Exploring Neural Networks For Predicting Sentinel-C Backscatter Between Image Acquisitions

Zhongdi Wu
zhongdiw@smu.edu

Follow this and additional works at: https://scholar.smu.edu/engineering_compsci_etds

Part of the Other Computer Engineering Commons

Recommended Citation
Wu, Zhongdi, "Exploring Neural Networks For Predicting Sentinel-C Backscatter Between Image Acquisitions" (2021). Computer Science and Engineering Theses and Dissertations. 23.
https://scholar.smu.edu/engineering_compsci_etds/23

This Thesis is brought to you for free and open access by the Computer Science and Engineering at SMU Scholar. It has been accepted for inclusion in Computer Science and Engineering Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.
EXPLORING NEURAL NETWORKS FOR PREDICTING SENTINEL-C BACKSCATTER BETWEEN IMAGE ACQUISITIONS

Approved by:

______________________________
Eric Larson
Computer Science
Dissertation Committee Chairperson

______________________________
Co-chair Ginger Alford
Computer Science

______________________________
Michael Hahsler
Computer Science
EXPLORING NEURAL NETWORKS FOR PREDICTING SENTINEL-C BACKSCATTER BETWEEN IMAGE ACQUISITIONS

A Dissertation Presented to the Graduate Faculty of the
Lyle School of Engineering
Southern Methodist University

in
Partial Fulfillment of the Requirements
for the degree of
Master of Science
with a
Major in Computer Science
by

Zhongdi Wu
B.S., Electric Engineering, North China Electric Power University

December 18, 2021
Measuring moisture dynamics in soil and overlying vegetation is key to understanding ecosystem and agricultural dynamics in many contexts. For a variety of applications, moisture information is demanded at high temporal frequency and fine spatial resolution over large areas. Sentinel-1 C-band radar backscatter satellite images provide a repeating sequence of fine-resolution (10-m) observations that can be used to infer soil and vegetation moisture, but the 12-day interval between satellite observations is infrequent relative to the sensed moisture dynamics. Machine learning approaches have been used to predict soil moisture at higher spatial resolutions than the original satellite images, but little effort has been made to increase the temporal resolution of the images. This study extends machine learning approaches to infer fine-resolution backscatter between observations relying on auxiliary data observations, including elevation and daily gridded weather. Several variations of Multimodal Neural Network architectures, problem setup, and training methods are explored for a predominantly rural area in southwest Oklahoma near the transition between humid subtropical and semiarid climates. This study find that the UNET architecture produced the most accurate and robust estimated backscatter patterns, with superior prediction compared to a prior observation baseline in nearly all cases investigated when geography was included in the training data. This superior performance also generalized to nearby areas when training data for a given geography was not available, where 86% of predictions performed superior compared to a prior observation baseline.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................. vii
LIST OF TABLES ................................................................. ix
ACKNOWLEDGMENTS ............................................................. x

CHAPTER

1. Introduction ................................................................. 1
2. Deep Learning Background ............................................. 5
   2.1. Objective Function Optimization ................................. 5
   2.2. Convolutional Layers ................................................ 6
   2.3. UNET Fully Convolutional Network .............................. 8
   2.4. Multi-task Neural Network Learning ............................ 9
3. Data Acquisition and Dataset Generation .......................... 10
   3.1. Data Acquisition ...................................................... 10
      3.1.1. Geographic Area of Study .................................... 10
      3.1.1.1. Elevation and Weather Data ............................ 10
      3.1.2. Sentinel-1 ........................................................ 10
      3.1.2.1. Speckle filter based on median values ............... 12
   3.2. Problem Setup and Dataset Generation ......................... 13
      3.2.1. Experiment I ...................................................... 15
      3.2.1.1. Data Augmentation for Experiment I ................... 16
      3.2.1.2. Initial Analysis of Experiment I with Clustering Method . 16
      3.2.2. Experiment II ...................................................... 19
      3.2.2.1. Data Augmentation for Experiment II ................... 19
4. Neural Network Architecture and Training ........................ 22
4.2. Architectures for Experiment I: Multi-Modal CNN 24
4.3. Architectures for Experiment II 26
  4.3.1. UNET 26
  4.3.2. Multi-Task 29
4.4. Experiments 30
  4.4.1. Packages and Computation Details 30
  4.4.2. Experiment details 31
5. Results and Analysis 32
  5.1. Common evaluation metric: Improvement Rate 33
  5.2. Results Summary 33
    5.2.1. Experiment I 33
    5.2.2. Experiment II 35
      5.2.2.1. Improvement Rate for UNET-10 37
  5.3. Additional Analyses 37
    5.3.1. Worst Scenario Visual Inspection 40
    5.3.2. Artifact Analysis in Test Set 43
    5.3.3. Crucial Weather Information Analysis 45
6. Conclusion and Future work 50
APPENDIX A 51
  6.1. Examples of well prediction on test set 51
  6.2. Examples of poor prediction on validation set 55
  6.3. Examples of poor prediction on test set 57
BIBLIOGRAPHY 65
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1. The study area of initial focus(Experiment I) and latter extension(Experiment II)</td>
<td>11</td>
</tr>
<tr>
<td>3.2. Example of special VV and VH values identified by elbows in upper tails of cumulative distribution functions, used to define threshold values to apply median filter</td>
<td>13</td>
</tr>
<tr>
<td>3.3. An example map before and after the de-speckling median filter.</td>
<td>14</td>
</tr>
<tr>
<td>3.4. Example of prediction function, predicting a current Sentinel-1 image (May 8) based on a previous map (May 3) and precipitation and temperature information prior to both May 3 and May 8.</td>
<td>14</td>
</tr>
<tr>
<td>3.5. Example of data split into training and validation (blue grid region), and test sets (red grid region).</td>
<td>21</td>
</tr>
<tr>
<td>4.1. Network design based on the ResNet Block architecture, used as network unit in more complex network architectures.</td>
<td>23</td>
</tr>
<tr>
<td>4.2. Example of the architecture of multi-modal neural network on 15-cluster intermediate scale</td>
<td>25</td>
</tr>
<tr>
<td>4.3. Example of the architecture of UNET on 10 meter resolution</td>
<td>28</td>
</tr>
<tr>
<td>4.4. The architecture of Multi-Task Network modification based on UNET on decoder</td>
<td>30</td>
</tr>
<tr>
<td>5.1. the clustering diagram for 15 clusters</td>
<td>34</td>
</tr>
<tr>
<td>5.2. the water body captured by clustering marked over the historical mean</td>
<td>34</td>
</tr>
</tbody>
</table>
5.3. Relative increase in MSE for the predicted image compared to the base image for UNET-10 validation and test sets. Each circle is the center of an 8 by 8 grid of 3-km-square weather grid pixels. Urban area density and mean terrain slope within each weather grid pixel are overlain, with urban area density indicated by yellow box size and slope indicated by shading. The urban areas in the northeast and southwest are Oklahoma City and Lawton, respectively. (figure data prepared by Zhongdi Wu, figure generated by Dr. Stuart Stothoff from SwRI) 38

5.4. the base, prediction, target, residual plot of the worst scenario in validation set with a MSE of $198.993 \times 10^{-4}$ 41

5.5. the base, prediction, target, residual plot of a poorer scenario in test set with a MSE of $54.694 \times 10^{-4}$ 42

5.6. the base, prediction, target, residual plot of a poorer scenario in test set with a MSE of $33.013 \times 10^{-4}$ 42

5.7. the base, prediction, target, residual plot of a poorer scenario in test set with a MSE of $21.505 \times 10^{-4}$ 44

5.8. the base, prediction, target, residual plot of a poorer scenario in test set with a MSE of $1.359 \times 10^{-4}$ 45

5.9. L2 norm and mean of gradient of precipitation-Sum of VV VH relation 47

5.10. L2 norm and mean of gradient of min temperature-Sum of VV VH relation 48

5.11. L2 norm and mean of gradient of max temperature-Sum of VV VH relation 48
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1. Resolution and Training details of Deep Neural Networks</td>
<td>31</td>
</tr>
<tr>
<td>5.1. MSE of Deep Neural Networks times $10^{-4} \pm 95%$ confidence interval</td>
<td>35</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

The author would like to thank Southwest Research Institute and Southern Methodist University for funding this study, Dr. Stuart Stohoff and Dr. Osvaldo Pensado from SwRI are also contributors to this work, SMU Computer Science students Phuoc Dinh Le and Jacob Barcelona for helping collecting and preprocessing data efficiently at the early stage of this study, and Southern Methodist University High Performance Center for providing access to the ManeFrame II for the computation and training of models. The body of this work is arranged as a paper currently under review by IEEE Transactions on Geoscience and Remote Sensing.
Chapter 1
Introduction

Soil moisture plays a crucial role in the fields of hydrology, meteorology, and agriculture. Sensing and predicting soil moisture are key for applications such as drought monitoring, flood prediction, and crop productivity monitoring. It is common practice to develop water-balance models based upon sensed soil moisture data, either directly from soil moisture sensors or indirectly from remote sensing. Regional-scale water balance models typically rely on a remote sensing practice with observations from satellites operating in the L band (1.20-1.41 GHz) and complemented with passive sensing with a radiometer to measure brightness temperature, such as SMAP, SMOS, and AMSR2 satellite data [1–3]. These satellites and sensing instruments can provide soil moisture related remote sensing information at ∼40 km spatial resolution (or better) and 2-3 days temporal resolution. That is, for a given 40 km by 40 km area, these satellites can provide coarse information about the relative soil moisture as often as two times per week. SMAP also had an active L-band sensor to measure backscatter with finer resolution (1 to 3 km), but unfortunately this sensor failed soon after mission inception.

