12-10-2021

Evaluation of hyperspectral band selection techniques for real-time applications

Samantha Butler
Mississippi State University, samantha.tidrick@gmail.com

Evaluation of hyperspectral band selection techniques for real-time applications

By

Samantha Butler

Approved by:

Ali Gurbuz (Major Professor)
John E. Ball
Mehmet Kurum
Stanton R. Price
Jenny Du (Graduate Coordinator)
Jason M. Keith (Dean, Bagley College of Engineering)

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Electrical and Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

December 2021

OPSEC #:

Name: Samantha Butler

Date of Degree: December 10, 2021

Institution: Mississippi State University

Major Field: Electrical and Computer Engineering

Major Professor: Ali Gurbuz

Title of Study: Evaluation of hyperspectral band selection techniques for real-time applications

Pages of Study: 56

Candidate for Degree of Master of Science

Processing hyperspectral image data can be computationally expensive and difficult to employ for real-time applications due to its extensive spatial and spectral information. Further, applications in which computational resources may be limited can be hindered by the volume of data that is common with airborne hyperspectral image data. This paper proposes utilizing band selection to down-select the number of spectral bands to consider for a given classification task such that classification can be done at the edge. Specifically, we consider the following state of the art band selection techniques: Fast Volume-Gradient-based Band Selection (VGBS), Improved Sparse Subspace Clustering (ISSC), Maximum-Variance Principal Component Analysis (MVPCA), and Normalized Cut Optimal Clustering MVPCA (NC-OC-MVPCA), to investigate their feasibility at identifying discriminative bands such that classification performance is not drastically hindered. This would greatly benefit applications where time-sensitive solutions are needed to ensure optimal outcomes. In this research, an NVIDIA AGX Xavier module is used as the edge device to run

trained models on as a simulated deployed unmanned aerial system. Performance of the proposed approach is measured in terms of classification accuracy and run time.

Key words: hyperspectral imagery, band selection, hyperspectral at the edge

ACKNOWLEDGEMENTS

This thesis would not have been possible without the guidance and encouragement of my committee, professors, mentors, and family, so it is with gratefulness that I write this page.

I would first like to express my appreciation for my advisor, Dr. Ali Gurbuz, for his support through out this entire process. His knowledge of the field and astute suggestions have greatly shaped this thesis.

I would also like to give my gratitude to my committee for all of their helpful comments and assistance through out this process. I want to extend an extra thank you to Dr. Stanton Price for his consistent feedback and intuitiveness.

I would also like to thank my family and friends for their prayers and encouragement through out my Masters experience. I especially want to express my appreciation for my husband, Ethan Butler. His support is everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE

