AFRL-RY-WP-TR-2023-0299

AN INVESTIGATION IN THE USE OF HYPERSPECTRAL IMAGERY USING MACHINE LEARNING FOR VISION AIDED NAVIGATION

Isaac Thomas Ege
University of Dayton

JANUARY 2024
Thesis

STINFO COPY

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
January 2024	Thesis	13 April 2023	13 April 2023

4. TITLE AND SUBTITLE

AN INVESTIGATION IN THE USE OF HYPERSPECTRAL IMAGERY USING MACHINE LEARNING FOR VISION AIDED NAVIGATION

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8650-22-C-1017	N/A	61102F/62020F/63203F
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
3001/0200/69DF	N/A	Y266

6. AUTHOR(S)

Isaac Thomas Ege

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Dayton
300 College Park
Dayton, OH 45469

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'SREPORT NUMBER(S)
Air Force Research Laboratory Sensors Directorate Wright-Patterson Air Force Base, OH 45433-7320 Air Force Materiel Command United States Air Forces	AFRL/RYWN	AFRL-RY-WP-TR-2023-0299

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goal of this research is to investigate the possible use of hyperspectral imagery in vision-aided navigation using machine learning. This paper is to show that hyperspectral can and should be used in vision-aided navigation and that it is more effective than traditional RGB images. The database of hyperspectral imagery used was NASA's AVIRIS, due to the availability of large amounts of data across the United States. The neural network that was trained and tested was convolutional neural network that uses a multi-scale filter bank and residual learning to improve the network. A second Network created, trained, and tested on RGB images was used to create comparison data to the results of the hyperspectral network. These neural networks were tested in two ways, the first being a full data test even though each of the classes had a considerable difference in data, and the second being a test that had an even size data between classes test. The results investigated the best hyper parameters that could have been used in both networks. The best hyperspectral network out performed the best RGB network on every test. This shows a strong potential of using hyperspectral images future research for vision aided navigation.

15. SUBJECT TERMS

hyperspectral, navigation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
Unclassified	Unclassified	Unclassified	SAR	54

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Dylan Bowald	N/A

AN INVESTIGATION IN THE USE OF HYPERSPECTRAL IMAGERY USING MACHINE LEARNING FOR VISION AIDED NAVIGATION

Thesis

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Electrical Engineering

By

Isaac Thomas Ege

Dayton, Ohio

May, 2023

AN INVESTIGATION IN THE USE OF HYPERSPECTRAL IMAGERY USING MACHINE LEARNING FOR VISION AIDED NAVIGATION

Name: Ege, Isaac Thomas

APPROVED BY:

Brad Ratliff, Ph.D. Advisory Committee Chairman Professor and Chair, Electrical and Computer Engineering	Jason Kaufman, Ph.D. Committee Member Professor, Electrical Engineering

Eric Balster, Ph.D.
Committee Member
Department Chair and Professor,
Electrical and Computer Engineering

Robert J. Wilkens, Ph.D., P.E. Associate Dean for Research and Innovation Professor School of Engineering	Gül E. Kremer, Ph.D. Dean, School of Engineering

ABSTRACT

AN INVESTIGATION IN THE USE OF HYPERSPECTRAL IMAGERY USING MACHINE LEARNING FOR VISION AIDED NAVIGATION

Name: Ege, Isaac Thomas
University of Dayton

Advisor: Dr. Brad Ratliff

The goal of this research is to investigate the possible use of hyperspectral imagery in vision-aided navigation using machine learning. This paper is to show that hyperspectral can and should be used in vision-aided navigation and that it is more effective than traditional RGB images. The database of hyperspectral imagery used was NASA's AVIRIS, due to the availability of large amounts of data across the United States. The neural network that was trained and tested was convolutional neural network that uses a multi-scale filter bank and residual learning to improve the network. A second Network created, trained, and tested on RGB images was used to create comparison data to the results of the hyperspectral network. These neural networks were tested in two ways, the first being a full data test even though each of the classes had a considerable difference in data, and the second being a test that had an even size data between classes test. The results investigated the best hyper parameters that could have been used in both networks. The best hyperspectral network out performed the best RGB network on every test. This shows a strong potential of using hyperspectral images future research for vision aided navigation.

My parents, They are wonderful people who have helped me by giving me the tools I needed to complete this thesis and who often know me better than myself.

ACKNOWLEDGMENTS

There are a large number of people I could talk about here. The most engaging and helpful person who guided me through machine learning is Dylan Bowald, my AFRL/RYWN customer. I couldn't have made it this far without his supporting attitude and patience with me. My supervisor, Andy Thompson helped pushed me when I needed it yet also gave me the space to work the way I wanted. Dr. Brad Ratliff is the great academic advisor who helped me find this project and gave me this wonderful opportunity.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Purpose

The goal of this investigation is to assess the use of hyperspectral imagery through machine learning as applied to vision-aided navigation. This is an initial investigation, with the hope that further studies can elaborate on these findings. These future studies should investigate if this approach is timely, accurate, and applicable. It can be an extremely useful application in defense work as another method of navigation without the use of GPS.

1.2 Hyperspectral imagery

Hyperspectral imagery is a method of capturing images. A hyperspectral image has hundreds of layers (usually between 100 and 200) of "color" compared to a 3 layer RGB Image. Each "color" of the hyperspectral image is called a band. Each band tries to measure a specific wavelength with 10nm of difference between bands. Hyperspectral images will also contain a larger range of information, containing wavelengths such as

1. Ultra Violet(UV) wavelengths(∼200-400 nm)
2. Visible wavelengths(∼400-900 nm)
3. Infrared(IR)(∼900-12000 nm) which can be split into
   (a) Near IR(NIR) wavelengths (∼900-1700nm)
   (b) Medium Wave IR(MWIR) (∼3000-5000nm)
   (c) Long wave IR(LWIR) (∼8000-12000nm)

The information in hyperspectral imagery allows for more fine grained information as compared to RGB images. Previous research[2] has used the spectral response of pixels to determine the material of that pixel. An example of previous research has been done with water and concrete[2]. This is a very useful tool that hasn't been utilized with RGB images. Hyperspectral imagery is relatively new, when considering how long image collection has been an activity. Hyperspectral imagery has only been in existence since the 1990s, meaning there has not been much time investigating the many powerful applications of this modality. Since this time, more accurate band widths have been able to be collected and it has been applied to a wide variety of fields, from medicine to farming[3].

The data used for this study came from NASA's Jet Propulsion Laboratory. Specifically, the Airborne Visible / Infrared Imaging Spectrometer(AVIRIS)[4] is the system that is used for hyperspectral imagery. AVIRIS has been integral in the study of remote sensing for many locations across North America, Europe, portions of South America. The AVIRIS data set has over 4000 images, many with gigabytes of image data. This data is consistent in that it takes images with the same 224 bands ranging (in wavelengths) from 365 nm to 2500 nm approximately separated by 9.5 nm. AVIRIS data has been successfully used for a variety of research from measuring the atmosphere to measuring biomass burning to spotlight calibration[4]. Lastly and most importantly, it is easy to access the data and use it for this project. With this much publicly available data, AVIRIS is an ideal choice for this project.

AVIRIS provides its data as orthocorrected radiance data. Orthocorrected means that the image has been adjusted to take into account the three-dimensional texture of the earth and the rotation of the plane such that the data was taken immediately above each pixel. This means that each image of the same area will be shown the same. This is extremely

useful for this study as it allows for better accuracy and comparison of images. Radiance data means that the data collected has been taken such that nothing in the air has been taken into account to remove noise. This will lead to effects seen in Figure 1.1, shows the average value amongst all of the pixels of each band across several images. This shows there are some regions (bands 1363-1383 nm(between the blue lines), 1830-1950 nm(between the red lines), and 2485-2495 nm(after the green line) that contain no information as it is just a blurry signal. The reason why these sections of the images are blurry is due to the water absorption lines[5]. The light waves captured in these images are interacting with the water molecules in the air. The water molecules refract the light waves in multiple directions and produce scatter, which reduces the clarity and magnitude of these bands as seen in Figure 1.1. Information captured in these pixels is not useful for the future of this project.

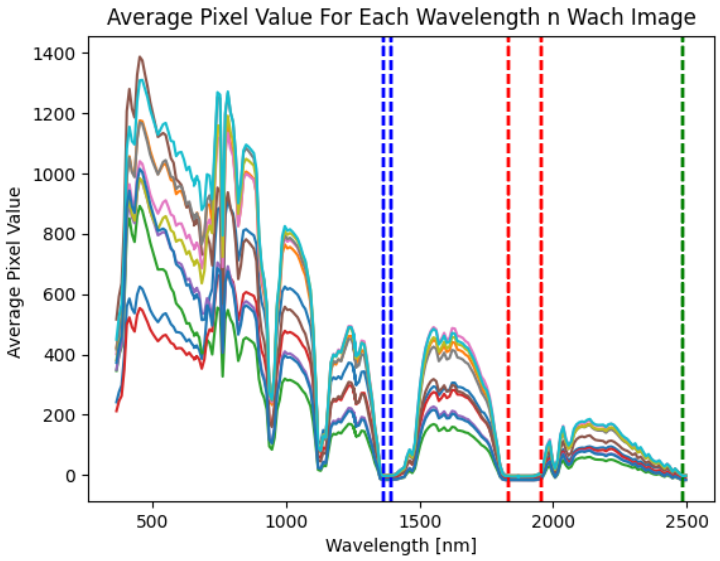

Figure 1.1: Average value per band across several images

1.3 Overview of Neural Networks

A neural network is a type of algorithm that learns a functional mapping when provided with sample input and output data. A traditional neural network starts with several layers of nodes. Each node from one layer is connected to the next layer by $net_j = N_x * W + n$ $F(net_j) = N_{x+1}$ Where N is the node, x is the current layer, F is the relu function, and W and N are variables to change. The sum each of the previous outputs for the node is used to make the next layer. This process of calculating a node can be seen in Figure 1.2. Some important notes are that N_0 is a single color of the image and N_{END} is the classification layer. This makes a network of thousands of connections between hundreds of nodes each with its own connections. This is the process of a neural network. To train a neural network it must be tested with an image. When the image gets to the classification layer (the last layer), it is compared to the actual result. Lastly, back propagation is conducted to process through the neural network in retrograde. This process estimates the best direction to change every node connection to most likely improve the function. Then, the process is repeated. The more times the process is repeated, the better the algorithm performs. It is often repeated hundreds of times [6].

A convolutional neural network is similar to a neural network but with a small difference in that it tries to consider spatial information while determining the output. The way convolutional networks consider spatial information is by including a window. This window is a square with the same depth as the previous layer. It uses similar calculations to the ones stated earlier in the paper except in the square window. Subsequently, it sums those values to be the output node. Then it repeats this function with the same N and W values as it moves across the previous layer until it has covered every node. This process can be repeated on the same layer but with different N and W values to get different results.

