effective strategy that takes into account the independent pixel approximation problem and is more accurate with less data, and is more computationally efficient. The strategy involves dividing the spatial domain up into smaller subdomains and solving subproblems on these domains. For our 1D problem, the x dimension is sliced into smaller domains of the same sizes of np “pixels.” Each slice is treated as an independent machine learning problem. In order to incorporate the reflectances of nearby pixels, ng “ghost pixels” on the boundaries of the domain are incorporated into the input. These ghost pixels are passed in as components of the input reflectance vectors, but the COT (output) is not computed on these pixels. In this way, we are able to split up the problem into many smaller subproblems, but still incorporate necessary neighboring information as needed to correct for the independent pixel approximation. This is shown schematically in Figure 3.4.

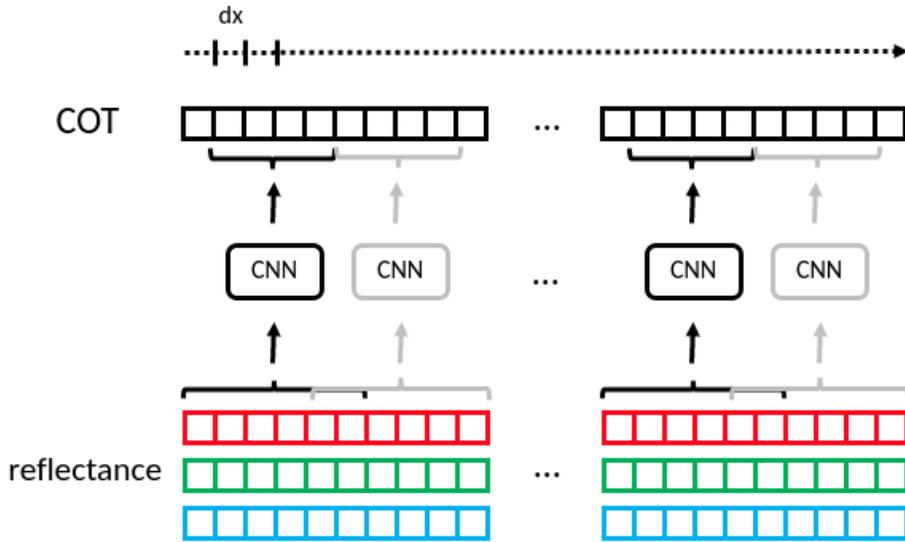


Figure 3.4: CNN Machine learning Model

Naturally, at the edge of the domain the approximation will not work as well because the slices do not have available ghost pixels at the edge of the domain. We ignore this issue by limiting our predictions to starting at the ng th pixel, and ending at the $(n - ng)$ th pixel. In addition, we operate on a reduced domain to ensure that np evenly divides the total size of the domain. For the purposes of this problem, we use $np = 8$, and $ng = 2$. We limit our domain of interest to 4000 pixels; thus we run 500 independent ML models.

Spatial slicing is of particular interest to us because it allows us to solve the problem on each slice independently, thus in parallel, dramatically reducing the computational time necessary to train. We solved this problem using 8 nodes with 8 processes on each, which we found to be optimal due to massive memory allocations within the code.

3.3.2 CNN Structure

The CNN structure is defined in 3.5. It is a standard simple CNN model that uses two convolutional layers. Because of the spatial slicing implemented, the input 1D slices are 12×3 (2 ghost pixels at the left of the slice, 8 pixels in the slices, 2 ghost pixels at the right edge, and 3 channels). The first layer of the CNN is a convolutional layer that uses a kernel of size 6, which is half of the total spatial size of the slice, and 100 ($\times 3$, for each channel) filters. The second layer is again convolutional and uses a kernel of size 1, and 4 filters. A dropout layer is utilized, and a final fully connected layer transforms the learned features into an 8 dimensional vector for the spatial slice.

These parameters were chosen through a combination of research, basing our model off the methods in [4] and standard simple CNN models, as well as parameter tuning. We varied batch sizes, filter sizes, and dropout factor through ranges in order to determine the most effective combinations. Although the search was not exhaustive, no combinations tested in tuning provided consistently better results.

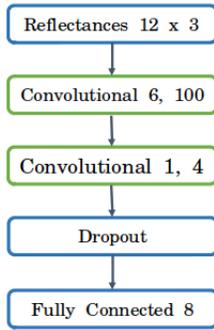


Figure 3.5: CNN Model Structure

4 Results

4.1 DNN Results

A sample result of prediction using the DNN model is shown in Figure 4.1. The true value represents the COT data in the test data set. The starting COT value at the beginning of the spatial interval, where the spatial position of the cloud is over 0 to 2.5km, shows better estimation results. As the cloud spatial position increases, the DNN model function performs worse. We found this pattern consistently throughout the testing data, and believe the reason for this result is the overfitting of the data in the less complicated neural network structure. The trade-off between the complexity of the neural network structure and the leak of number of training data features constrains the estimation results of the DNN. Running 10 epochs, we reach a loss value mean squared error of 15 in the final epoch of model fitting.