The way that these satellites function is by the principle of having a radiometer and a synthetic aperture radar (SAR) instruments. The instruments operate at L-band(1.20-1.41GHz), that measure emission and backscatter.

Established water balance models provide higher spatial and temporal soil moisture scales by merging satellite observations from different bands. For example, using CYGNSS with an assumed linear relationship or Machine Learning-based approach has been used to estimate both in situ and remote sensed soil moisture [4]. However, factors including vegetation, surface temperature, and additional surface characteristics have an effect on interpretations of satellite observations with respect to soil moisture that make them less reliable [5]. Thus, water balance models that use coarser backscatter require a strong understanding of rela-
tionships between surface soil moisture and remote sensing data as well as heavy feature engineering in retrieving soil moisture [6–8]. Model approaches for combining satellite observations with water balance models are usually traditional machine learning models or statistical models.

Sentinel-1 carries a C-band synthetic aperture radar instrument operating at a centre frequency of 5.405GHz. Satellite data products provided include radar backscatter data with up to 5-by-20meter finest spacial resolution in two orientations, vertical emit/vertical return(VV) and vertical emit/horizontal return(VH). The VV and VH backscatter images can be correlated to the amount of moisture in shallow soil and above-ground vegetation.

In most United States locations, Sentinel-1 acquisitions are limited to a single satellite (i.e., a 12-day repeat cycle), although acquisition tracks overlap and some tracks are imaged by both satellites. A 12-day cycle is infrequent compared to the time scale of moisture dynamics in shallow soil, limiting the applicability in water balance models. If the acquisition frequency can be enhanced, then inferred moisture information becomes more relevant to applications such as agriculture (e.g., monitoring field health and irrigation needs), emergency response, and drought monitoring. We seek to artificially increase the temporal resolution of fine-scale backscatter images using machine learning combined with additional information (e.g., satellite weather products), taking advantage of the border regions in overlapping satellite tracks to train the deep learning model at different intervals. The simulated backscatter images could be used for water balance models, although we do not directly estimate soil or plant moisture content. With this end use in mind, we are most interested in representing backscatter dynamics in areas with significant soil and plant cover, which tend to have relatively small backscatter that is sensitive to moisture changes, but are also interested in identifying potential sources of error from features with large backscatter. By predicting the remotely sensed images, rather than derivative products, we expect to more directly isolate errors in remote sensing from errors in translating backscatter into moisture predictions, which can be used to assess the reliability of derived downstream products across a landscape. For example, derived moisture estimates would be more reliable in areas with high image predictability than in areas with low image predictability, all else being equal.

To investigate the merits of this technique, we employ Fully Convolutional Networks
(FCNs). Over the last decade, Convolutional Neural Network (CNN) techniques have been applied to a variety of domains with great success. Similarly, FCNs are neural networks that use convolutional filters throughout the network, such that the output is also another image. FCNs have shown great success in applications such as image segmentation [9], instance segmentation [10], and image generation [11].

Use of deep learning techniques with Sentinel-1 data to produce temporally enhanced Sentinel-1 artificial backscatter maps has received relatively little attention. Established CNN techniques have developed largely as a solution to a variety of problems related to very large data sets of images significantly different from Sentinel-1 backscatter images. Over the past few years, CNN techniques have been applied to several SAR-based tasks and reached reasonable achievements [12]. Some researchers have performed multi-class classification based on Sentinel-1 and Sentinel-2 images with CNNs [13–17] and Recurrent Neural Networks (RNNs) [18]. That is, these researchers used neural networks that were trained to regress the surface soil moisture from the backscatter images. Other researchers have attempted to increase the accuracy of surface soil moisture prediction by merging Sentinel-1 backscatter data with other data sources [19–21]. A Random Forest model that merged several data sources reached an average of 0.020 cm$^3$/cm$^3$ ubRMSE [22], compared with 0.050 cm$^3$/cm$^3$ from a generalized regression neural network that only includes SMAP [23]. Pre-trained models trained with natural images have been applied to SAR images to accomplish classification tasks [24]. Some researchers have combined Sentinel-1 backscatter images from different satellites to do change detection [25]. The Sentinel-1 backscatter image-based applications above indicate that (1) temporally high frequencies of Sentinel-1 backscatter images are in high demand and (2) deep CNNs are suitable for a number of Sentinel-1 backscatter image-based tasks. These observations motivate the proposed approach.

This work differs from these previous studies and approaches because we seek to produce artificial Sentinel-1 backscatter maps at more frequent time internals by using deep CNN techniques and secondary data sources. That is, we propose using FCNs to synthesize backscatter images, therefore artificially boosting the temporal resolution of backscatter satellite coverage. In pursuing this research, however, several factors need to be considered in the development of CNN solutions. This includes (1) data set size, (2) data noise, (3) high
variability of measured radar intensity, (4) secondary satellite data gathered in maps of variable spatial resolution, and (5) regional characteristics. A goal of this study is to assess the feasibility of CNNs to enhance the temporal and spatial resolution of Sentinel-1 data outputs and to develop practical insights. This research is an early step towards providing high resolution input for soil moisture models at more frequent intervals than currently exist.

In this study, a reasonable problem setup to produce artificial Sentinel-1 backscatter images merging secondary data sources was established. Several convolutional deep neural networks were trained and compared. More importantly, an analysis focused on UNET generating artificial Sentinel-1 backscatter at 10 meter resolution was carried out to identify failure and outlier cases in validation and test sets for further improvement.

We outline our contributions in this work as follows:

1. We investigate the feasibility of using fully convolutional neural networks to generate synthetic backscatter images.

2. We investigate the use of multi-task and multi-resolution outputs, showing that there is little advantage over the single resolution FCN model.

3. We provide an analysis of the failure cases for the model and categorize the mechanisms for failure.

4. We provide an analysis of the most important side information for synthesizing backscatter images.

We organize the remainder of the thesis as follows: Chapter two discusses the related deep learning background; Chapter three introduces the data acquisition and preprocessing pipeline; Chapter three introduces the concepts related to problem setup and how data was arranged into pairs of examples; Chapter five discussed the details of the neural network architectures and how the experiments were conducted; Chapter six provides a in depth discussion of the performance from a variety of angles.
Chapter 2
Deep Learning Background

This chapter discusses the basics of using neural networks in our application of synthetic backscatter image generation. Specifically, we introduce the concepts of back propagation, neural network regression, and multi-modality in the context of backscatter image generation. The actual architecture of the network employed is introduced in Chapter 4—this chapter focuses on basic neural network optimization.

2.1 Objective Function Optimization

One objective of this research was to design a function to produce temporally enhanced Sentinel-1 backscatter images. A natural input to this function is the most recent previous backscatter image of the same geographic area and additional secondary data sources to inform the network how to alter the input backscatter and produce a new backscatter image [26]. Which leads to an idea of developing a feed forward artificial neural network function where additional information is given in the form of a weather grid for a given area. In this study the secondary weather grid is organized by 7-day precipitation, min temperature and max temperature for the following reasons: Rainfall is a crucial factor on the amount of water in vegetation and soil. Temperature is a key factor of Water evaporation. Temperature can also encodes the season information of a similar geographic area.

If we denote \( \hat{B} \) as the predicted backscatter image, \( B' \) as the previous Sentinel-1 backscatter at the same location, \( S_b \) as the 7-day daily weather grid before previous Sentinel-1 observation, and \( S_p \) is the 7-day daily weather grid before Sentinel-1 backscatter that is being predicted, we can define the neural network function \( F \) as follows:

\[
\hat{B} = F(S_b, S_p, B')
\]
The concept of using $S_b$, $S_p$, and $B'$ as separate inputs is commonly known as multi-modal machine learning. Each input connects to a separate branch in the neural network and the network learns how to combine their information appropriately.

Traditionally, the training of a neural network is accomplished with the back-propagation algorithm [27, 28], where the errors from a large number of training examples are used to optimize the weights of a neural network in order for its predictions to more closely resemble the desired outputs. The method employs the use of gradient calculations according to an objective function. The gradient at the final layer can be used to calculate the sensitivity of the gradient in the previous layer. This sensitivity calculation is applied recursively throughout the neural network until all weights have been updated to reduce errors.

In our application, the training labels are continuous valued backscatter images. The nature of predicting a set of continuous values leads to the selection of Mean Squared Error (MSE) as a natural loss function, which in our case can be written as:

$$E = \frac{1}{h \cdot w \cdot c} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \sum_{k=0}^{c-1} (B_{(i,j,k)} - \hat{B}_{(i,j,k)})^2$$

(2.2)

Where $h$, $w$, $c$ are the height, width, channel size of the Sentinel-1 backscatter map, $B$ is the actual backscatter map, $\hat{B}$ is the predicted backscatter image, and $(i, j, k)$ are the indices of pixel in the maps. Thus, our problem can be thought of as a regression problem where we update the weights of the network to better approximate the actual backscatter images.