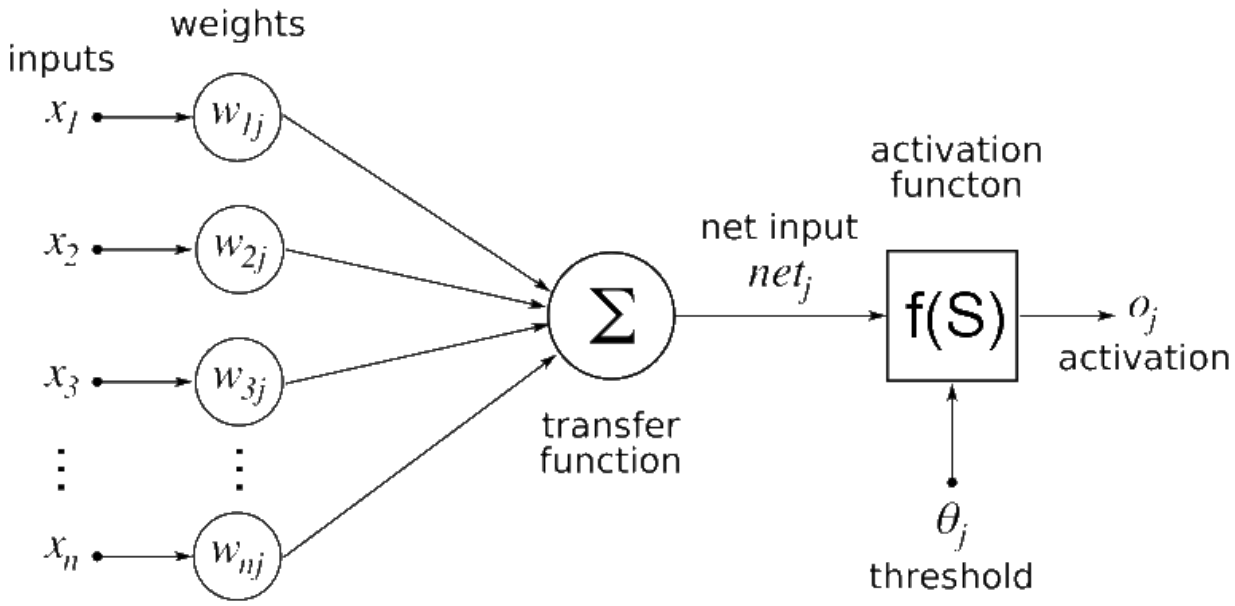

Figure 1.2: Single layer of the neural network

Those processes are combined to form a convolutional neural network layer. Again, this process is repeated slowly shrinking between each layer until a relatively small output layer then transforming it into a traditional neural network. This process utilizes the spatial information because the window allows the adjacent nodes of one layer to influence the output node. The method to train a convolutional neural network is similar to the neural network but requires less time and training data data[6].

1.4 Previous Research

There have been multiple investigations about neural networks and hyperspectral imagery [7, 8, 9]. However, neural networks and hyperspectral imagery have not been used to aid in visual navigation as done here. There has been research into all types of neural networks and research into hyperspectral imagery. There have been investigations into using hyperspectral imagery for object detection/classification using neural networks. This is

similar to the method used here but instead of within a single image, this study is investigating its usefulness across several images. This is similar in the idea that it determines the class of one pixel vs one image but differs in the amount of information because it would have the rest of the to classify it. Another challenge with several of these studies is that not enough information is published to effectively investigate the idea[10]. Convolutional neural networks are useful because they try to extract more of that information by repeatedly going over the same area and by combining the spectral (between bands) information and spatial (between pixels) information to see if the network can identify different relationships. There are several in-depth investigations about the effectiveness of several different neural networks on image classification using hyperspectral imagery[11, 12, 10, 13]. They each take their own approach to classify images but all show promising results in applying convolutional neural networks to sub image classification.

1.5 Network Architecture

Since this project is similar to the image classification of the previous research but with more information, it was decided to use one of the previously designed networks. The network that would work best for this was Lee et al's network[1]. This network has several processes that allow for better network training. The first process that allows for better training is the multi-scale filter bank. This process is seen in the first couple of layers. A $1 \times 1 \times c$, $3 \times 3 \times c$, and $5 \times 5 \times c$ (as long as there is more than one filter, it is a multi-scale filter bank) filter is combined when going over the image. This allows the network to detect patterns in the spectral dimension with the 1x1 filter and then slowly give more importance to the spatial information with the larger filters. A visual representation of the multi-scale filter bank can be seen in 1.3 second major contribution is residual learning. This is a

process of allowing the network information from the previous layers of the network to compute future layers. This allows the network to skip or regain information that was lost in previous layers of the network improving the accuracy of deep neural networks [14]. A visual representation of the residual learning used in the paper can be seen in figure 1.4

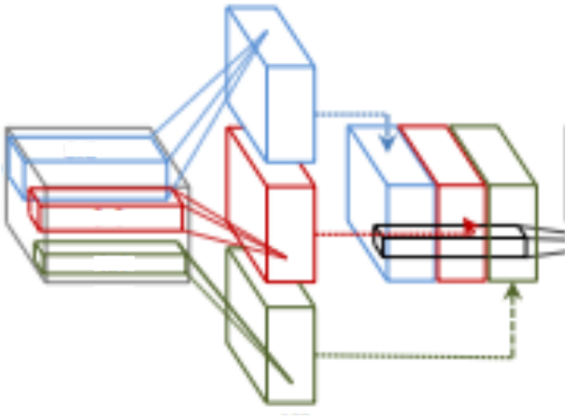

Figure 1.3: The Multi-Scale Filter [1]

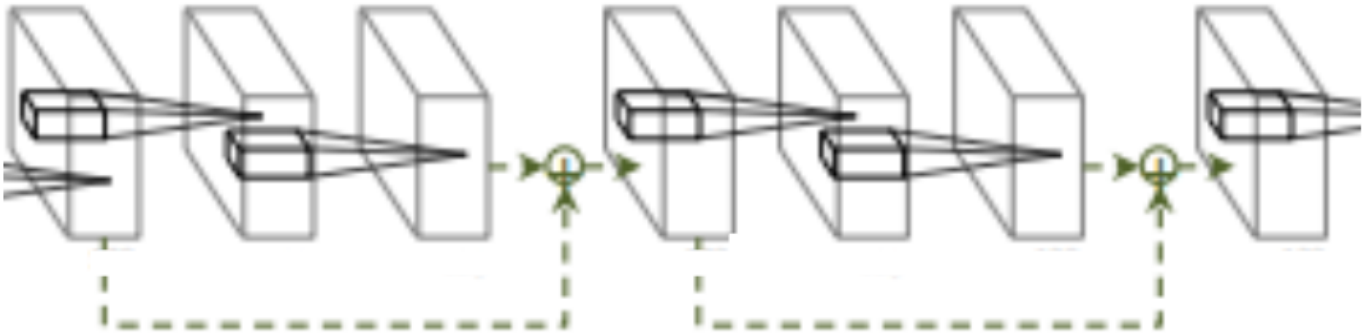

Figure 1.4: The Residual Learning [1]

1.6 Processing Plan

When working the network, it needs to be able to download a large amount of data from AVIRIS such that it can be used in classification based on location. This means it will have to download, edit and modify the AVIRIS data such that it is in a usable state for a neural network. Another neural network using the same architecture will be used on modified AVIRIS data to represent RGB data. This will allow an clear and effective comparison of the same approach. This will show an effective comparison if it is able to work more effectively than RGB images.

CHAPTER II

APPROACH

The goal of this chapter is to explain the how and why the data was chosen to split, update and modify. What were the processes to interacting with and how to best optimize them. This will be the entire process from AVIRIS to the exact input into the neural network.

2.1 Interacting with AVIRIS

AVIRIS data is extremely useful data as it contains thousands of hyperspectral images each with metadata. All of this information is valuable to selecting which images are useful and analyzing the criteria for why some images should be removed. Examples of these criteria are the ground sampling distance, longitude and latitude values, dates taken, download links, etc. All of this information is available to researchers in Google Sheets, freely available to interact with and download. Next, all of the specified criteria which will be discussed later was filtered out. With the set of images the link is downloaded the file is unzipped and looking for two file types: the image the ".hdr" file and the matching hyperspectral file. The ".hdr" file is a file that gives context of how to read the hyperspectral file containing all of the data. With the Python library Spectral, there was a method to extract the hyperspectral image data.

2.2 Selecting Images

AVIRIS gives access to all of the data. The selection of data needs to be considered. It was expected that locations with distinct appearances would be identified, such that it is clear that these are from different locations. Secondly, the more amount of data the better

the learning algorithm will perform. Figure 2.1 is an example of the dataset collection. The red rectangles are each a hyperspectral image. The green squares are each a different region that I took data from. From left to right:

1. California: This is a desert region.
2. New Mexico: This is covering specifically White Sands National Park, a unique looking desert region.
3. Minnesota: This is a forest-covered region.
4. Michigan: This is the wetlands around the Great Lakes.
5. Florida: This is a swamp land.

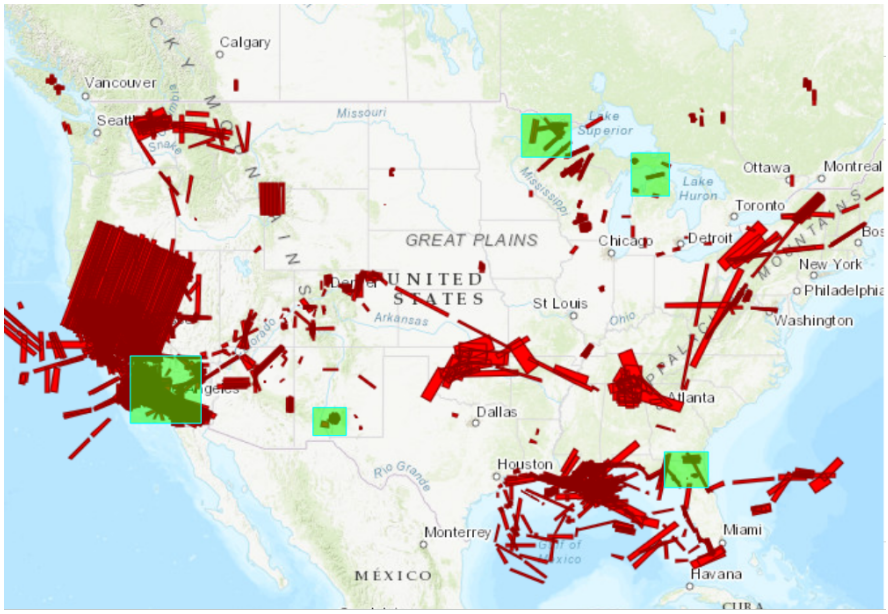

Figure 2.1: Locations of data collection

Each region will be a class for the output of the neural network. This means that the network should be able to guess each of these regions which has a major characteristic that is associated with them.

The third major characteristic for each image is that they have the same ground sampling distance (GSD). This means that each pixel had to represent the amount of area on the ground. The AVIRIS dataset has a large variety of GSDs. This variety can be seen in Figure 2.2 where it ranges from some images having a GSD of 1 to 18 meters/pixels while there are two most common occurrences are around 3.5 and 16. This is an important thing to change due to the limitations of machine learning. A neural network expects a consistent-sized input in order to determine the image. There are two approaches to resolving this problem with differing GSD's. The first is to add empty space to the outside edges of the images but then the features change making it take longer to identify and detect the outcome. The second option, the one which was opted for, was to up sample or down sample the images such that they have the same GSD. The reason to choose this option is to have the features be the same size. This makes the neural network quicken its learning compared to the flexibility of the other option. The first thing to choose was a GSD size that would be matched to. Five meters/pixel would be the most likely GSD for where this is going to be implemented. The reason this was chosen as if we were to take images they would be approximately that size and there was a large sample of data in that region. The images with smaller GSD images were then down sampled with a third-order filter such that they match the GSD of 5 meter/pixel. This down sampling filter allows the filter to react to changes in data more while still being relatively quick.