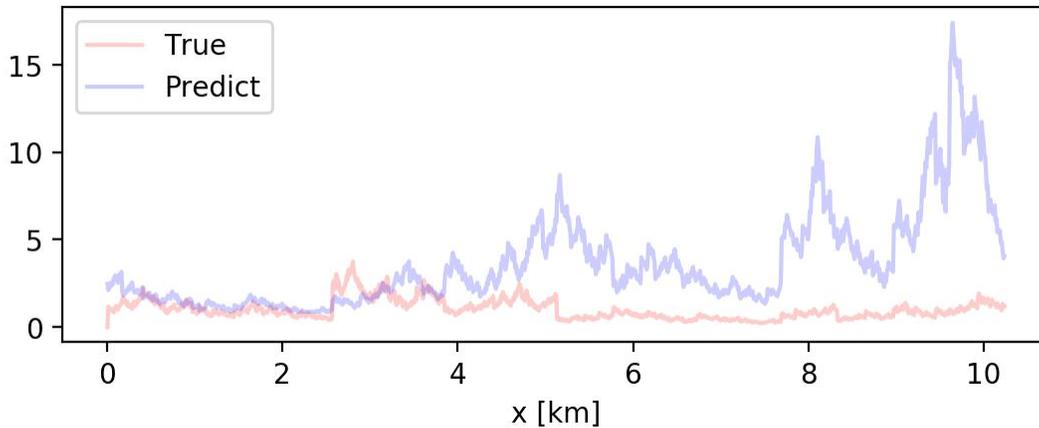


Figure 4.1: Sample DNN Training Result

4.2 CNN Results

Some sample results of prediction using the CNN model are shown in Figure 4.2.

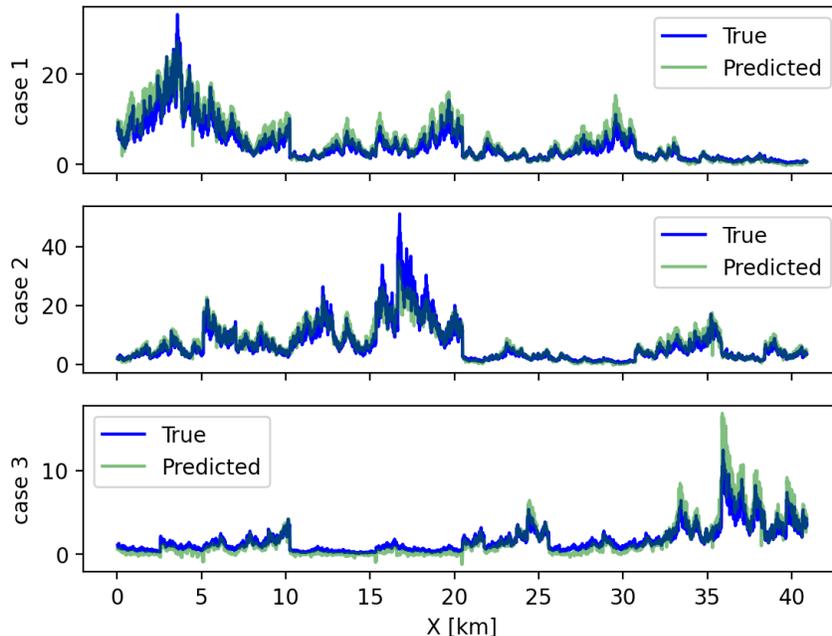


Figure 4.2: Sample CNN Training Results. The three panels represent the output of the model for three different reflectance profiles from the test data set.

Qualitatively, we can see that the CNN model is able to predict the true COT relatively well, when compared to the DNN method. The overall shape and jumps within the COT are reflected in the predicted COT from our model. However, as expected, the global effects of the COT are not captured particularly well in the predicted COT. For example, in case 3, we can clearly see that the predicted COT is lower than the true COT consistently over the first 10 km of the spatial range. For more extreme values, such as at around 16 km in case 2, the model is not able to predict the highest range of the COT. After tuning, the average mean squared error (MSE averaged over each slice of the spatial domain) is around 25, indicating that the method requires improvement.

We suspect that some of these problems are caused by the spatial slicing, as the global effects (the mean over the spatial range is a good indicator of the global effects) are not captured well, but local effects are. We also suspect that some of the issue would simply be solved by more or higher quality data, as COT over 20 is not a realistic case.

5 Conclusions

In this project, we intended to retrieve COT, or other cloud properties, using a machine learning method to approximate a mapping between reflectance and cloud properties. For this project, the synthetic cloud data was generated from a 1D fractal cloud model. We spent time tuning our models, and presented the comparison between different radiative transfer models with 1D MSCART, 3D MSCART, and SHDOM. To generate enough of a training set as the input for the neural network, we adopted the SHDOM model to generate the corresponding data set of 4000

profiles, from which we ended up with the input/output pairs of COT and reflectance data to the forward model predicting reflectance from COT.