The neural networks were trained with an adaptive momentum (Adam) optimizer, which has an adaptive learning rate [29]. This optimization algorithm seeks to apply normalized gradient updates using the first and second moments of the gradients over batches of training data. Intuitively, this method serves to accelerate training steps sizes for weights that are changing slowly, while decelerating step sizes for weights that are shifting too quickly.

2.2 Convolutional Layers

Classic artificial neural network architectures used many fully connected layers, whereby every input was multiplied by a separate optimized weight. For images, however, it has become common practice to use filters that convolve with the input features, rather than
training a separate set of weights for each pixel. In this method, the weights of the filter are randomly initialized and then updated to reduce the value of the objective function. These optimized filters are commonly referred to as convolutional layers [30]. The convolution with filter \( h \) working on an activation \( f \) can be written as:

\[
A_{out}[i, j] = (f * h)[i, j] = \sum_{k=-(\frac{n_w-1}{2})}^{\frac{n_w-1}{2}} \sum_{l=-(\frac{n_h-1}{2})}^{\frac{n_h-1}{2}} h[k, l] f[i - k, j - l] \tag{2.3}
\]

where \( n_w \) and \( n_h \) are the width and height of the convolution filter. They are assumed odd integer numbers. \( i \) and \( j \) are the index of a particular value on output activation. \( A_{out} \) is the output activation. The convolution filter \( h \) has a center at coordinate \( [0, 0] \).

Convolutional layers were selected based on the type of data that the function is working with: grid-like shift-free signals, like images. CNNs can learn representations from grid-like data by sliding a learned kernel with the same weights on the grid, extracting different features from the grid [31]. Additional convolutional kernels can then be used to process the output of the previous kernels. This process can be repeated many times, creating a deep convolutional representation of the input grid. However, the gradient of the layers can be unstable when there are too many layers, and optimization of the network is increasingly difficult or time-consuming [32]. This is typically referred to as the “vanishing gradient problem” or “unstable gradient problem.”

One solution to the vanishing gradient problem is to provide multiple back propagation pathways for the gradient to be calculated from. This is the mechanism of operation for the popular Residual Network (or ResNet) model [33]. In this network, two parallel neural network paths are created and added together before feeding to the next convolutional layer. The mechanism of Residual connection can be written as:

\[
A_{out} = F_{activation}(x, \{W_i\}) + W_S x \tag{2.4}
\]

where \( A \) is the output activation of the sub block, \( F_{activation} \) is the nonlinear activation function, \( x \) is the input activation, \( \{W_i\} \) is a sequence of multiple convolution layers described in 2.3, \( W_s \) is the linear projection added to the output of the sequence of convolution layers.
and activation function. The added terms are guaranteed to have same dimensions as the output of the sequence operation.

In order to train a deeper network, we employ a ResNet block architecture to make deep neural networks easier to optimize with skip connections and allow a deeper network design [33]. Skip connections are similar to residual connections, but are employed across multiple layers. In each ResNet block, Batch Normalization was applied between convolutions and nonlinear functions, because the ability to normalize layer inputs allows larger learning rates and less careful initialization [34]. Intuitively, batch normalization optimizes parameters that normalize the mean and dynamic range of the features at the input of each layer. This normalization helps reduce gradient instability.

### 2.3 UNET Fully Convolutional Network

One popular FCN approach restricts training to filters only; that is, no fully connected layers are employed in the network. One such network is known as UNET, which first gained wide acceptance as a segmentation technique from its success in creating clearly defined regions in medical images for biomedical applications [35] and has since been more widely adopted for other uses.

UNET is an Auto-Encoder (AE) with skip connections. For a pure convolutional network (convolution layers that do not insert or remove rows/columns in the output of the convolutional output, sometimes referred to as up sample or down sample operations), the only way to extend the receptive field to process a large object is to increase the size of convolution kernel. However, with the up sample and down sample architecture of AE, it is possible to increase the size of receptive field by doing convolution multiple times, which reduces the computational complexity with a hierarchical structure [36].

However, down sample process of AE reduces the resolution information (high frequency component), and the up sample process can have difficulty re-introducing high frequency components. Thus, the output of AE can have blurry edges and bring strong noise for high resolution images.

The UNET architecture, in this sense, reintroduces high frequency information by saving (in memory) the activation with high frequency information during the down sample
process and concatenates this saved activation with the newly computed activation in the up sampling operation of the same size, which reduces distortion enormously. Essentially, the activations computed during the down sampling are saved and then concatenated with activations in the up sampling branch. In this way high resolution information is less likely to be lost.

The UNET was selected as the basic architecture to be combined with ResNet blocks in the neural network design. We independently applied the UNET algorithm on several grid resolutions to explore performance benefits and constraints.

2.4 Multi-task Neural Network Learning

Multi-Task Learning (MTL) is a technique to design and train a network to produce multiple outputs or optimize on multiple loss functions. MTL training is known to produce robust neural networks well adept at generalizing [37]. We used MTL to simultaneously solve for all of the different UNET cases, in order to explore potential performance benefits from simultaneously considering several resolutions during a single training exercise.
Chapter 3
Data Acquisition and Dataset Generation

In this chapter, we describe the data collection from different data source and two ways of machine learning problem setup that leads two separate experiments.

3.1 Data Acquisition

3.1.1 Geographic Area of Study

The study area is shown in Figure 3.1. This study initially focused on a 34 by 21 km region approximately 80 km southwest of Oklahoma City, latter extended to a larger region (latitude 33.8 to 35.8° N, longitude 97 to 99° W) with a mix of agriculture, rangeland, urban areas, forested streams, surface water bodies, highways, railroads, and drill pads. This area was ideal for our dataset due to the familiarity our researchers had with the area and the variety of different wildlife and natural formations, which also allows contrast of performance on different land types.

3.1.1.1 Elevation and Weather Data

Elevation data from the National Elevation Dataset [38] at a 10-m resolution were collected from the Google Earth Engine (GEE). The elevation data were used as reference data to define common coordinates. Additionally, PRISM daily weather data [39] (precipitation and minimum / maximum temperatures) were obtained from the GEE as 3-km resolution tiles, Data assets have a record time of 12:00 PM UTC (6:00 AM Central Standard Time).

3.1.2 Sentinel-1

Sentinel-1 data were collected from European Space Agency (ESA)\(^1\), provided with a 10-m resolution and the precise satellite capture time. Sentinel-1 data are provided in approximately square tiles over orbital tracks observed by two mission satellites, Sentinel-1A

\(^1\)ASF DAAC 2020, containing modified Copernicus Sentinel data 2015 to 2020, processed by ESA.
Figure 3.1: The study area of initial focus (Experiment I) and latter extension (Experiment II)
and the newer Sentinel-1B, that follow the same orbit six days apart. Each of two ascending tracks (relative orbits 34 and 107) has a tile that covers part of the region of interest. The overlap zone represents the most distal part of the image for one track and the most proximal for the other; the different look angle produces slightly different backscatter signals for the two tracks in the overlap zone. Most of the area covered by Sentinel-1 images in the United States is limited to a 12-day repeat. Sentinel-1A images both tracks and Sentinel-1B images relative track 107; in the overlap zone, these combinations result in images with temporal offsets of 1, 5, 6, 7, 11, and 12 days after Sentinel-1B became operational. The tracks have time stamps of approximately 0:30 AM UTC (6:30 PM Central Standard Time), which is approximately 12 hours out of sync with the daily weather data.

The ESA provides the SeNtinel Application Platform (SNAP) software tool to post-process satellite raw data outputs by performing functions such as alignment with reference coordinates and de-speckling. The SNAP tool was executed to return Sentinel-1 backscatter data as two channels, vertical-send/vertical-receive (VV) and vertical-send/horizontal-receive (VH). The transformed two-channel data were processed with the terrain-correction function to output information with square pixels at an average resolution of 10 meters. Spikes and noise were removed using the Lee-sigma speckle filter provided by the SNAP tool. The SNAP collocation tool was applied with bilinear interpolation, using coordinates of the elevation map as the master reference, so that the longitude and latitude coordinates of the transformed backscatter data were aligned with the reference elevation map.

3.1.2.1 Speckle filter based on median values

A large number of speckles on the processed backscatter radar data commonly remain after the SNAP Lee-sigma based speckle filter; these are not necessarily correlated to moisture content but caused by other reasons including detection artifacts. The cumulative distribution functions (CDFs) of both VV and VH channels were employed to identify threshold VV and VH intensity values that could be used for a median filter. CDFs prepared with pixel values from 100 randomly sampled Sentinel-1 backscatter maps are displayed in Figure 3.2. The elbow corners labeled with blue dots in Figure 3.2 were selected as the threshold values. In applying the median filter, pixel values exceeding the threshold values were replaced by
Figure 3.2: Example of special VV and VH values identified by elbows in upper tails of cumulative distribution functions, used to define threshold values to apply median filter the median value of neighbor pixels (neighbor diameter of 13 pixels). Figure 3.3 displays an example of the smoothing from the filter. From Figure 3.2, approximately 38 percent of the VV intensities and 30 percent of the VH intensities were changed by the action of the median filter.