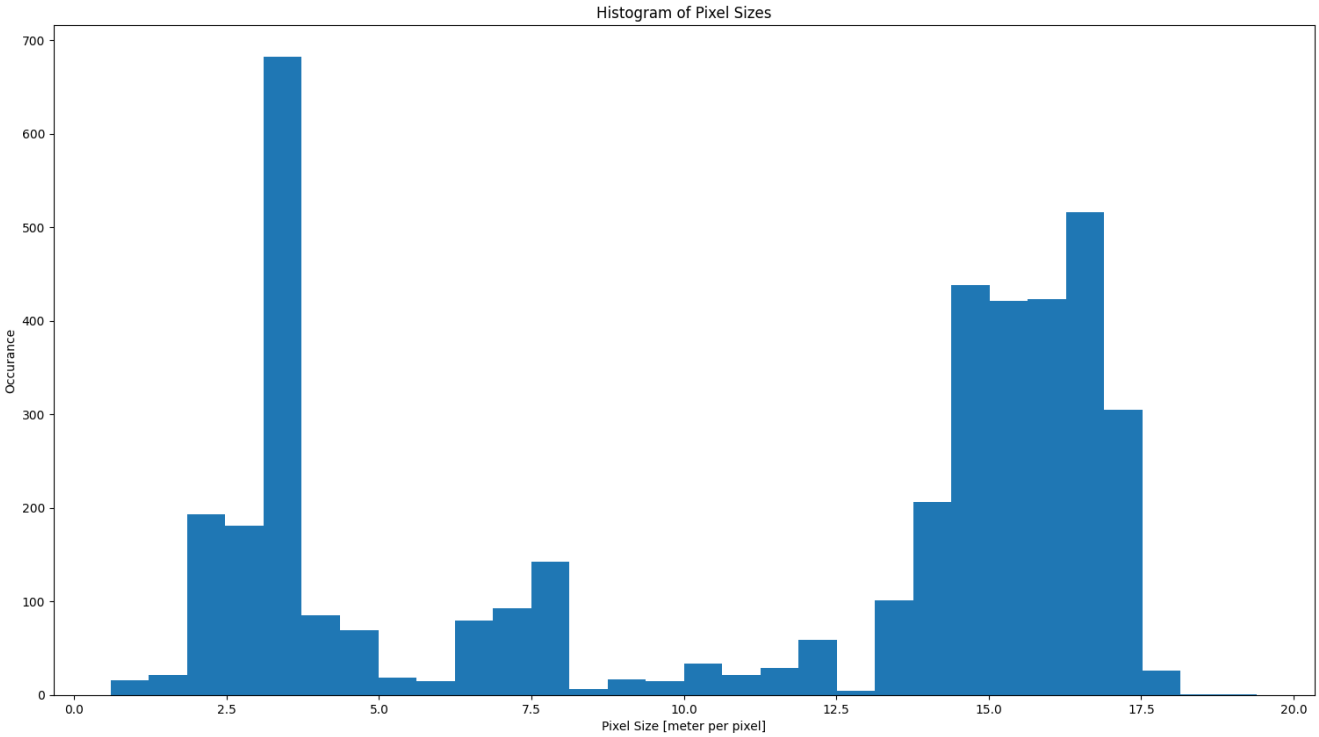

Figure 2.2: Histogram of Ground Sample Distance

2.3 Splitting Images

When training a neural network, to properly learn it needs a consistent input size image both in terms of area covered and pixel size. The easiest way to do this is to base it on longitude and latitude by accessing all of the metadata from the AVIRIS website. Then, image and the image-specific metadata was downloaded,unzipped, and combined into a Geotiff, which is a more familiar format to manipulate in the future as previous code has used the Geotiff formatting that allows the access of metadata in a quicker manner. Before splitting the images, a grid of coordinates between any 2 whole-value longitude and latitudes was created for reference. This was done by

$$new_lat = arcsin(start_{lat}) * cos(distance/R) + cos(start_{lat}) * sin(distance/R) \text{ [15]}$$

$$new_{long} = start_{long} + arctan2(sin(90) * sin(distance/R) * cos(start_{lat}), cos(distance/R) - sin(start_{lat})) * sin(start_{lat}) \text{ [15]}$$

Where $start_{lat}$ and $start_{long}$ are the starting point of the grid square in radians, distance is the size of the grid square meters and R is 6,378,100 which is the radius of the earth in meters. Lastly, end_{lat} and end_{long} are the ending points of the grid square. This makes it such that there is a consistent splitting of squares no matter where the image was taken. This process was repeated until it crossed a longitude or latitude whole value size. This created a repeatable set of coordinates that were approximately the same size in meters and were quickly calculable no matter where the image was taken. The effect of this chipping process can be seen in Figure 2.3. Each of these were taken from a different flight and on a different day now they have been chipped to show the same area. The black parts of the image are the unmeasured area of images due to the way the hyperspectral images were taken but added to make the images rectangular.

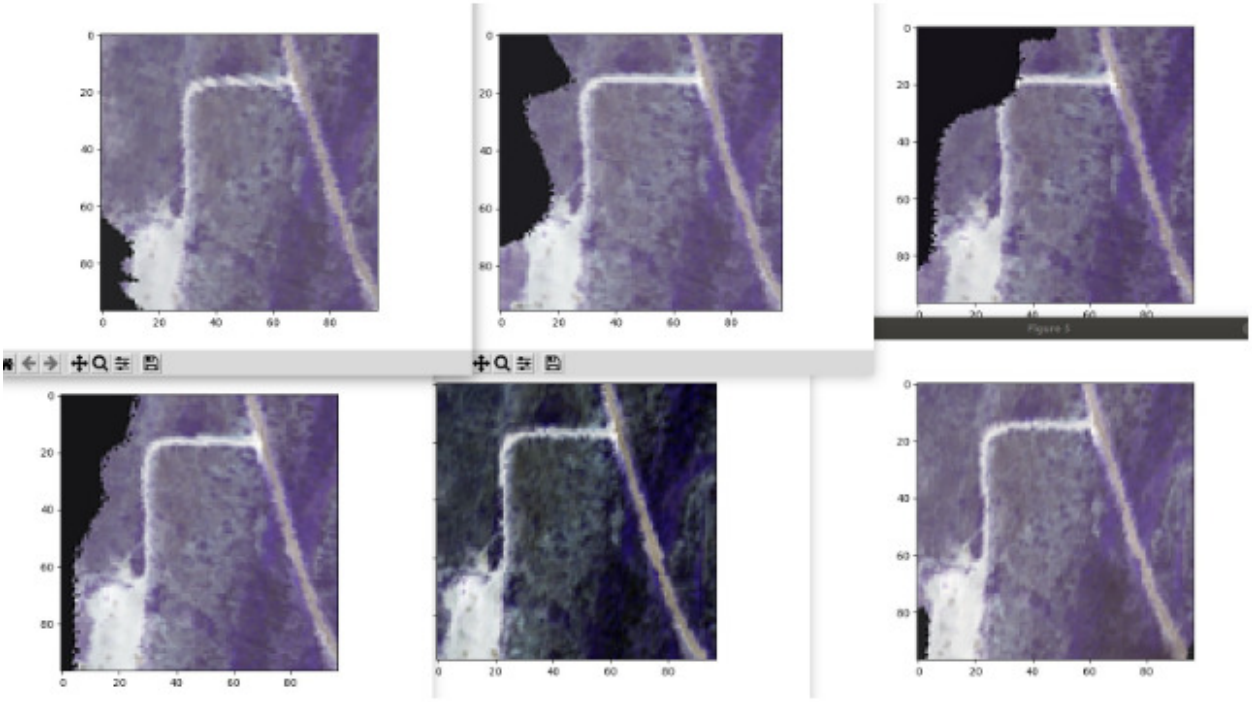

Figure 2.3: Six chipped images of the same location

Table 2.1: Size of the Classes

	California	New Mexico	Minnesota	Michigan	Florida
Number of Full Images	8	8	8	8	8
Number of chipped images	4988	190	3811	502	1177
Number of trained images	3481	139	2678	353	808
number of validation images	992	24	753	100	245
number of testing images	515	27	380	49	124

2.4 Data specifications

This section will describe the particular aspects of the data. This process followed the criteria of being under five pixel per meter, had at most eight images, and within the selected regions. After acquiring the data, the data was split into three sets at random: train, validation, and test set. The train category has 70% of the data. This is because it needs as much information as possible in order to get the best accuracy[16]. The validation category contains 20% of the data. This category allows for the optimization of the meta-variables of the network in order to optimize the network. Lastly, the testing category contains 10% of the data. This category is used as a final test to see without the knowledge of the network or optimizer. The data in Table 2.1 shows the number of images in each category within each class. This shows us that California data set has the most data, which means that the original images were considerably larger, and New Mexico data set has the least amount of data, correspondingly. This means the New Mexico images were significantly smaller in number.

2.5 RGB image

The next major hurdle was the creation of the RGB images that are based on the hyperspectral images. This is a process that was started with the Prosilica GT 6400 camera that is currently being used on another project nearby so it was chosen as a baseline camera. This camera has a spectral response to detecting levels of red green and blue. This response can be seen in Figure 2.4. This shows the red response in red, green response in green, blue response in blue, and grayscale response in grey. The number of hyperspectral bands from 400nm to 1000nm was first determiend to be 63. Next, each band's location was determined and the spectral response of each of the colors was approximated. Lastly, this value was multiplied by each pixel value for that band, then it was divided by the maximum. This approximates the response that an actual RGB image would take as seen in Figure 2.5

2.6 Network Choices

This network was based on the network from Lee et al[1] and was designed to quicken the speed and accuracy of deep convolutions neural networks. Lee et al found that the Multi-Scale Filter Bank helped hyperspectral the network determine the images quicker than without it. With this reason it was added into the neural network. the Multi-Scale Filter Bank was slightly change in their paper they had 3 filters which were $1 \times 1 \times c$, $3 \times 3 \times c$, and $5 \times 5 \times c$. $5 \times 5 \times c$ was removed to quicken the process. Each of these filters created an output of the same size of the input image and 128 channels deep. They were stacked to form a layer with 256 channels. The second major idea they added to their paper was the residual learning. This allows for the neural network to skip layers if the neural network is too deep, and carries some information from previous layers if needed. This allows for a more accurate deep neural network. This was essentially implemented the same way where