To find the best version of a neural network that maps the reflectance data to COT, we developed and compared DNN and CNN models. The results show that CNN with data pre-processing and “ghost pixel” edge slicing generates a better model than the general DNN with fully connected layers. The method is qualitatively acceptable, but there remain issues in the global predictability of the model and for outlying cases. Also, due to the speed of the SHDOM simulations, we don’t get enough data for our CNN model. So although we used a 95/5 training/test split ratio here, we will use better split ratio (e.g., 80/20) once we get more data.

Future directions for this work including refining the model utilizing greater quantities of data, and refining the data generation so that the generated outputs are generally more physically realistic by either data pre-processing (removing outliers) or tuning the way COT profiles are generated. We are also interested in fixing the global predictability issue by emulating a multi-scale model that uses the DNN-based global method for predicting some global characteristics of the profiles, such as average COT and maximum COT, and feeding these global characteristics into the CNN model as additional inputs that could better inform the CNN over the small spatial slices of the general global trends of COT.

In addition, in future work a 2D model should be tested, as the 1D case has little practical application. We would also be interested in applying this method for computing the CER in addition to the COT. Once this is implemented using the trained CNN model, the retrieving of COT and CER could be applied in real applications on 2D satellite data, such as from MODIS climate satellite reflectance data.

Acknowledgments

This work is supported by the grant “CyberTraining: DSE: Cross-Training of Researchers in Computing, Applied Mathematics and Atmospheric Sciences using Advanced Cyberinfrastructure Resources” from the National Science Foundation (grant no. OAC–1730250). The hardware used in the computational studies is part of the UMBC High Performance Computing Facility (HPCF). The facility is supported by the U.S. National Science Foundation through the MRI program (grant nos. CNS–0821258, CNS–1228778, and OAC–1726023) and the SCREMS program (grant no. DMS–0821311), with additional substantial support from the University of Maryland, Baltimore County (UMBC). See hpcf.umbc.edu for more information on HPCF and the projects using its resources.

References

- [1] Nobuhiro Kikuchi, Teruyuki Nakajima, Hiroshi Kumagai, Hiroshi Kuroiwa, Akihide Kamei, Ryosuke Nakamura, and Takashi Y. Nakajima. Cloud optical thickness and effective particle radius derived from transmitted solar radiation measurements: Comparison with cloud radar observations. *Journal of Geophysical Research: Atmospheres*, 111(D7), 2006.
- [2] Francois Chollet et al. Keras, 2015.
- [3] Ryosuke Masuda, Hironobu Iwabuchi, Konrad Sebastian Schmidt, Alessandro Damiani, and Rei Kudo. Retrieval of cloud optical thickness from sky-view camera images using a deep convolutional neural network based on three-dimensional radiative transfer. *Remote Sensing*, 11(17):1962, 2019.

- [4] Rintaro Okamura, Hironobu Iwabuchi, and K. Sebastian Schmidt. Feasibility study of multi-pixel retrieval of optical thickness and droplet effective radius of inhomogeneous clouds using deep learning. *Atmospheric Measurement Techniques*, 10(12), 2017.
- [5] Robert F Cahalan, William Ridgway, Warren J Wiscombe, Warren J, Thomas L Bell,, and Jack B Snider. The albedo of fractal stratocumulus clouds. *Journal of the Atmospheric Sciences*, 51(16):2434–2455, 1994.
- [6] Teruyuki Nakajima and Michael D. King. Determination of the optical thickness and effective particle radius of clouds from reflected solar radiation measurements: part i: theory. *Journal of Atmospheric Sciences*, 47(15):1878–1893, 1990.
- [7] Zhibo Zhang, Andrew S. Ackerman, Graham Feingold, Steven Platnick, Robert Pincus, and Huiwen Xue. Effects of cloud horizontal inhomogeneity and drizzle on remote sensing of cloud droplet effective radius: Case studies based on large-eddy simulations. *Journal of Geophysical Research: Atmospheres*, 117(D19), 2012.
- [8] Chamara Rajapakshe and Zhibo Zhang. Using polarimetric observations to detect and quantify the three-dimensional radiative transfer effects in passive satellite cloud property retrievals: Theoretical framework and feasibility study. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 46:106920, 2020.
- [9] Steven Platnick, Kerry G. Meyer, Michael D. King, Galina Wind, Nandana Amarasinghe, Benjamin Marchant, G. Thomas Arnold, Zhibo Zhang, Paul A. Hubanks, Robert E. Holz, Ping Yang, William L. Ridgway, and Jérôme Riedi. The modis cloud optical and microphysical products: Collection 6 updates and examples from terra and aqua. *IEEE Transactions on Geoscience and Remote Sensing*, 55(1):502–525, 2016.
- [10] Nan Ye, Farbod Roosta-Khorasani, and Tiangang Cui. *Optimization Methods for Inverse Problems*, pages 121–140. Springer International Publishing, Cham, 2019.