3.2 Problem Setup and Dataset Generation

To train and evaluate the proposed approach for augmenting the temporal resolution of Sentinel-1 backscatter images, a prediction function was designed taking as input a Sentinel-1 image at one time and location of interest to output a Sentinel-1 image at another time for the same location. For machine learning training purposes, the ground truth was defined as the Sentinel-1 image at the new time of interest. The prediction function was designed to take additional information, such as weather conditions (e.g., temperature and precipitation) leading up to each image, for the machine learning algorithm to account for moisture dynamics. Needed input/output datasets were constructed from numerous databases. Figure 3.2.1.2 displays a visual example of the prediction function inputs and outputs.
Figure 3.3: An example map before and after the de-speckling median filter.

Figure 3.4: Example of prediction function, predicting a current Sentinel-1 image (May 8) based on a previous map (May 3) and precipitation and temperature information prior to both May 3 and May 8.
Because of the incredible resolution difference between Sentinel-1 images and available secondary source grids, this study conducted two experiments:

1. The first experiment attempts to aggregate the original Sentinel-1 images to an intermediate representation (i.e., a resolution configured by clusters of spatial regions) and performs the prediction on the mean values of each clusters (assume zero deviation) on a particular rectangular region (hereafter referred to as Experiment I). This experiment is primarily used for analysis with clustering methods combined with a neural network.

2. The second experiment attempts to predict Sentinel-1 images directly with data pairs sampled on a much larger region (hereafter referred to as Experiment II). This experiment is primarily used for investigation involving deep learning methods that produce an entire image.

In terms of the volume of data available, Sentinel-1 mission was launched in 2015 with a 12-day frequency, or 6-day frequency for regions of with two-track overlap. This means the total number of available Sentinel-1 image on a particular region is modest (less than 500 in total). To mitigate the relatively few images in the database, different data augmentation methods were applied in both experiments.

3.2.1 Experiment I

Overall, experiment I focused on a 34km by 21km static region and all prediction happened on an intermediate representation carried out by a K-means based clustering algorithm, explained in detail below. In this experiment, the analysis is separated into two different steps: (1) clustering similar regions in a given geographic area, and (2) predicting the backscatter for each region with a CNN. In this way, the CNN dimensionality is reduced because it only needs to predict backscatter values for each region. One Sentinel-1 image was selected as the base image and another latter (in time) image was selected as the ground truth to train the CNN function. The seven days preceding each image collection were then used to provide additional weather and temperature data. That is, the 7-day daily weather grids before the base Sentinel-1 image and prediction Sentinel-1 image were provided for the CNN function. We hypothesized that giving the neural network the seven preceding days
for both images would help the network better match rainfall and temperature patterns, thereby reducing the residual between base and prediction Sentinel-1 images.

3.2.1.1 Data Augmentation for Experiment I

Because the Sentinel-1 mission was launched in 2015, the total number of available Sentinel-1 images of a particular region is limited. This is exacerbated by the pairing method in Experiment I that pairs two Sentinel-1 image with the closest available one, which restricts the number of available pairs. To mitigate this during the dataset generation process, the pairing between two Sentinel-1 images was expanded. That is, one image was paired with multiple images as long as the paired images were within 15 days of one another. This greatly expanded the number of training examples by expanding the number of pairs.

3.2.1.2 Initial Analysis of Experiment I with Clustering Method

The resolution of a precipitation pixel is 3km, when that of a sentinel-1 pixel is 10m. That is, given the precipitation and sentinel-1 image of a same region, the width and height of the sentinel-1 image is 300 times larger. The resolution difference was considered hard for the information to combine at the early stage of the study when Experiment I was setup. Thus, the idea was to somehow reduce the dimension of sentinel-1 image to be relatively smaller to work with. Thus, we first needed to apply a clustering algorithm that grouped similar backscatter regions. After similar backscatter regions was determined, we predicted the mean values of the pixel values assuming that the pixels belong to a same backscatter region would have similar deviation in backscatters of different date. This clustering was carried out on one geographic region. A grid was created of the region and each element in the grid was clustered using K-Means. The historical mean and standard deviation of both VV and VH channels of Sentinel-1 and North-South, East-West gradient calculated from Elevation were selected as the data to do clustering on the region. Data were normalized to zero mean before fed to K-means clustering algorithm.

The assumed zero deviation representation can be written as follow: Let $B$ represent the set of Sentinel-1 images that will be clustered and let $B_i$ represent a Sentinel-1 image on a particular day. It is intuitive the view the $B_i$ as a single channel of $B$. The objective is then to cluster across channels such that $B$ is divided into spatial regions. If clustering
is conducted on $B$, $B^k_i$ represents all the pixels that belong to the $k^{th}$ cluster on the $i^{th}$ example. Where $k$ is consistent across all channels and $K$ is the total number of clusters in $B$. Thus, we can formulate a method by assuming that backscatter values in a cluster have similar mean values:

$$\hat{B}^k_i = \mu^k_i - \mu^k_j + B^k_j \quad (3.1)$$

where $\mu^k_j$ is the mean of $k$ cluster on the $j^{th}$ example. $B^k_j$ is the Sentinel-1 image value from a previous date of $j$, which is available, $\hat{B}^k_i$ is the Sentinel-1 image being predicted.

Furthermore, We can define a prediction for the mean of each cluster as:

$$\hat{\mu}^k_i = F(S_b, S_p, \mu_j)^k \quad (3.2)$$

$$e_{NN}^k = \mu^k_i - \hat{\mu}^k_i \quad (3.3)$$

Where $F$ is the prediction function and $S_b$, $S_p$, $\mu_j$ are the 7-day daily weather grid before base, 7-day daily weather grid before prediction and the mean values of all clusters from base sentinel-1 image.

MSE was selected as loss function for Neural Network training. Relation was found between the variance across clusters, the Neural Network prediction loss, and the MSE of the whole reproducing method. We define the objective function, $J$, as:

$$e^{k2} = \frac{1}{N^k} \sum (B^k_i - \hat{B}^k_i)^2 \quad (3.4)$$

$$J = E\left\{\frac{1}{K} \sum_{k=0}^{K-1} e^{k2}\right\} \quad (3.5)$$

where $N^k$ is the total number of pixels in the $k^{th}$ cluster, and we try to minimize $J$ for better performance.

And we separate deviation with:

$$B^k_i = \mu^k_i + \hat{B}^k_i \quad (3.6)$$
plug into the expression of $e^{k^2}$,

$$e^{k^2} = \frac{1}{N^k} \sum \left( \tilde{B}_i^k - \tilde{B}_j^k + e_{NN}^k \right)^2$$  \hspace{1cm} \text{(3.7)}$$

where both $\tilde{B}_i^k$ and $\tilde{B}_j^k$ are set of values and $e_{NN}^k$ is scalar, which means it gets add to the sum $N^k$ times. Let $\sigma_{(i,j)}^k$ be the deviation of $\tilde{B}_i^k - \tilde{B}_j^k$, $\sigma_i^k$ and $\sigma_j^k$ be the deviation of $\tilde{B}_i^k$ and $\tilde{B}_j^k$. It is hard to measure $\sigma_{(i,j)}^k$ because we need to traverse all the history data and measure the per-grid region deviation from the difference of any two Sentinel-1 images, however, it can be bounded by:

$$0 \leq \sigma_{(i,j)}^{2k} \leq \sigma_i^{2k} + \sigma_j^{2k}$$ \hspace{1cm} \text{(3.8)}$$

As $e_{NN}^k$ is the error from the prediction on mean of the machine learning algorithm, $\sigma_{(i,j)}^k$ is the deviation of difference of two Sentinel-1 images, we assume they are likely to be uncorrelated, Thus:

$$J = E\{ \frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{N^k} \sum (\tilde{B}_i^k - \tilde{B}_j^k + e_{NN}^k)^2 \}$$ \hspace{1cm} \text{(3.9)}$$

$$J = E\{ \frac{1}{K} \sum_{k=0}^{K-1} e_{NN}^{2k} \} + E\{ \frac{1}{K} \sum_{k=0}^{K-1} \sigma_{(i,j)}^{2k} \}$$ \hspace{1cm} \text{(3.10)}$$

Then, the loss of the prediction can be bounded by:

$$E\{ \frac{1}{K} \sum_{k=0}^{K-1} e_{NN}^{2k} \} \leq J \leq E\{ \frac{1}{K} \sum_{k=0}^{K-1} e_{NN}^{2k} \} + E\{ \frac{1}{K} \sum_{k=0}^{K-1} \sigma_i^{2k} \} + E\{ \frac{1}{K} \sum_{k=0}^{K-1} \sigma_j^{2k} \}$$ \hspace{1cm} \text{(3.11)}$$

From here, the inequality indicates that the performance of final prediction of the method can be bounded by the sum of the MSE of the prediction function and the expectation of variance across each cluster. Because of the nature that $B_i^k$ and $B_j^k$ are Sentinel-1 images in a same set, the last two term in the inequality essentially represents the same quantity.

Thus, the error for the CNN can be bounded. Figure gives an overview of the process for the neural network prediction. For each cluster, the neural network predicts the residual between the given cluster mean and the actual mean in the observed backscatter image.
Further explanation of the architecture used is discussed in chapter 4.