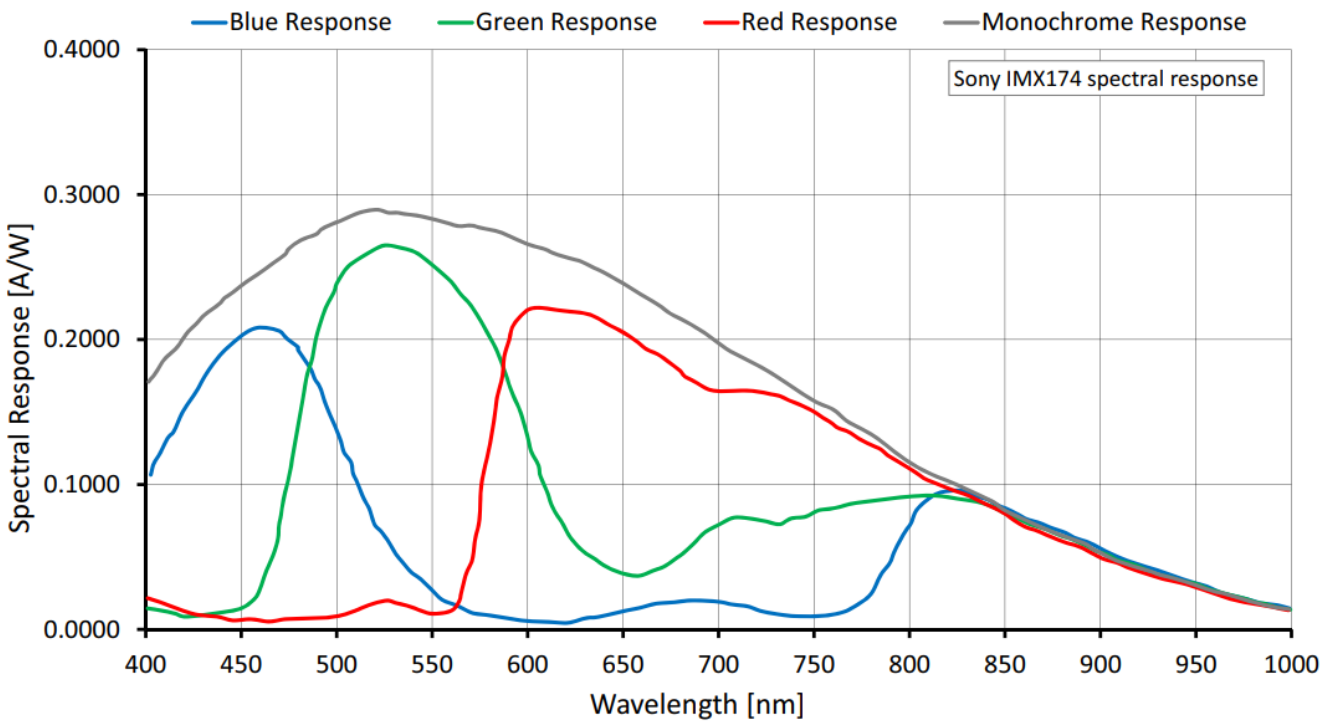

Figure 2.4: Spectral Response of the Prosilica GT 6400

Figure 2.5: Example RGB converted image

there was two layers of the network then the residual learning. The other major network choice is an optimization algorithm called Adam. This is an algorithm that will help the back propagation perform quicker with a learning rate and beta values. In order to find the optimal values for the learning rate and beta values Several runs will need to be made in order to determine the best values and will be discussed in the results section.

CHAPTER III

TESTING AND RESULTS

Since there was an imbalance in the classes of data there were two test created. The first one will use all of the data to train and test the neural network in order to see what the results are. The second test will limit the training, validation, and testing sets of each class to the same size. This means each class will have a randomly selected set of 190 train images, 24 testing images, and 27 validation images.

3.1 All Data Test

This was the first test to be performed. This test used all of the available information and was run seven times for each hyperspectal and RGB network. First, it ran though several learning rates to see which learning rate was the best. After determining the best learning rate for both the RGB and the hyper network it then determined the best values for the beta values. Each program ran for 30 times though the training and validation data, also know as an epoch. The starting beta values for Adam optimization were (0.9, 0.999) these will be tested afterwards.

Table 3.1: All Data Test for Hyper Spectral Results for Variable Learning Rate

Learning rate	best loss	best loss validation	best accuracy	best accuracy validation
0.05	1.31	1.21	0.447	0.466
0.01	1.19	1.21	0.879	0.872
0.005	0.075	0.095	0.971	0.965
0.0001	0.00045	0.0045	1	0.992
0.00009	0.00035	0.00085	1	0.993
0.000075	0.00037	0.00099	1	0.992
0.00005	0.00032	0.00099	1	0.992

Table 3.2: All Data Test for RGB Results for Variable Learning Rate

Learning rate	best loss	best loss validation	best accuracy	best accuracy validation
0.05	1.37	1.39	0.461	0.466
0.01	1.177	1.181	0.498	0.466
.005	0.39	0.605	0.86	0.692
0.0001	0.0055	0.444	0.996	0.849
0.00009	0.00067	0.474	0.997	0.842
0.000075	0.00023	0.445	1	0.88
0.00005	0.00023	0.435	1	0.88

The information in Table 3.1 and Table 3.2 show much of the available information from each of the tests. These Tables show the hyperspectral network performed best after using a learning rate of 0.0001 while the best learning rate for the RGB network was 0.00005, half of the magnitude of the hyperspectral network. Since the hyperspectral network performed similarly between learning rates of 0.0001 and 0.00005, both networks will be set to a learning rate of 0.00005 to keep a consistency between networks when testing for beta values. Figure 3.2 and Figure 3.3 show the data accuracy and loss respectively during the training process of the RGB network. Figure 3.1 displays a confusion matrix for the last validation run of the RGB network. This shows where the hyperspectral network predicted each image to be compared to where each true image was. Figures 3.5, 3.6, and 3.4 Similarly, show the same images except for the hyperspectral network. There is plenty to discuss in the discussion section but first we will optimize the beta values for both Networks.

The information in Table 3.3 and Table 3.4 show the information for the variable beta values for the RGB network and hyperspectral network, respectively. The best beta values for the RGB network are either 0.88 and 0.999 or 0.95 and 0.99999. The best beta values for the hyperspectral network is 0.88 and 0.999. In Figure 3.9 and Figure 3.8, it shows the

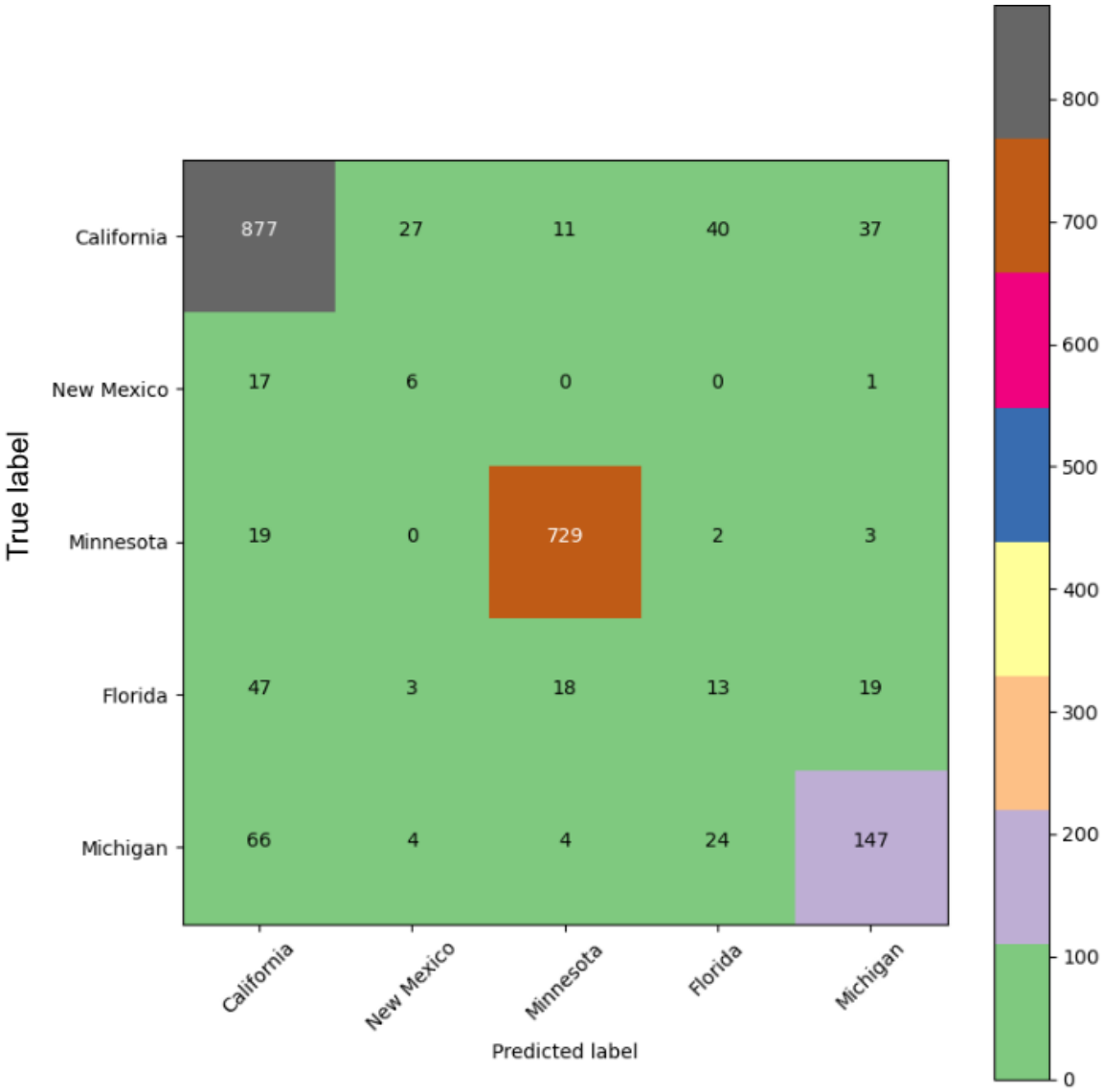

Figure 3.1: Confusion Matrix for validation for RGB Learning Rate 0.00005

Table 3.3: All Data Test for RGB Results for Variable beta values

$Beta_1$	$Beta_2$	best loss	best loss validation	best accuracy	best accuracy validation
0.9	0.999	0.33	0.40	0.88	0.85
0.8	0.9	0.4026	0.91	0.87	0.65
0.85	0.99	0.44	0.48	0.87	0.4
0.88	0.9999	0.16	0.39	0.94	0.86
0.87	0.95	0.36	0.47	0.87	0.84
0.95	0.99999	0.133	0.42	0.95	0.86

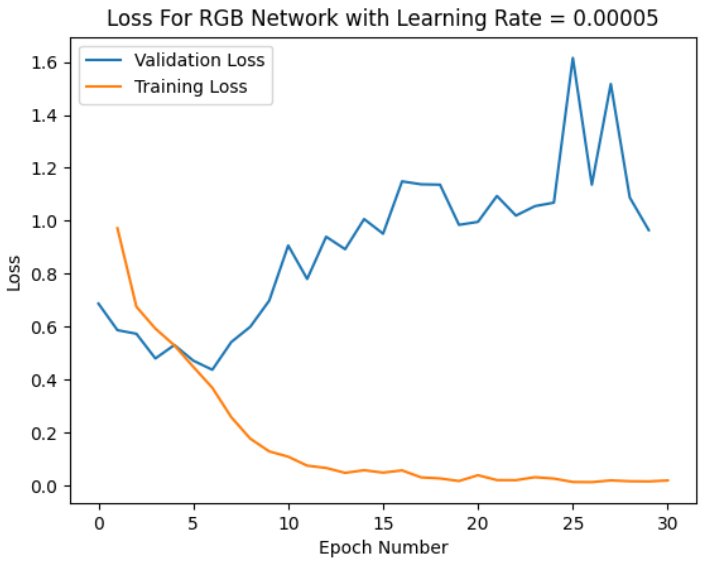

Figure 3.2: Learning Accuracy of hyperspectral network with Learning Rate of 0.00005