ANN Artificial Neural Network

ACOBS Ant Colony Optimization-based Band Selection

AVRIS Airborne Visible/Infrared Imaging Spectrometer

CNN Convolutional Neural Network

CIG Computational Intelligence Group

CPU Central Processing Unit

E-FDPC Enhanced Fast Density-Peaked Based Clustering

GPU Graphics Processing Unit

HYDICE Hyperspectral Digital Imagery Collection Experiment

ISSC Improved Sparse Subspace Clustering

LP Linear Prediction

MVPCA Maximum-Variance Principal Component Analysis

NC-OC-MVPCA Normalized Cut Optimal Clustering MVPCA

NVIDIA NVIDIA Jetson AGX Xavier

PCA Principal Component Analysis

PSO Particle Swarm Optimization

SVM Support Vector Machine

UAV Unmanned Aerial Vehicle

VGBS Fast Volume Gradient Band Selection

CHAPTER I

INTRODUCTION

1.1 Purpose of the Research

Hyperspectral image data can provide unique spectral information on collected scenes and has been proven to be helpful in numerous fields, such as in the fields of food quality inspection [16], [32], [35], vegetation and agriculture [30], [48], medical studies and diagnosis [26], [29], and military applications [2], [27], [34], [41]. The resulting data is three dimensional, in which the components are pixels by pixels by spectral bands. It can be made of hundreds of bands that span the regions of visible and infrared within the electromagnetic spectrum, within spatial resolutions of a particular value, that provide important information of the earth's surface [1]. However, the resources are not always readily available to process its rich information, and the hyperspectral image data can have a tendency to contain repetitious information. To counter this, the task of dimensionality reduction has been introduced for the creation of a new, smaller hyperspectral image set. Typically the methods of reducing the hyperspectral image data high dimensionality can fall within the categories of feature extraction or feature selection [37]. Feature extraction works to create a reduced hyperspectral image data set by mathematically transforming the original image data into a smaller set of combined variables [9]. Band selection seeks to create a reduced hyperspectral image data set by determining which spectral bands provide the most information, selecting a predetermined number of those elucidative bands, and removing the remaining spectral bands from the set. This is done with the expectation that the descriptive and overall general information given by the hyperspectral image data will still remain as informative with less processing requirements. Within this study, band selection is applied on the hyperspectral image data and then tested with classification models for performance evaluation. Classification of hyperspectral image data is done to identify specific components of the earth's surface, with each of the pixels representing a certain characteristic [45]. This thesis seeks to compare the performance and effectiveness of these band selection techniques when running on a simulation device and a device acting as an edge device.

1.2 Contributions of this Research

To the authors' knowledge, there has not been a presented study done in which the classification performance of the VGBS, ISSC, MVPCA, and NC-OC-MVPCA are reviewed on an edge device, e.g. the NVIDIA Jetson AGX Xavier, for the purposes of studying the effects of this use for use in real-time applications. The research also provides insight into cross-platform interaction with the building, placing, and testing of the classification models. This thesis provides a foundation for continued work with band selection for those classification models and show how the NVIDIA Jetson AGX Xavier is useful for hyperspectral sensor processing.

1.3 Outline of the Thesis

The thesis is outlined as followed: Chapter II introduces the hyperspectral image data usefulness as well as its disadvantages. It provides band selection techniques as a method for reducing data acquisition and computational strain, and it includes classification models to show their effectiveness. The chapter closes by discussing the background work and existing literature in this area. Chapter III describes the testing and results for the first implementations of this process that were done. Chapter IV is an implementation of the previous testing on MATLAB, but the selected band values are lower values and the training models are also tested on the NVIDIA Jetson AGX Xavier. The final Chapter V gives a review of the results obtained from the thesis, derives conclusions, and provides input for future work.

CHAPTER II

BACKGROUND

2.1 Hyperspectral Classification

Hyperspectral sensors capture image data rich with spectral information that can be helpful in diverse range of fields. Two hyperspectral sensors that are currently in use are the Airborne Visible/Infrared Imaging Spectrometer (AVRIS) and the Hyperspectral Digital Imagery Collection Experiment (HYDICE) [6]. Figure 2.1 from [31] provides a well illustrated concept of how hyperspectral imaging works with the multiple wavelengths to extract details from the Earth's surface being represented by the multiple layers and colors. The problem with using this 3D image data is that it can often be computationally expensive to acquire and to use, can frequently require longer processing times, and usually contains redundant information. This causes it to not be realistic for applications in which these resources are limited, such as use at the edge with unmanned AI devices, like unmanned aerial vehicles (UAV). In response to these disadvantages, there has been research done to reduce the overall volume of the hyperspectral image data, with the most common of these approaches typically falling within the realm of feature extraction or feature selection, e.g., band selection. Feature extraction works to create a reduced hyperspectral image data set by mathematically transforming the original image data into a smaller set of combined variables. An issue that occurs with using this approach is that important spectral information can be removed or altered from the original set. Most notably, the discriminability of the hyperspectral