3.2.2 Experiment II

Given the limited prediction ability from the first experiment, we now turn our attention to Experiment II, which directly predicts Sentinel-1 images with more complex and deep Convolutional Neural Networks. That is, we abandon the idea of initially clustering the spatial regions and attempt to directly predict the backscatter images. While the results may improve, this method is not without limitations. For instance, the training time for this CNN, because it is many more parameters, is drastically increased. However, this approach may have additional advantages, such as not needing calibration data from a given region to initiate clustering. We explore these challenges and opportunities in the remainder of this thesis. Further explanation of the architecture used is discussed in chapter 4.

3.2.2.1 Data Augmentation for Experiment II

Differently than Experiment I, we no longer needed to cluster grids of regions at low spatial resolution. This allowed our models to use higher spatial resolution backscatter image pairs. This also presented a challenge as additional alignment of these higher resolution images was also needed.

Moreover, Deep Neural Networks need large datasets for training, especially to identify relations between Sentinel-1 images and secondary source data. Therefore, for Experiment II a data augmentation strategy was adopted and implemented in the following steps. First, the largest rectangular region oriented to the weather grid with a level of overlap of two Sentinel-1 tracks was established, shown in the yellow rectangle in Figure 3.5, based on information output by the SNAP tool, which tags pixels common to an elevation map (used as reference coordinate system) and to the Sentinel-1 image. Second, an approximated approach was developed to establish a region with guaranteed two-track overlap and with pixels aligning with the daily weather grid. This region is indicated by the interior of the green rectangle outline in Figure 3.5. This approach considered nearly constant directions of Sentinel-1 tracks, weather grid matching, and preservation of the height/width proportion of the yellow rectangle of Figure 3.5. Third, weather/Sentinel-1 pairs were sampled with a resolution of 8 by 8 daily weather grid pixels and with coordinates precisely aligning with Sentinel-1 image
coordinates within the green rectangle in Figure 3.5. A sampling pixel window was used to scan this rectangular region with a stride of 1 daily weather grid pixel, to maximize the generation of weather/Sentinel-1 data pairs. Fourth, the same pixel sampling process was conducted on every possible Sentinel-1 tracks observed neighboring in time, which extended out of the green rectangle that covers one particular pair. The universe set of result grids that were sampled in this way is indicated by the red and blue grids in Figure 3.5. Fourth, the resulting weather/Sentinel-1 data pairs were segmented into batches for the design of the prediction function, with 70 percent used for training, 15 percent for validation, and 15 percent for testing. The number of radar records available at different positions in the constructed database differ from location to location, in part because the two Sentinel-1 tracks have different numbers of orbits and in part because the acquisitions were not always consistently aligned. For the data segmentation as training, validation, and test sets, the regions with the fewest Sentinel-1 radar records were chosen as test set to ensure region diversity for robustness of the testing. The blue grid region in Figure 3.5 is the region with information used for training/validation of the CNN function; data from the red grid region was used for testing.

The dataset created for Experiment II is used for the remainder of this thesis. To the best of our knowledge, this dataset is of backscatter image pairs is of finer spatial resolution than any other backscatter dataset used for prediction. That is, this dataset represents the state-of-the-art in complexity for backscatter image prediction spatial resolution.
Figure 3.5: Example of data split into training and validation (blue grid region), and test sets (red grid region).
Chapter 4
Neural Network Architecture and Training

This chapter introduces the architecture elements employed in the convolutional networks for Experiments I and II. Elements are also motivated based on their historical image processing-based motivations. The networks employed are different based on their usage in Experiment I (a traditional CNN) and Experiment II (a fully convolutional network). We first describe some of the common architectural elements employed in each network, and then describe elements specific to each architecture separately.

4.1 Common Architecture Elements: Separable Convolution and ResNet Block

In both architectures, separable convolution was widely used in this study for the purpose of reducing the number of parameters as well as to accelerate the calculation [40]. Separable convolution in deep learning is different than the definition employed in signal processing. In the context of neural networks, separable convolution refers to the separation of trained filters for each input channel, followed by single dimensional convolutions that combine each convolved channel. This process was first used in the “X-ception” architecture [40] and was shown to drastically reduce the trainable parameters while retaining the expressiveness of the network. As prediction focused on continuous values, MSE was selected as the loss function in all architectures.

When building each network, we made use of a number of repeating elements throughout. We refer to these elements as a “sub-network sequence.” The sub network sequence using residual connections is displayed in Figure 4.1. The input to the network is an activation, presented as multi-channel 2-dimensional arrays. The separable convolution layers included 3 by 3 kernels and arrays with zero padding so that transformed output arrays were of the same dimension as the input image. The network design included batch normalization layers, rectified linear unit (ReLu) activation layers, and an addition layer (entry-to-entry
array addition operation) adding information of the first and last separable convolution outputs, so that the output of the network is perturbated multi-channel image similar to the input image. A detailed description is residual blocks and batch normalization is given in Chapter 2. The ResNet Block network in Figure 4.1 was used as a basic network unit used in the design of more complex network architectures. That is, we repeat this block numerous times in each architecture.

Figure 4.1: Network design based on the ResNet Block architecture, used as network unit in more complex network architectures.
4.2 Architectures for Experiment I: Multi-Modal CNN

Recall that the input to the architecture in Experiment I takes three branches: (1) the previous Sentinel Cluster Data, (2) the Daily Weather grid for the base image, and (3) the daily weather grid for seven days leading up to the prediction image. Because of assumption in Experiment I that the deviation within each cluster is zero. The reproducing of the fine resolution sentinel-1 backscatter relies on adding the residual between predicted mean and previously calculated mean. The neural network instead predicted the residual itself. The neural network processes each input (modality) separately before combining their representation within the neural network. This separation allows for features to be extracted from each information source independently before merging the information for additional feature extraction. Recall that the extracted features are learned by the convolutional network (rather than expertly designed), which is a major advantage for convolutional neural networks. Each daily weather grid modality consists of 7-days of information and each day has 3 channels of information which are minimum temperature in a day, maximum temperature in a day, and amount of precipitation. Thus, the daily weather grid input modality has $7 \times 3 = 21$ channels. Both the daily weather grid for base image and for prediction image were fed into the same architecture which shared parameters for the purpose that they were trying to encode same information. That is, the learned weights for each input modality are identical and optimized identically during back propagation. This weight tying approach, because each modality is extracting features from the same type of data, can be advantageous for reducing parameters. The architecture of the neural network used in Experiment I is shown in 4.2. Note that for Experiment I, the training time of the neural networks are significantly shorter than that in Experiment II. The shown parameters were the best performance model after hyper parameter search. In each input branch, the data is fed through two ResNet sequence blocks that keep the same channel size. After this, one ResNet block is used that doubles the channel size followed by another ResNet block that keeps the same channel size. This means that there are two activations with 42 channels of output. Then these two activations were concatenated and down sampled with 5 ResNet blocks which outputs the same number of channels as the target.
The cluster mean assumption was slightly extended in the implementation that we segmented the sentinel-1 map into fractions of daily weather grid pixels. That is, instead of compute the mean of a cluster by calculating the mean of all pixels in a whole map, pixels that align with a same daily weather grid pixel compute a separate mean. Which extended
the output dimension from only 30 (15 clusters with 2 channels) to 6 by 12 by 30 values.

4.3 Architectures for Experiment II

4.3.1 UNET

The history and detailed explanation of UNET can be found in Chapter 2. Recall that the UNET architecture has shown great capability (with its symmetric architecture) to reintroduce former activation in reconstructing an output image. UNET is primarily used in the medical community, but is ideally suited for our application because it allows the daily weather information to merge into the middle latent activation in the network of two symmetric processes. Thus, the extracted weather features can be used to alter the encoding process and therefore can contribute to the pattern for reconstruction of a new backscatter image.

For this network, the input base Sentinel-1 image, which has a resolution of 10 meters and a shape of (2384, 2376, 2) is perfectly aligned up with precipitation that has a resolution of 3km, which has a shape of (8, 8, 21). Figure 4.3 is an example of UNET architecture on 10 meter resolution. The Sentinel-1 base backscatter image is processed via 7 down sampling ResNet blocks. Each block down samples (spatially) the size of the activation with bilinear resizing that was half as large as previous one. However, the channels size is doubled. This increase in channel size acts as an incentive in the network to learn a more abstract set of features (in each channel) even in the presence of reduced spatial acuity. This is often referred to as the information distillation pipeline in CNNs [41].