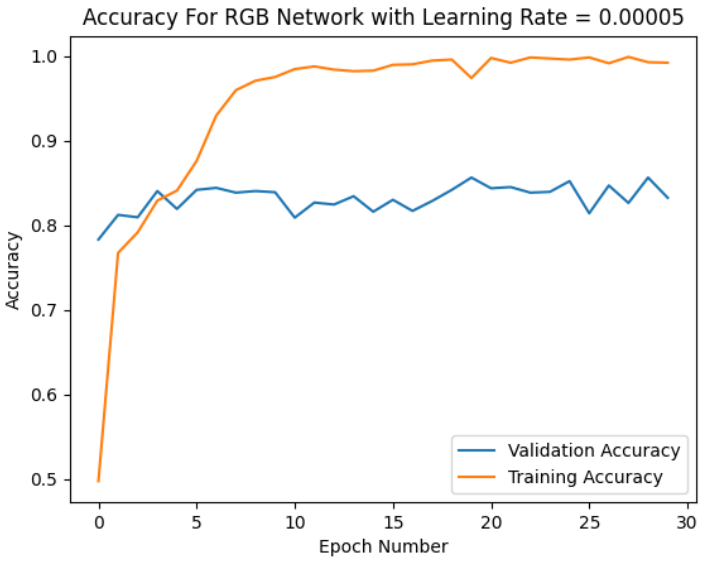

Figure 3.3: Learning Loss of hyperspectral network with Learning Rate of 0.00005

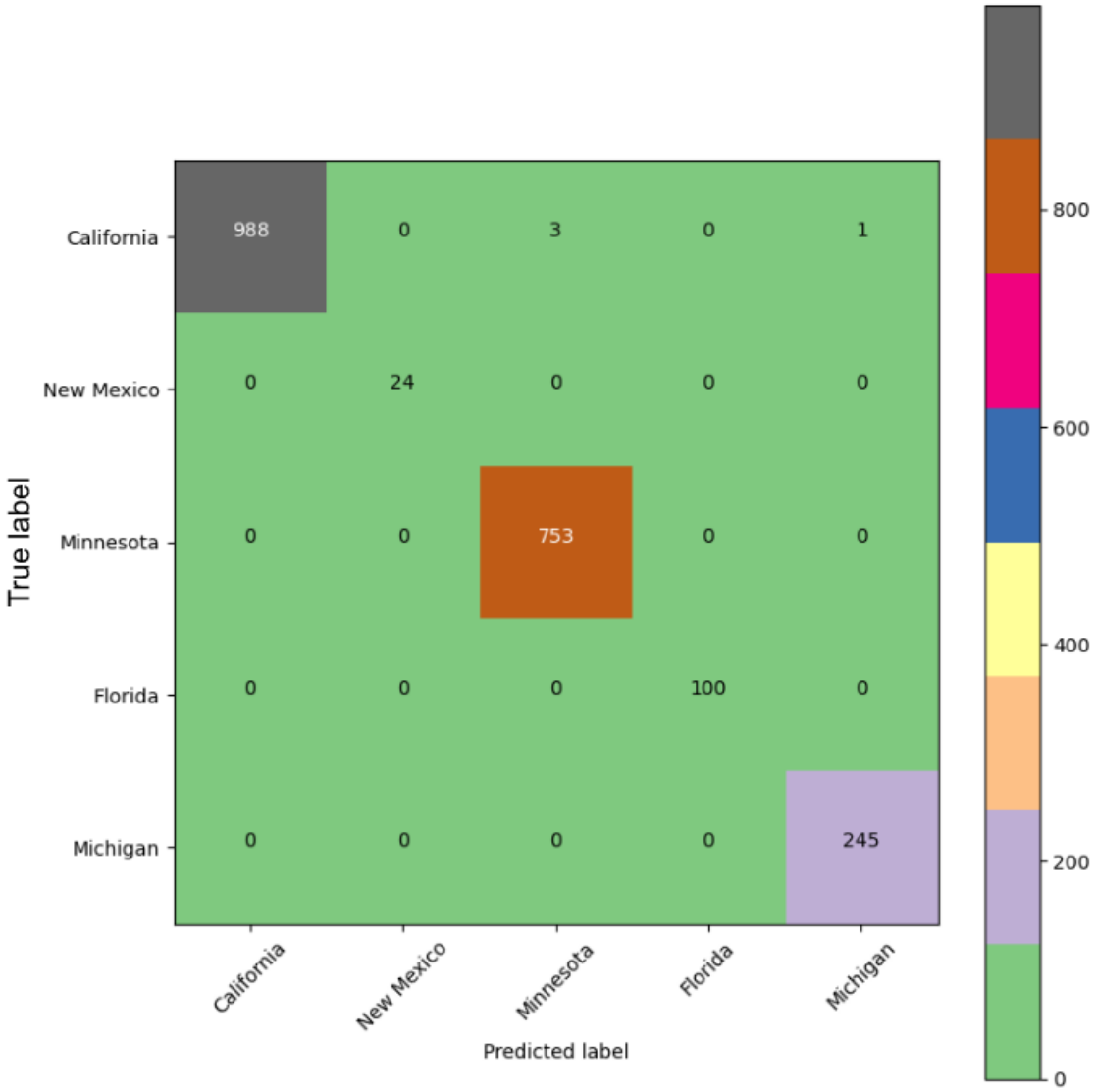

Figure 3.4: Confusion Matrix of hyperspectral network with Learning Rate 0.00005

Table 3.4: All Data Test for Hyperspectral Results for Variable beta values

$Beta_1$	$Beta_2$	best loss	best loss validation	best accuracy	best accuracy validation
0.9	0.999	0.0062	0.061	1	0.98
0.8	0.9	0.016	0.041	1	0.91
0.85	0.99	0.0015	0.052	1	0.98
0.88	0.9999	0.030	0.023	1	0.99
0.87	0.95	0.0087	0.079	0.99	0.98
0.95	0.99999	0.00091	0.13	1	0.98

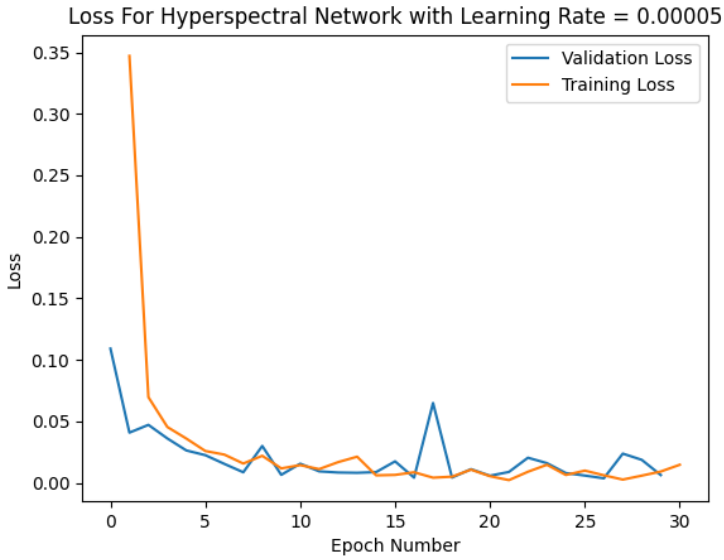

Figure 3.5: Learning Accuracy of hyperspectral network with Learning Rate of 0.00005

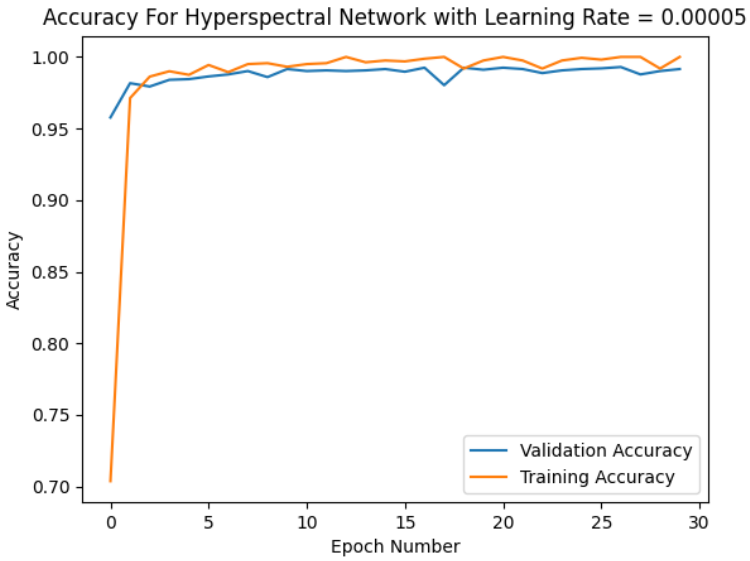

Figure 3.6: Learning Loss of hyperspectral network with Learning Rate of 0.00005

running of the loss and accuracy progression, respectively, progression as the training for the RGB network. Similarly, in the Figure 3.12 and Figure 3.11 shows the progression of the training for the hyperspectral network for the loss and accuracy. The Training was extended from 30 to 60 epoch as this was to see if further improvement was possible, especially for the RGB network. Lastly, Figure 3.7 and Figure 3.10 show the confusion matrix of the validation data after the 60 epoch of training. Further discussion about these results will be in Chapter IV

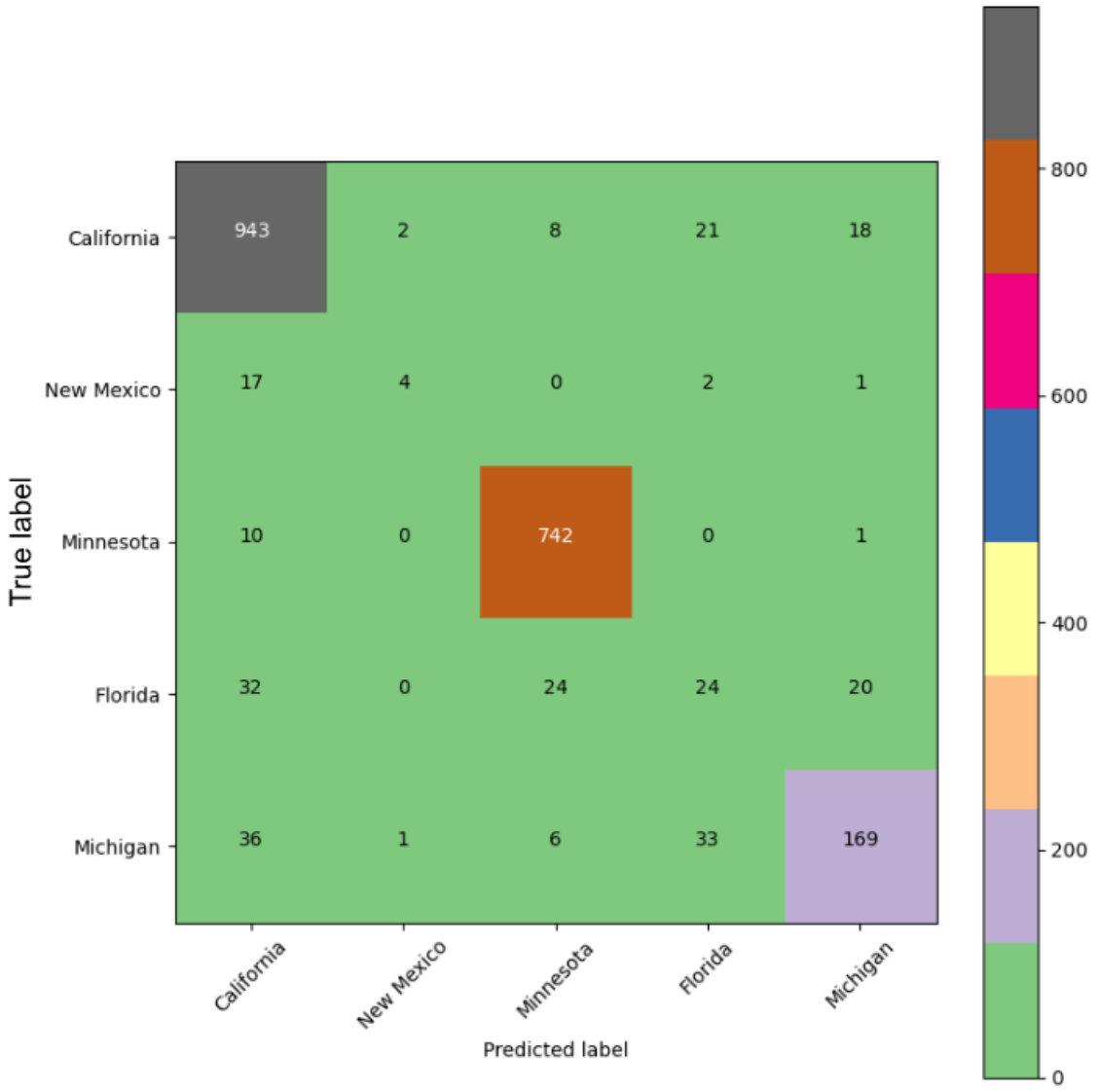

Figure 3.7: Confusion Matrix for validation for RGB and Beta values of 0.88,0.9999