Meanwhile, the modalities for both daily weather grid inputs for base and prediction went through three ResNet blocks that keep the same shape followed by three ResNet blocks that up sample their size to match the size of the activation of sentinel-1 backscatter (in the middle latent feature space) and increased their channel size to 32. These input branches are also kept identical (similar to the weight tying used in Experiment I) and they serve to extract relevant features from the weather data that can add value to the latent representation of the baseline input backscatter image. These three branch outputs are then concatenated together into a larger channel representation. The merged information then goes through a decoding process which has a symmetric architecture as the encoding process. The decoding
process consists of seven up sampling (spatially) ResNet blocks with bilinear resizing that resizes the output activation to match the size of the activation from previous symmetric step. This is a staple of the UNET architecture. Each upsampled activation in the decoder is merged with a previous downsampled activation from the encoder. Recall that this merging help to mitigate spatial information loss in the encoding process. 20 meter resolution and 40 meter resolution were also experimented to see if the Network can learn the pattern on different resolutions. For these two networks, the first or first two ResNet blocks do not down sample, which keep the same depth as the described architecture. In total, the UNET architecture employed in our research (on different resolutions) have 45 convolutional layers and 270,928 trainable parameters in total. We note that many architectures can be employed and investigated for this application. However, because of the training time (weeks) and number of layers used, only a minimal number of architectures and hyper parameters were reasonable to investigate.
Figure 4.3: Example of the architecture of UNET on 10 meter resolution
4.3.2 Multi-Task

Multi-Task Learning is known to make neural networks that generalize better by sharing parameters on different tasks. The idea of multi-task learning in neural networks is simple: use multiple output or multiple classification tasks. For each output we can update the model with back-propagation from error measured at each output branch [17]. The resolution variation in previous section was combined into a Multi-Task Learning in the following way: For the 10meter UNET, during the symmetric decoding process, the activation at the last layer was trying to approximate the 10meter resolution Sentinel-1 image. However, the activation that goes through one ResNet block less has the same shape as the 20meter resolution Sentinel-1 image. When the activation that goes through two blocks less has the same shape as the 40meter resolution Sentinel-1 image. Thus, two more separate blocks were designed for those activations to train on the Sentinel-1 images that were essentially on the corresponding resolution. Figure 4.4 indicates the modification of Multi-task learning based on UNET on 10meter resolution. Notice that architecture has three outputs, one for each predicted resolution of each backscatter image. We hypothesized that this use of multiple outputs could assist the network in predicting higher spatial resolution backscatter images—similar to a bootstrapping approach.
Figure 4.4: The architecture of Multi-Task Network modification based on UNET on decoder

4.4 Experiments

4.4.1 Packages and Computation Details

Neural Network architectures were developed using TensorFlow 2.2 and training processes were executed on SMU ManeFrame II. Computations were accelerated by one NVIDIA P100 GPU accelerator which contains 3584 CUDA cores and 16 GB CoWoS HBM2 memory.
4.4.2 Experiment details

We describe the details of our optimization runs so that other might be able to reproduce the results presented in the next chapter. Due to the edge of data augmentation, we were able to train models with a relatively larger data set. To keep a fair race, in all training processes for Experiment I, the batch size was set to be 50. In all training processes for Experiment II, the batch size was set to be 6, which exhausted the GPU memory for the finest resolution training data. That is, a batch size of 6 was the maximum allowed by the GPU memory. MSE was selected as the loss function for all models and experiments. All models were trained with Adam as optimizer and an initial learning rate of 1e-4. A learning rate reducer was implemented that if the model does not achieve better performance in 3 consecutive epochs, the learning rate would be multiplied by a factor of 0.1. This mechanism has a cool down of 3 epochs and a min learning rate of 1e-8. We also employed an early stop mechanism that if the model does not achieve better performance on a validation set after five epochs, the training would be terminated and the model that performed the best on validation set would be saved. Figure 4.1 shows the training time that different architectures on different resolutions took to converge. Convergence was judged visually using Tensorboard visualizations.

<table>
<thead>
<tr>
<th>Model</th>
<th>Experiment Index</th>
<th>Resolution</th>
<th>Training time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-aggregated CNN</td>
<td>Experiment I</td>
<td>Cluster-level</td>
<td>12h</td>
</tr>
<tr>
<td>Pre-aggregated ANN</td>
<td>Experiment I</td>
<td>Cluster-level</td>
<td>19h</td>
</tr>
<tr>
<td>UNET-40</td>
<td>Experiment II</td>
<td>40m</td>
<td>2d 6h</td>
</tr>
<tr>
<td>UNET-20</td>
<td>Experiment II</td>
<td>20m</td>
<td>39d 6h</td>
</tr>
<tr>
<td>UNET-10</td>
<td>Experiment II</td>
<td>10m</td>
<td>66d 1h</td>
</tr>
<tr>
<td>Multi-Task</td>
<td>Experiment II</td>
<td>10m,20m,40m</td>
<td>51d 17h</td>
</tr>
</tbody>
</table>
Chapter 5
Results and Analysis

This chapter presents the main evaluation of the backscatter prediction models for Experiment I and Experiment II. We also present and explain various evaluation criteria. We conclude that the deep learning model used in Experiment II is superior in predictive performance. More specifically, we provide evaluative analyses to help answer the following research questions:

- RQ1: Are superior results achieved using a clustering-based architecture (Experiment I) compared to a direct backscatter image generation approach (Experiment II)?
- RQ2: Does the multi-task (multi-resolution) architecture provide any advantages compared to the single task network?
- RQ3: What are the common failure cases and successes for the best performing network? Furthermore, are there obvious patterns that illustrate when the model might perform well versus when the model performs poorly?
- RQ4: What performance versus calibration tradeoffs exist for the model? That is, does the model readily generalize to new geographic regions?
- RQ5: What weather information is most crucial for the model to predict backscatter images?
- RQ6: Compared to simply using the last known backscatter image, is there an advantage to predicting intermediate backscatter images?

In our analyses, we answer each question in turn and provide discussion regarding the degree to which each research question is answered. For research questions that are only partially addressed, we also provide a discussion of the limitations of our analysis.
5.1 Common evaluation metric: Improvement Rate

To investigate the final research question: **RQ6: Compared to simply using the last known backscatter image, is there an advantage to predicting intermediate backscatter images?**, and establish a metric that is able to compare the performance achieved between Experiment I and Experiment II. (The MSEs in Experiment I are not the actual reproducing error.) We assume that the model would be usable when the predicted backscatter image has a lower MSE compared with base backscatter image that serves as prediction without going through the Neural Network. That is, we ask how often is the MSE of the predicted image better than simply using the last known backscatter image? Thus, we define another metric, Improvement Rate, as:

\[
f(B_i, \hat{B}_i, B_j) = \begin{cases} 
0, & \sum (\hat{B}_i - B_i)^2 \geq \sum (B_j - B_i)^2, \\
1, & \sum (\hat{B}_i - B_i)^2 < \sum (B_j - B_i)^2
\end{cases}
\]  

(5.1)

\[
I = \frac{\sum_{i=0}^{N-1} f(B_i^l, \hat{B}_i^l, B_j^l)}{N}
\]  

(5.2)

where \( I \) is the improvement rate, \( N \) is the total number of examples in the set, \( B_i^l, \hat{B}_i^l \) and \( B_j^l \) are the target, predicted and base backscatter image for the \( l^{th} \) example.

5.2 Results Summary

5.2.1 Experiment I

**Description:** The 15 clusters clustering result is shown in Figure 5.1. One of the cluster very apparently captured the water bodies in the region studied is shown in Figure 5.2. **Result:** The clustering algorithm based on historical statistics and elevation data captured the water bodies. Water bodies have unique reflection characteristics. The appearance indicates that clustering algorithm reduces the dimension of the problem in a way that is not completely unknowable. However, the issues still remains that the clustering variance adds to the reproducing error. The improvement rate of Pre-aggregated-ANN is 0.37 and the improvement rate of Pre-aggregated-CNN is 0.42. Note that because of the aggregation function is calculated through historical statistics on a particular region, the prediction is not
Figure 5.1: the clustering diagram for 15 clusters

Figure 5.2: the water body captured by clustering marked over the historical mean
able to generalize to other geographic areas. Which should be comparing with the validation performance with architectures in Experiment II.

5.2.2 Experiment II

**Description:** The goodness of fit of the various neural network functions (measured based on the MSE index) is shown in Table 5.1. The mean MSE and 95% confidence interval in the mean of the MSE are shown for the training, validation, and test sets. Recall that the validation set is taken from a same geographic area as the training, while the test sets are taken from geographic regions that are considerably farther from the training set backscatter images. Single task networks are denoted as UNET-XX and MTL networks are denoted as MTL-XX. The “XX” in each name refers to the resolution of 10 m, 20 m, or 40 m. **Result:** The training and validation set exhibit similar values of the MSE. However, the test set data has considerably larger MSE values, consistently, regardless of the network trained. The differences of MSE at varying resolutions are slight, but observable, with finer resolution networks having slightly worse MSE.

<table>
<thead>
<tr>
<th>Model</th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNET-10</td>
<td>5.862 ± 0.087</td>
<td>6.040 ± 0.196</td>
<td>9.915 ± 0.194</td>
</tr>
<tr>
<td>UNET-20</td>
<td>5.558 ± 0.100</td>
<td>5.802 ± 0.245</td>
<td>9.726 ± 0.209</td>
</tr>
<tr>
<td>UNET-40</td>
<td>6.654 ± 0.140</td>
<td>6.839 ± 0.300</td>
<td>9.674 ± 0.179</td>
</tr>
<tr>
<td>Multi-Task-10</td>
<td>6.650 ± 0.115</td>
<td>6.888 ± 0.267</td>
<td>10.387 ± 0.205</td>
</tr>
<tr>
<td>Multi-Task-20</td>
<td>6.335 ± 0.114</td>
<td>6.570 ± 0.265</td>
<td>9.897 ± 0.201</td>
</tr>
<tr>
<td>Multi-Task-40</td>
<td>6.325 ± 0.111</td>
<td>6.553 ± 0.258</td>
<td>9.798 ± 0.197</td>
</tr>
</tbody>
</table>

**Implications:** We are able to answer a number of research questions from these performance numbers and observe some phenomenon not directly related to any of our research questions:

1. Because the training and validation set exhibit similar values of the MSE, it indicates
that networks have no severe over fitting during training.