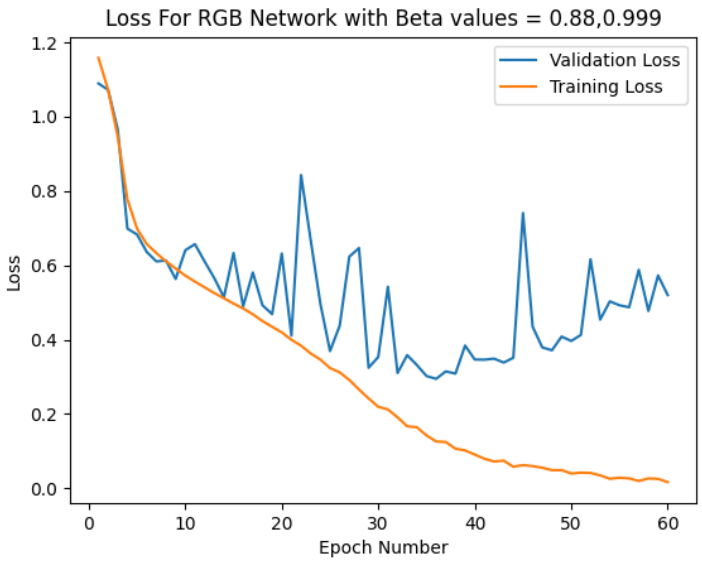

Figure 3.8: Learning Accuracy of RGB network with Beta values of 0.88,0.9999

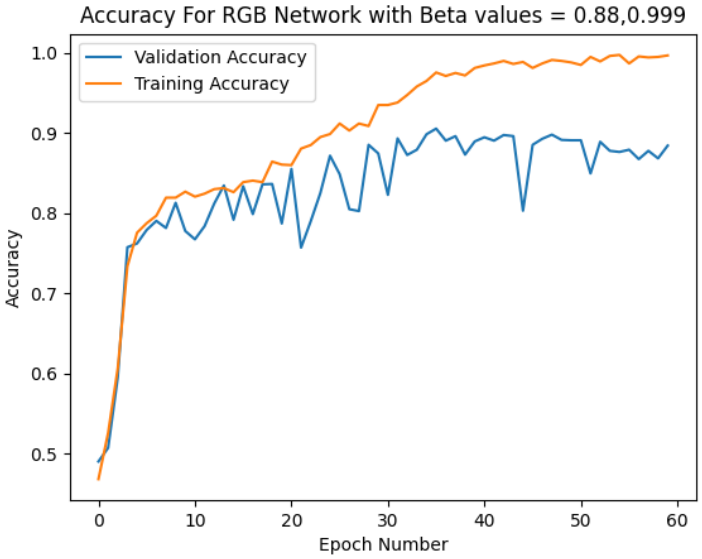

Figure 3.9: Learning Loss of RGB network with Beta values of 0.88,0.9999

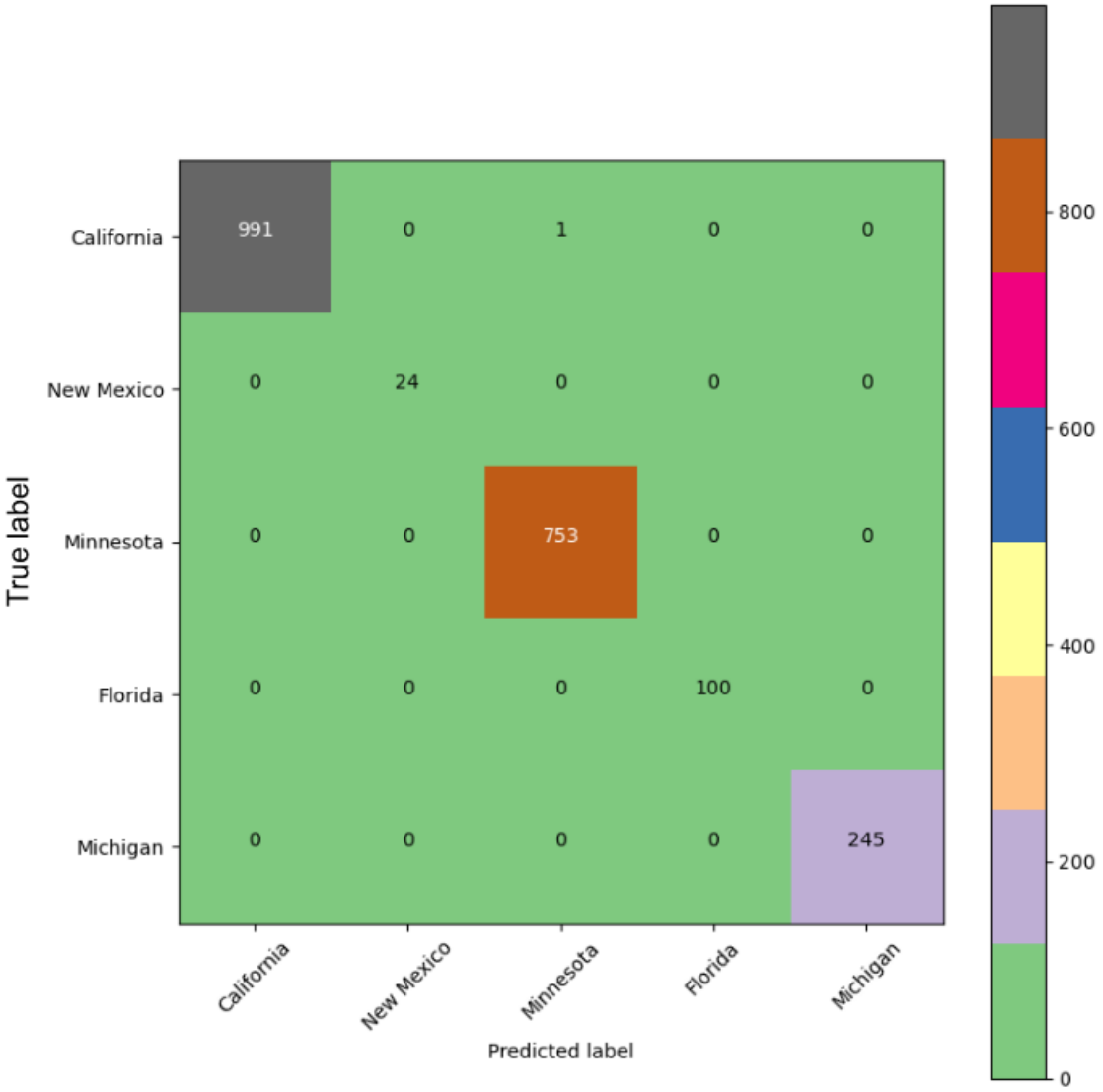

Figure 3.10: Confusion Matrix of hyperspectral network with Beta values of 0.88,0.9999

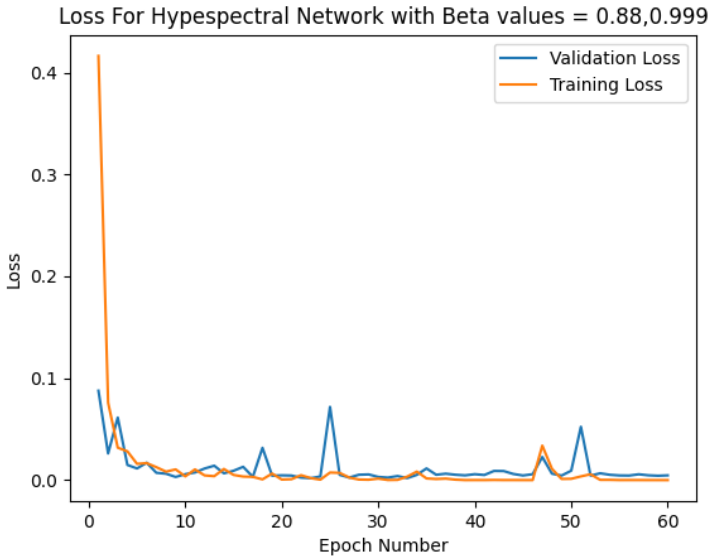

Figure 3.11: Learning Accuracy of hyperspectral network with Beta values of 0.88,0.9999

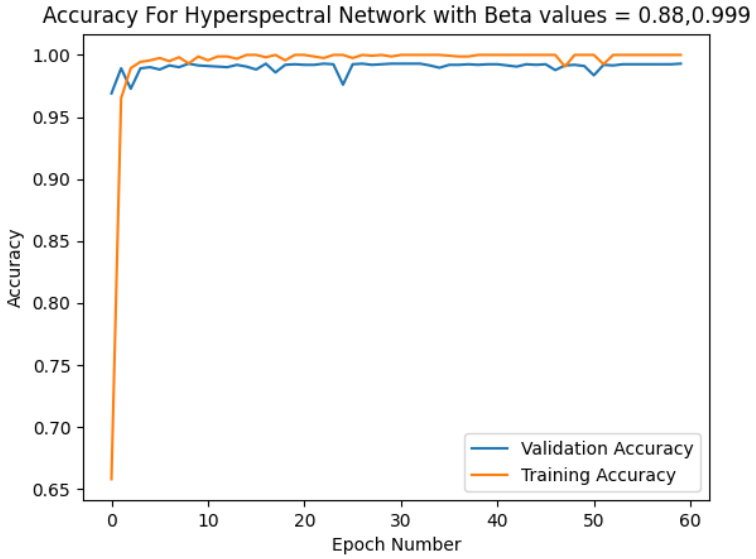

Figure 3.12: Learning Loss of hyperspectral network with Beta values of 0.88,0.9999

3.2 Same Size Test

Due to the size bias of the California, Minnesota, and Michigan data, another test was evaluated but using the same size data set for each class. This will remove the large bias in the training category allowing for a more even share of information. This will assume the best learning rate and beta values are the same from the full size experiment, which are learning rate of 0.00005 and beta values of 0.88 and 0.9999. Lastly, they were trained for 60 epoch as there is less data and will need more time to train.