2. The test set served the purpose of testing the robustness of the neural network function which applied on close geographic areas that the neural networks were not trained with. This is related to our fourth research question, **RQ4: does the model readily generalize to new geographic regions?** It is apparent that the MSE performance measure degrades significantly on the test set. While this degradation is significant, it is unclear if the MSE indicates that the model is unreliable. That is, even with an MSE of 9.5-10, these predictions may still have utility in soil moisture models. Further analysis is needed to understand this.

3. In regards to our first research question: **RQ1: Are superior results achieved using a clustering-based architecture (Experiment I) compared to a direct backscatter image generation approach (Experiment II)?** The cluster-based architecture performed prediction on an intermediate scale of a static region, which resulted in different testing robustness for other geographic areas compared to the deep Neural Network. The aggregation function that worked with the normalized values of the static region was not applicable to another geographic area, resulting in these methods and models unable to extend to other regions. Therefore, we need to calculate the improvement rate of fully convolutional deep neural networks to conclude whether this type of architecture is superior.

4. We can also conclude that there is no advantage to using a Multi-task model for different resolutions. This answers our second research question **RQ2: Does the multi-task (multi-resolution) architecture provide any advantages compared to the single task network?** In all situations, the multi task model provides no meaningful performance boost to the overall MSE. That is, for a given resolution the single task UNET model has a smaller MSE than the corresponding multi-task model. We therefore conclude that the single task model is preferable. All remaining analyses will therefore focus on the single task UNET model.

5. Finally, we can conclude that the finest resolution model (UNET-10) performs similarly
to the MSE of other resolutions. Because there is not a meaningful difference in MSE across resolutions, we will focus on the finest resolution (UNET-10) model for the remainder of our analyses.

5.2.2.1 Improvement Rate for UNET-10

For UNET-10, the improvement rate for validation set is 0.999 and the improvement rate for test set is 0.816. Practically, this means that the predicted image is almost always a better estimation, especially when the geographic region is included in the training data. Moreover, even when the geographic region is not in the training set the model outperforms the simple baseline approach for 81.6% of images. Also, it answers our first research question: **RQ1:** Are superior results achieved using a clustering-based architecture (Experiment I) compared to a direct backscatter image generation approach (Experiment II)? The direct backscatter image generation approach worked significantly better. However, the training time is hundred times longer.

5.3 Additional Analyses

We begin our analysis by further considering our fourth research question: **RQ4:** What performance versus calibration tradeoffs exist for the model? That is, does the model readily generalize to new geographic regions? To further elucidate these questions, an analysis of the test set performance was carried out on UNET-10 to determine whether the degradation on MSE on test set was due to model failure (consistently poor prediction) or outlier data (a few outlier images or artifacts). There is little difference in MSE across the different test set models, thus the same root cause is expected for each model.

**Description:** Figure 5.3 illustrates the spatial location in plan view of the validation and test sets in comparison to urban areas and local terrain slope. Each circle locates the center of an 8 by 8 grid of 3 km by 3 km weather pixels and the color of the circle represents predictive skill for the 8 by 8 grid compared to the previous observation. Each weather pixel is colored according to the average ground slope in the pixel. Yellow boxes in each weather pixel are scaled according to urban density (where urban density is at least 2 percent). The lines outlining the circle area indicates the boundary of the validation and test zones; the
Figure 5.3: Relative increase in MSE for the predicted image compared to the base image for UNET-10 validation and test sets. Each circle is the center of an 8 by 8 grid of 3-km-square weather grid pixels. Urban area density and mean terrain slope within each weather grid pixel are overlain, with urban area density indicated by yellow box size and slope indicated by shading. The urban areas in the northeast and southwest are Oklahoma City and Lawton, respectively. (figure data prepared by Zhongdi Wu, figure generated by Dr. Stuart Stothoff from SwRI)
The measure of performance uses the following summary values:

\[
MSE_{\text{pred}} = \frac{1}{N} \sum_{i=1}^{N} (\hat{B}_i - B_i)^2 / N \tag{5.3}
\]

\[
MSE_{\text{base}} = \frac{1}{N} \sum_{i=1}^{N} (B_j - B_i)^2 / N \tag{5.4}
\]

where \( B_i, \hat{B}_i \) and \( B_j \) are the target, prediction, and base backscatter, and the sum is over all pixels in the 8 by 8 weather grid. The measure \((\frac{MSE_{\text{pred}}}{MSE_{\text{base}}} - 1) \in [-1, \infty)\) compares the prediction to the default assumption of no change in backscatter between image acquisitions.

Blue circles indicate predictive skill by the neural network using weather data, red circles indicate prediction deterioration, and white circles indicate either minimal predictive skill or good skill but little change from the base image to the target image.

**Result:** The neural network exhibited predictive skill (blue circles on Figure 5.3) in the entire validation domain and most of the test domain. For the validation set, the predicted images are almost always substantially superior to baseline as indicated by the dense clustering of blue circles. The south areas have an even better prediction where there was minor urban proportion and flatter plain area. For the test set, however, there are two clusters of much poorer performance. One cluster is located at the northeast corner of the study area, in the vicinity of the Oklahoma City airport. The other cluster is over Lawton and extends north adjacent to the Wichita Mountains.

The shape of the clusters is generally consistent with a strong but localized backscatter feature that the neural network has no way of accounting for properly. The feature must be highly local for the Oklahoma City cluster, because the poorly resolved zone is 8 cells wide, consistent with the feature localized to one weather cell. There are a pair of very strong reflectors west of the airport, which are within the boundaries of the training set, but it appears that the center of the Oklahoma City cluster may be located outside of the training area, in an industrial area north of the airport with many strong potential reflectors. Much of the Lawton cluster is consistent with an observed very strong reflection from the roof of a large manufacturing facility just west of the training set. North of Lawton, the cluster is
parallel to the Wichita Mountains, which feature locally steep slopes and rocky outcrops that are strong reflectors. Most of the images with steep Wichita Mountain slopes also include the rooftop reflector, making it difficult to distinguish between the two potential causes of poor predictive skill. The neural network training occurred in areas with relatively mild slopes, and none of the inputs provide information on ground orientation. These few images hint that ground orientation may be important to consider in more rugged terrain.

**Implication:** From Figure 5.3, the general performance did not collapse outside of the training area, which means that the model can generalize to some of the new geographic regions. Relatively flat-lying areas with low urban fraction did especially well. However, certain features associated with urban areas and variable terrain did greatly degrade performance related to the land conditions. These features appear to be rather localized (e.g., individual large rooftops, rocky outcrops), but the metric for calculating performance includes the local error in each 24-km by 24-km domain intersecting the feature. Therefore, the neural network may be performing quite well in almost all of the 10-m pixels in the test domain but an extreme local feature obscures this good performance. In fact, the neural network may be more representative of the actual soil and vegetation backscatter near the local feature, because it does not include spurious effects from the local feature.

This answers part of our third research question: **RQ3: Are there obvious patterns that illustrate when the model might perform well versus when the model performs poorly?** With the plan view illustrating the general failure conditions, we then take a close look at the worst individual cases in validation and test set.

### 5.3.1 Worst Scenario Visual Inspection

**Description:** Figure 5.4 shows several images taken from the validation set where the top row represents the first channel (VH) of the visualizing backscatter image and the bottom row represents the second channel (VV). For each row, the four images are the base backscatter image that serves as input to the Neural Network; the prediction backscatter image that is generated by the Neural Network; the target backscatter image that the Neural Network prediction should predict. Also reported is the absolute value of the residual between the prediction and target. We draw conclusions from this single example, but addi-
tional numerous examples (with similar conclusions) can be found in the appendix. **Result:** It is clear that in Figure 5.4, there is severe satellite noise in the target backscatter image. Meanwhile, because of the shift clipping nature of data set generation showed in Figure 3.5, this noise pattern has a chance to appear in the training set that, over numerous epochs, the Neural Network learns that noise is consistently found in this geographic area and recognized the land pattern of the location where the noise appeared due to satellite failure or misinformation. **Implication:** Thus, we can conclude that the worst performing images in the validation set are due to satellite noise, rather than any failure of the UNET-10 model.

Figure 5.4: the base, prediction, target, residual plot of the worst scenario in validation set with a MSE of $198.993 \times 10^{-4}$
Figure 5.5: the base, prediction, target, residual plot of a poorer scenario in test set with a MSE of $54.694 \times 10^{-4}$

Figure 5.6: the base, prediction, target, residual plot of a poorer scenario in test set with a MSE of $33.013 \times 10^{-4}$

**Description:** Similarly we visualization in Figure 5.5 and Figure 5.6 typical examples
of bad performance in test set. **Result:** The most common failures occur with images that have a large portion of urban areas. This observation is corroborated with images in the Appendix. For example, in Figure 5.5 the top right portion contains Oklahoma city, where the bright positions show apparent patterns corresponding to shapes and locations of urban structures. Meanwhile, the bottom left portion consists mostly of hills and vegetation areas. There is a clear difference in the residual image where we can observe that the Neural Network predicts these non-urban areas well. Another example of this scenario is shown in Figure 5.6, which includes small portion of architectural structures (the bright spots in the residual image) but is otherwise mostly consistent of vegetation. The model has good prediction at most of the pixels, but have a severely bad overall MSE due to the failure near the man-made constructions. **Implication:** We therefore conclude that the model can generalize to a limited number new geographic regions, but care must be taken to eliminate urban areas as these do not generalize well in our modeling. This is an exciting result because it indicates that the model can be trained without specific information from one geographic region. Further research is needed to understand if this conclusion holds for a much wider range of geographic areas, but we have yet to find evidence that refutes a hypothesis of generalization with the exception of urban areas. We therefore leave research question 4 partially unanswered, where additional analysis is needed to understand the limits of generalization. For example, even though our test set was from a wider geographic range, it still was generally in a region near Oklahoma city (albeit with different landscapes and structures).