The best validation accuracy and best validation loss for the RGB network was 0.60 and 0.71 respectively. The best validation accuracy and best validation loss for the hyperspectral network was 0.93 and 0.074, respectively. As in the previous experiment, there is a confusion matrix for the RGB and hyperspectral networks as seen in Figures **??** and 3.16, respectively. There are also the accuracy and loss graphs for RGB network in Figures **??** and **??**. Lastly, there are graphs for accuracy and loss for hyperspectral network in Figures 3.17 and 3.18.

Lastly, as the test results for the final test in both the all data test and even data test can be seen in 3.5. The main reason this final test is done is to show that the meta-variables were not over fitted to the training and validation data. These results where tested when the networks performed their best accuracy to extract the best results. That fact that these values show approximately the same to the values previously show that they were not over analyzed.

Table 3.5: Final Test Results

Test type	Network type	accuracy	loss
All	hyper	0.98	0.021
All	RGB	0.86	0.40
Even	hyper	0.91	0.095
Even	RGB	0.58	0.67

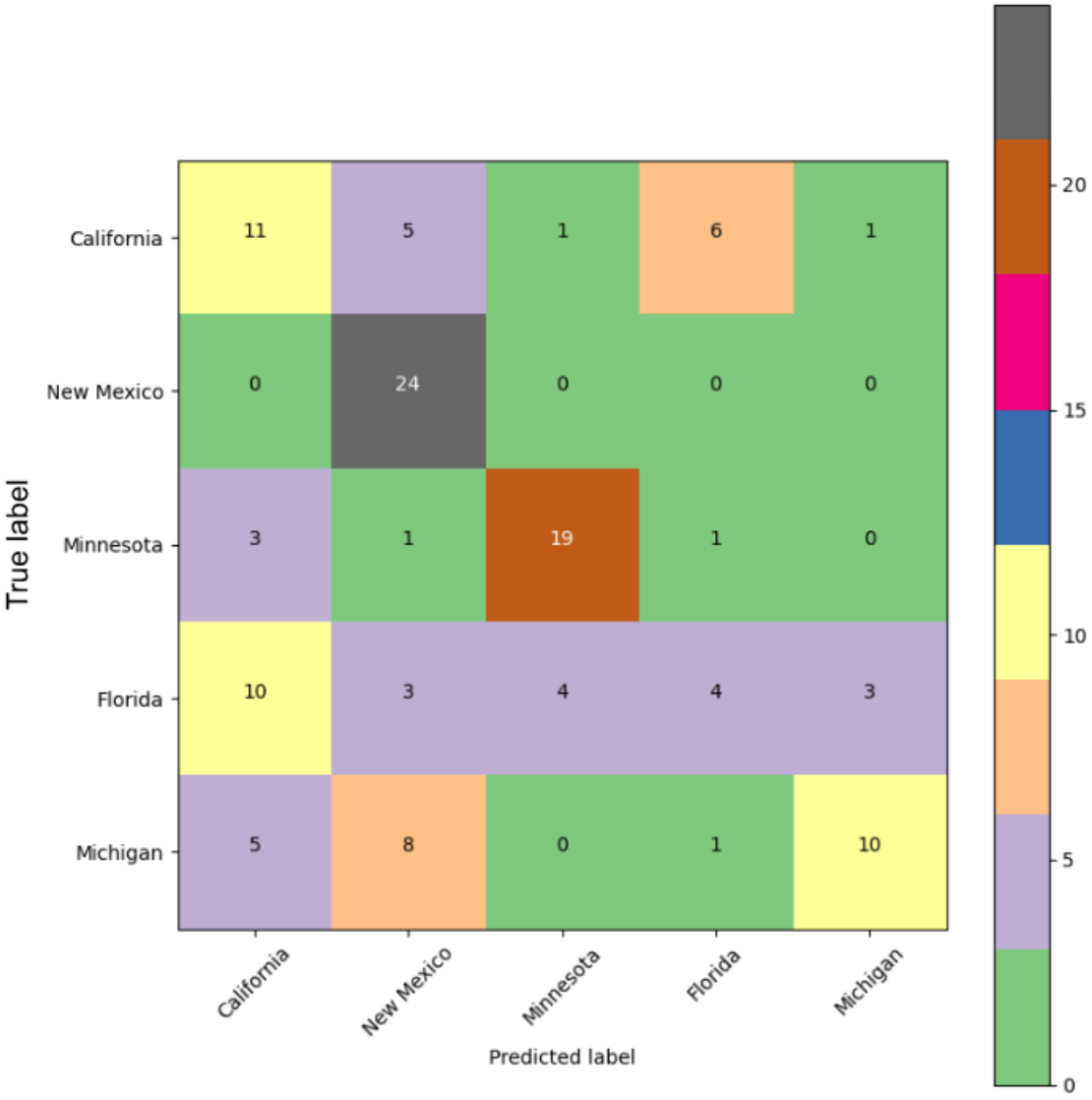

Figure 3.13: confusion Matrix for validation for RGB network of Same Size Test

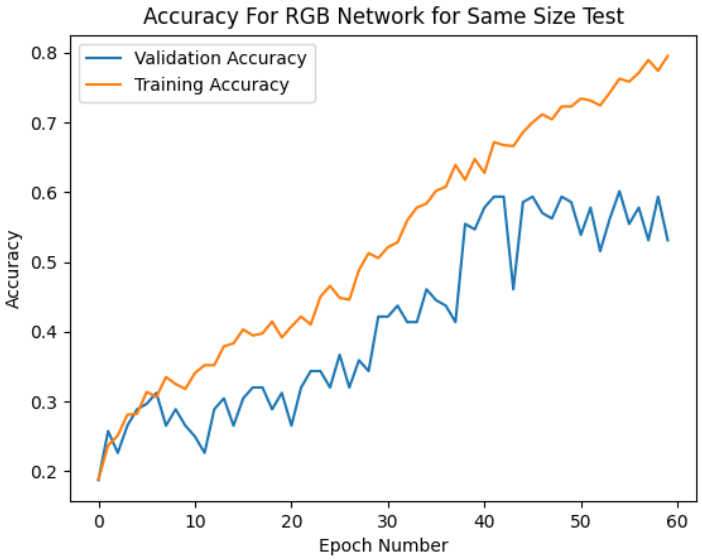

Figure 3.14: Learning Accuracy of RGB network with Beta values of Same Size Test

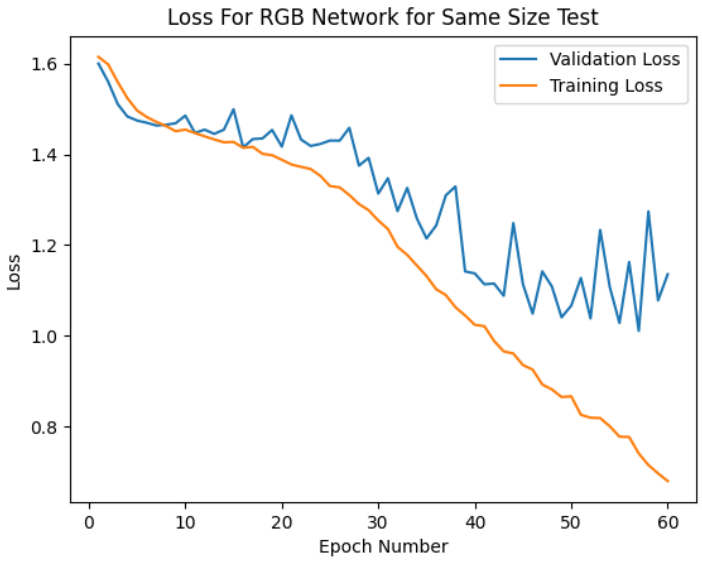

Figure 3.15: Learning Loss of RGB network with Beta values of Same Size Test

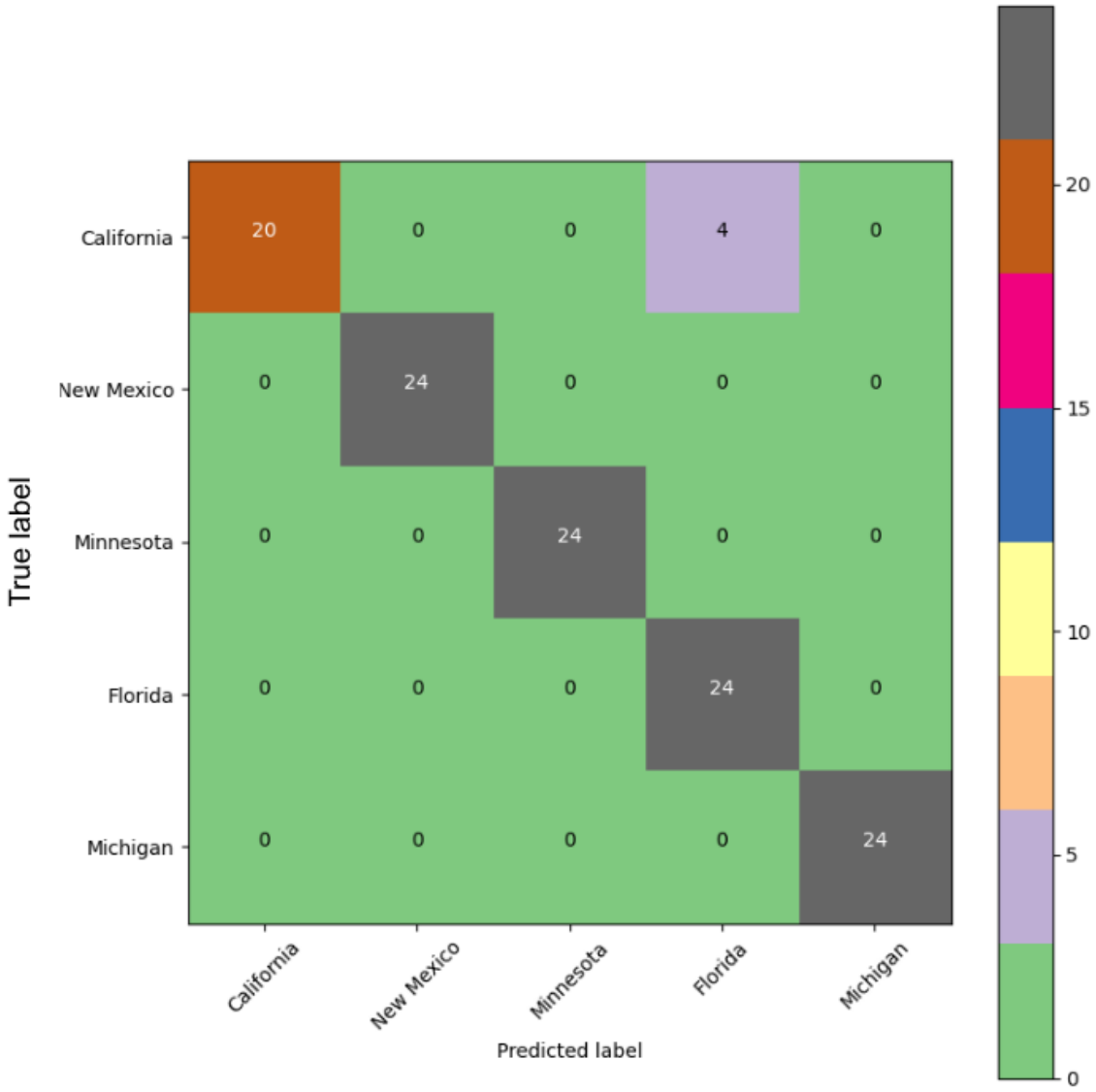

Figure 3.16: Confusion Matrix of hyperspectral network with Beta values of Same Size Test