### 5.3.2 Artifact Analysis in Test Set

We now turn our attention to our third research question, **RQ3:** What are the common failure cases and successes for the best performing network? To investigate this, we continue to visualize results from our test set. **Description:** Figure 5.7 and Figure 5.8 show examples of a poor and good prediction, respectively, from UNET-10 on testing set data. **Result:** Figure 5.7 shows an apparent failure in prediction. At the center of the predicted backscatter images, there is apparent artifacts that have blurry edges. This scenario can potentially caused by the satellite noise in the training set that, in this par-
ticular example, the pattern in the base backscatter image is similar to that of the noised example, resulting in a large bright area that fails to predict. Figure 5.8 is an example where the target backscatter image was much brighter than that of base, the neural network succeeded in capture that even when the geographic area did not appear in the training set. Further similar examples can be found in the appendix. **Implication:** We conclude that the Neural Network did managed to generated good backscatter images under most of the circumstances. With the exception of a few blurry spots caused by satellite and sensor noise, the overall backscatter predictions are consistently good, with extremely small residual values. We therefore conclude that the methodology could be further improved by detecting and eliminating satellite and noise artifacts in the training data. While this might provide to be a significant effort as finding these noises is resource intensive, it may provide significant prediction improvements.

Figure 5.7: the base, prediction, target, residual plot of a poorer scenario in test set with a MSE of $21.505 \times 10^{-4}$
5.3.3 Crucial Weather Information Analysis

We now turn our attention to our fifth research question, **RQ5: What weather information is most crucial for the model to predict backscatter images?** To investigate this, we freeze the parameters of the neural network and take the derivative of the backscatter values with respect to daily weather grid inputs. However, because of the nature that the output shape of the backscatter is huge, which is $(2384, 2376, 2)$, the derivative operation would result in a gigantic Jacobian matrix that is hard to interpret. We decided to take the sum of two channels separately, which resulted in two values:

$$S_{VH} = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \hat{B}(i, j, 0)$$

$$S_{VV} = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \hat{B}(i, j, 1)$$

(5.5)

(5.6)

where $S_{VH}$ and $S_{VV}$ are the sum of VH and VV channel, $h$ and $w$ are the height and width of the backscatter, $\hat{B}$ is the predicted backscatter. Then take the derivative of the two sums separately with respect to daily weather grid inputs:
\[ G_{VH-b} = \frac{\partial S_{VH}}{\partial S_b} \]  
\[ G_{VV-b} = \frac{\partial S_{VV}}{\partial S_b} \]  
\[ G_{VH-p} = \frac{\partial S_{VH}}{\partial S_p} \]  
\[ G_{VV-p} = \frac{\partial S_{VV}}{\partial S_p} \]

where \( S_b \) and \( S_p \) are the 7-day daily weather grid before base and prediction. All \( G \)s have a shape of (8, 8, 21). Where every first of three is the derivative for precipitation, second of three is min temperature and third of three is max temperature. We then visualize the impact of each term. Taking precipitation before base’s relation with VH sum \( G_{VH-b} \) as example:

\[ L2_{k-p} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\sum_{x=0}^{7} \sum_{y=0}^{7} (G_{VH-b}(x, y, 3k))^2} \]  
\[ M_{k-p} = \frac{1}{N} \sum_{i=1}^{N} \sum_{x=0}^{7} \sum_{y=0}^{7} (G_{VH-b}(x, y, 3k)) \]

where \( k \) is the day index before the sentinel-1 observation, \( L2_{k-p} \) is the L2 norm of gradient of day \( k \) for precipitation, \( M_{k-p} \) is the mean of gradient of day \( k \) for precipitation.

**Description:** Figure 5.9 shows the L2 norm and Mean of gradient relation with respect to the sum of two channels described in 5.11. The x axis indicates the \( k \) value that marks the distance of the day away from the observation. That is, for \( k = 0 \), it means the date given by daily weather grid is the same as the sentinel-1 observation date, for \( k = 1 \), it means the daily weather grid is one day prior to the sentinel-1 observation etc. **Result:** From Figure 5.9, the gradient L2 norm of base precipitation(blue for VH and green for VV) is large when the mean value is close to zero, indicating that the increase in precipitation before base sentinel-1 backscatter does not contribute too much to the sum of predicted backscatter. On the opposite, the precipitation 1 and 2 days before prediction observation have a very high impact and the impact quickly fades away at day 3. However, the impact
Figure 5.9: L2 norm and mean of gradient of precipitation-Sum of VV VH relation

for day 0 is relatively low. **Implication:** The day 0 precipitation having a low impact is counter-intuitive. The observation time described in Chapter 3 indicates that the sentinel-1 observation for the study region is approximately 0:30 AM UTC and the record time of precipitation is 12:00 PM UTC. Which indicates that the precipitation at the same date reported by the system can cover a more than 12-hour time after the sentinel-1 observation of the same date. Therefore, the day 0 readings can be uncorrelated with the soil moisture related value sensed. Such gradient curve indicates that the neural network recognized to ignore same date precipitation with sentinel-1 observation because these precipitations did not influence the sensor result. Meanwhile, the quick fade of impact of precipitation corresponds to the estimation and assumption before the study. That is, precipitation 2-3 days prior is highly impactful to the sensor reading.

**Description:** Figure 5.10 and Figure 5.11 are of same mechanism as Figure 5.9, however, the gradient was taken with respect to min and max temperature (second and third of every three). **Result:** For all the temperature relation curves, the L2 norm of the gradients are
Figure 5.10: L2 norm and mean of gradient of min temperature-Sum of VV VH relation

Figure 5.11: L2 norm and mean of gradient of max temperature-Sum of VV VH relation
large when the mean of the gradients are small (close to zero). **Implication:** The large L2 norm indicates that small temperature change can result in strong sum of backscatter change. The intuition is drastic change in temperature is usually accompanied with rainfall or drought, bringing extreme soil conditions. The small mean indicates that temperature sometimes contributes to an increase in the sum of backscatter and sometimes contributes to a decrease. This indicates that temperature may not be the most crucial weather information for sentinel-1 backscatter. We can now answer our fifth research question: **RQ5: What weather information is most crucial for the model to predict backscatter images?** the answer is the 1 and 2 day precipitation before the prediction observation, with some influence from temperature.

With all the analysis discussed, we conclude that the UNET-10 model is highly preferred for non-urban landscapes. Backscatter image prediction, therefore, is a viable alternative to increase the temporal resolution of Sentinel-1 data.
This thesis proposed a Deep Convolutional Neural Network based method for the estimation of Sentinel-1 backscatter images from Sentinel-1 observations near Oklahoma for the 2015 to 2020 time period. Several Deep Neural Networks were trained on resolutions of 10, 20 and 40 meters, with varying classification task outputs. Among these networks the single task UNET architecture showed greatest capacity on the finest resolution, compared to that of Multi-task learning. The evaluation results based on 10 meter resolution (UNET-10) have an improvement rate of 0.999 on validation data and 0.816 on test data. An analysis of the most crucial input data was also completed showing that daily weather and precipitation are most influential for creating backscatter images. This indicates that there is great potential in producing artificial Sentinel-1 backscatter maps by merging daily weather data with the 2 day precipitation before prediction sentinel-1 backscatter. Even with the modest amount of data collected and analyzed in this study, the proposed Neural Network is able to provide excellent prediction on similar geographic areas.

Meanwhile, the analysis of failure cases in validation and test set exposed several areas with room for improvement via future work: (1) some analysis remained to be done to explore the characteristic of satellite noise and implement appropriate methods to eliminate these noise cases before being used for model training; (2) More geographically diverse data need to be collected to potentially increase the robustness to be able to apply to any given region globally; (3) Urban area or areas with above average human development can be masked to avoid noise and to provide more accurate prediction.
APPENDIX A

6.1 Examples of well prediction on test set
6.2 Examples of poor prediction on validation set
6.3 Examples of poor prediction on test set
BIBLIOGRAPHY


[30] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature Cell Biology*, vol. 521, no. 7553, pp. 436–444, May 2015, funding Information: Acknowledgements The authors would like to thank the Natural Sciences and Engineering Research Council of Canada, the Canadian Institute For Advanced Research (CIFAR), the National Science Foundation and Office of Naval Research for support. Y.L. and Y.B. are CIFAR fellows. Publisher Copyright: © 2015 Macmillan Publishers Limited. All rights reserved. 7

67


[38] USGS. National Elevation Dataset.


“Ecologically-relevant maps of landforms and physiographic diversity for climate