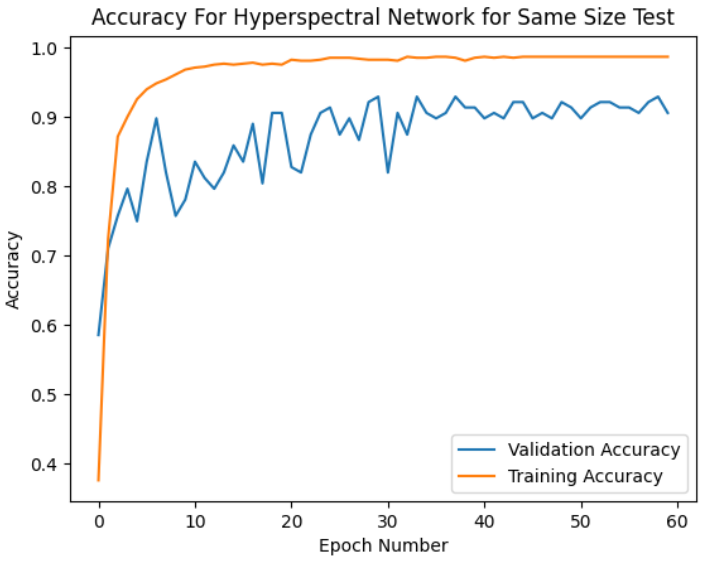

Figure 3.17: Learning Accuracy of hyperspectral network with Beta values of Same Size Test

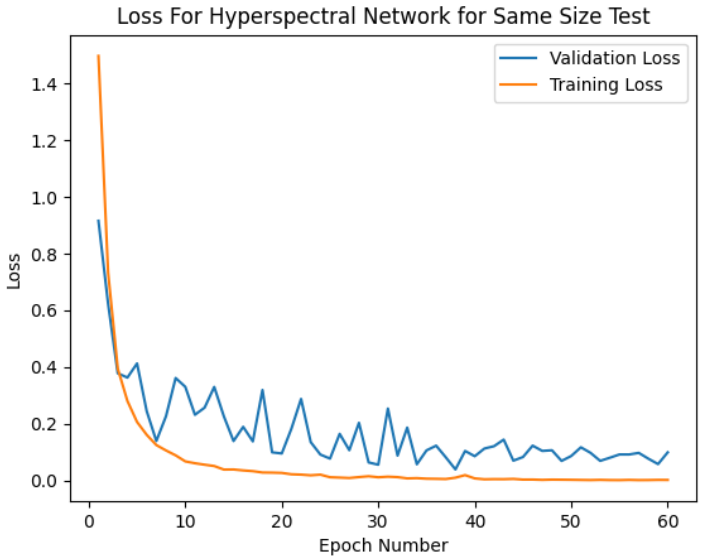

Figure 3.18: Learning Loss of hyperspectral network with Beta values of Same Size Test

CHAPTER IV

DISCUSSION

This section will discuss the results of the earlier section. This will go into detail about both the complete data test and the equal data test. It will also suggest ways in which this research could be improved and further areas in which we could investigate.

4.1 Complete data test

As noted from the complete data test, the hyperspectral network performed considerably better than the RGB network. The best hyperspectral network performed a maximum validation accuracy of 99%. This is wonderful but this doesn't show a complete picture. There is a lot of information in the hyperspectral imagery which some implicit patterns could arise based on the data. For example, in the Michigan and New Mexico image data was collected in the same week. With the sensitivity of hyperspectral images, they could possibly detect that on this week there was a spectral characteristic to know that it would go into one of those categories. For the RGB network, it performed with a maximum validation accuracy of 86%. This shows that the hyperspectral imagery has some benefit to classifying images as it increased the maximum performance by 13% even in this naive approach. The data shows that there is a behavior of over fitting for the RGB network. This idea is that the network is over conforming to the training data and such doesn't improve or even worsen when validating the data. Figure 3.9 and Figure 3.8 show the over fitting effect. There are several methods to reduce over fitting such as using drop out. Drop out is a method of randomly selecting nodes in the neural network to set to 0. This forces the network to use more than one method of categorizing data reducing over fitting[17]. An interesting aspect of the comparison between the networks is how quickly the hyperspectral

network reaches its highest accuracy/ lowest loss compared to the RGB network, as seen in figures 3.11, 3.12,3.8,and 3.9. The hyperspectral network achieves its peek in about 5 epochs while the RGB network took about 30 epochs. This is a 6x quicker learning as 1 epoch took approximately the same time to run. The last thing investigated was the worst performing images from the RGB network. The way I pulled the information was I kept a score for each image across each epoch. Then pulled the 3 worst performing in each class. Lastly I found their spectral responses from the saved data. The result of this process can be seen in Fig 4.1. The first image has the red spectral response, then green, then blue. From the spectral response of the worst performing images I could guess that most of the Wisconsin data has a lower values in the initial bump than any other category. But the neural network cannot be reduced to such simple classifications as this.

4.2 Even Data Test

The results of this section show that especially in a information poor environment the hyperspectal network performed considerably better than the RGB network. This is especially impressive that it was still able to properly categorize the images with larger previous data like the California data and the New Mexico data set. This means that it recognized some implicit pattern in that region and used that to determine images it didn't see before. This suggest an investigation into other similar ideas. Such as how much data of a region is needed to figure out where it is? or Perhaps it easily determine this data if the images were taken in different times of the year?

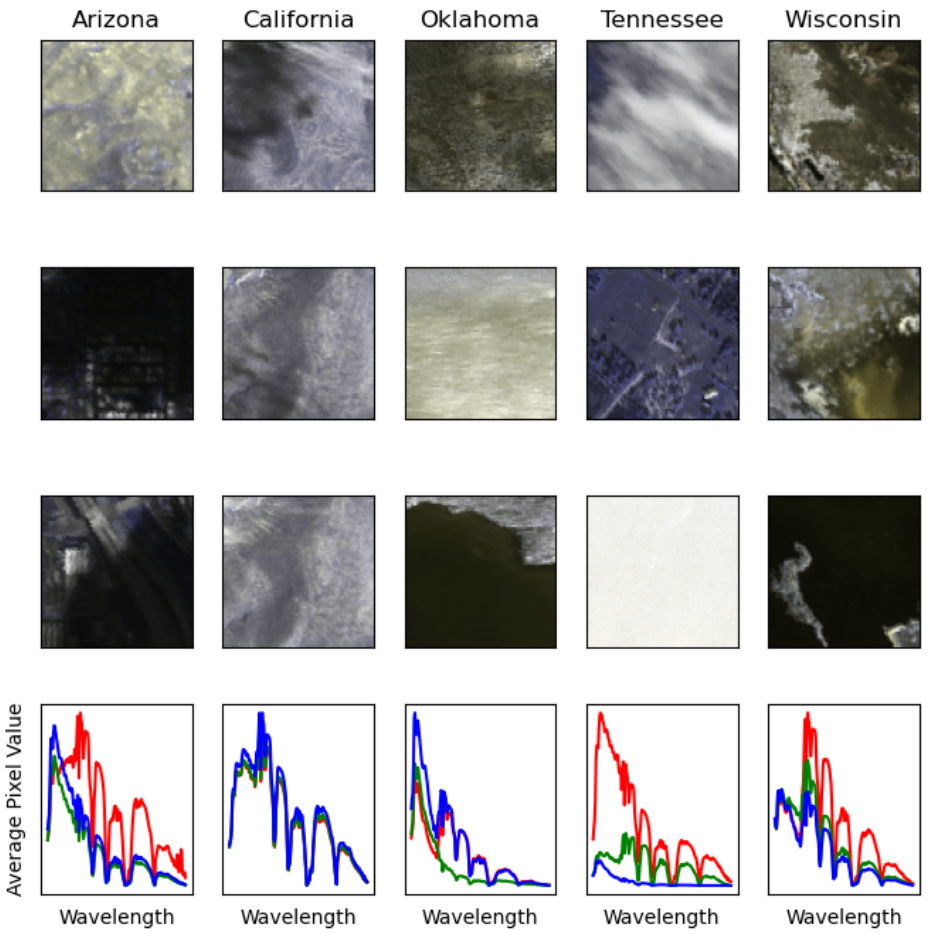

Figure 4.1: Worst performing images from the RGB full data test

4.3 Future Research

This section will discuss possible other ideas going from this project. These should be performed with the knowledge of this study and using some of the architecture that was created to preform these experiments. The easiest line of new information that should be investigated is what if the data used the larger pixel/meter as seen in Figure 2.1. There is another large set of substantial data that would be relativity easy to implement to see if there would be similar results with less detail in each of the images but with less detail. This would be interesting to see if the network can adapt to the larger GSD. Knowing that a network could adapt to the data would be useful. This would allow the use of this tool at a variety of altitudes as the GSD is highly dependent of the altitude. Another line of investigation directly from this research would be a more dynamic version of this where it would guess a longitude and latitudes location rather than locations. This would be beneficial in two major ways. Firstly, it would allow to use more data and there would be no manual labor in classifying the data as the longitude and latitude values would be available. Secondly, this would allow a more dynamic system and would be able to determine regions apart and then within regions apart. If this was possible it could be incredibly powerful in future defense research. This research would be more powerful if there was a more full data set of the locations it was investigating. This would mean one of two things if there was a process of collecting more data quickly. Which would be an incredibly tedious and expensive process. Or if there was a method of gathering aspects of hyperspectual data from RGB data which has already be collected across the world. Increase the liklyhood of investigations into the use of hyperspectral imagery. This would be a stretch as there is clearly much more data in the hyperspectral data that the RGB couldn't effectively extract all of the data.

4.4 Conclusion

This research shows a promising future in the use of hyperspectral imagery in research. There are many different areas that would need developed. This thesis shows that there are areas in which traditional RGB images are used that could have a great increase in accuracy if hyperspectral image data is used in combination. Or areas where hyperspectral imagery could provide more data than the Traditional RGB images would provide.

BIBLIOGRAPHY

[1] H. Lee and H. Kwon, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017.

[2] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal processing magazine*, vol. 19, no. 1, pp. 17–28, 2002.

[3] H. Grahn and P. Geladi, *Techniques and applications of hyperspectral image analysis*. John Wiley & Sons, 2007.

[4] "MS Windows NT kernel description," https://aviris.jpl.nasa.gov/dataportal/, accessed: 2022-09-30.

[5] R. V. Rossel and A. McBratney, "Laboratory evaluation of a proximal sensing technique for simultaneous measurement of soil clay and water content," *Geoderma*, vol. 85, no. 1, pp. 19–39, 1998.

[6] B. Mehlig, *Machine learning with neural networks: an introduction for scientists and engineers*. Cambridge University Press, 2021.

[7] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[9] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3365–3385, 2019.

[10] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected topics in applied earth observations and remote sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[11] B. Pan, Z. Shi, and X. Xu, "Mugnet: Deep learning for hyperspectral image classification using limited samples," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 108–119, 2018.

[12] L. Zhang, G.-S. Xia, T. Wu, L. Lin, and X. C. Tai, "Deep learning for remote sensing image understanding," 2016.

[13] B. Pan, Z. Shi, and X. Xu, "Hierarchical guidance filtering-based ensemble classification for hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 4177–4189, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] R. Bullock, "Great circle distances and bearings between two locations," *MDT, June*, vol. 5, 2007.

[16] K. K. Dobbin and R. M. Simon, "Optimally splitting cases for training and testing high dimensional classifiers," *BMC medical genomics*, vol. 4, no. 1, pp. 1–8, 2011.

[17] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.