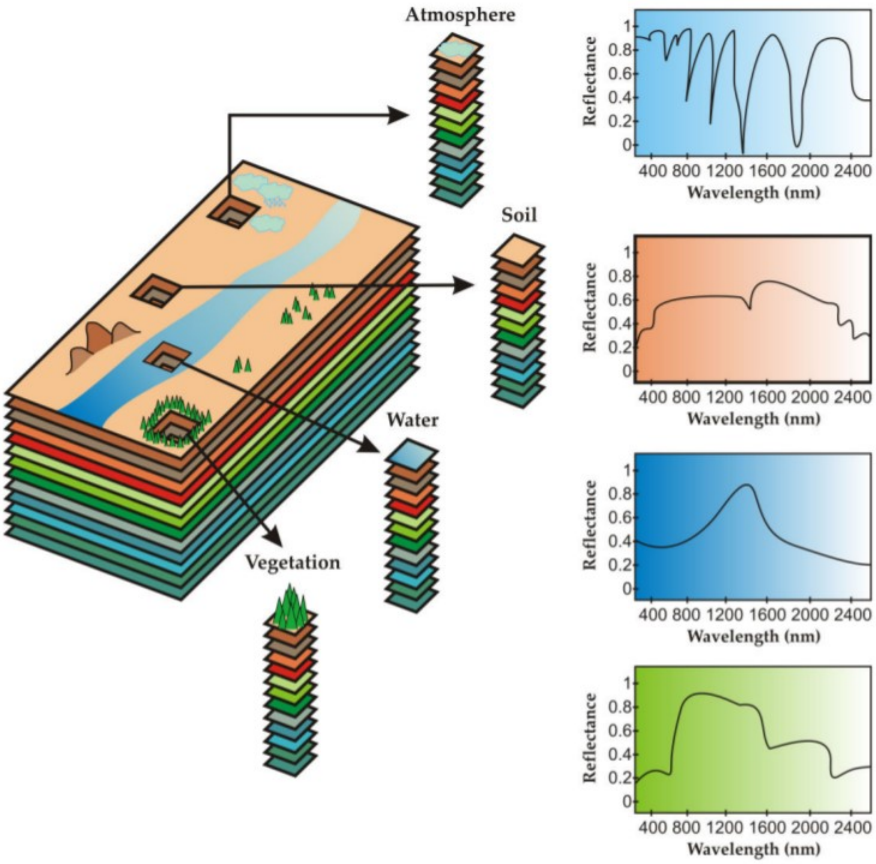


Figure 2.1: Hyperspectral Imaging Illustrated Example [31]

image dataset can be lost with these tests. Band selection retains the original data set, and instead focuses on selecting a lower amount of spectral bands without degrading classification performance much.

2.2 Overview of Band Selection Techniques

Work done with band selection techniques has ranged from selecting bands based off of distance measurements, such as the Euclidean distance or Jeffries-Matusita distance [19], to selecting bands using a clustering technique, such as *k*-means clustering [13] or affinity propagation-based clustering [21]. With the diverse nature of band selection techniques, there are numerous categories that it can be divided into, with the categories being focused on a particular type of learning or process in which these methods seek to find the optimal band set solutions. In Ref. [38], an in-depth overview of some of the most popular band selection techniques are listed, and within this paper, the majority of these techniques can be placed within one of the following six categories:

- ranking-based
- searching-based
- clustering-based
- sparsity-based
- embedded learning-based
- hybrid scheme-based methods

Ranking-based band selection techniques hierarchically sort the spectral bands by an importance criterion [7], [22]. The bands that are the highest ranked are the ones selected from the down-selected hyperspectral input matrix. The principal component analysis (PCA) is a ranking-based method that used the variances of the bands to determine the most informative of bands from the data. The Maximum-Variance Principal Component Analysis PCA (MVPCA) sorts the bands by their variance values, with the lower variance ranking higher than the higher variances. The ranking is done this way due to the idea that bands with the lowest variances are more likely the most dissimilar from one another and that bands with high variances can tend to have redundant information shared between them [7]. Another ranking based technique is the Enhanced Fast Density-Peaked Based Clustering (E-FDPC) method [22], which selects the bands by calculating their local density and intracluster distance and then gives these calculations a certain weighted value. Figure 2.2 shows an example of how this method chooses these bands. Determining the band

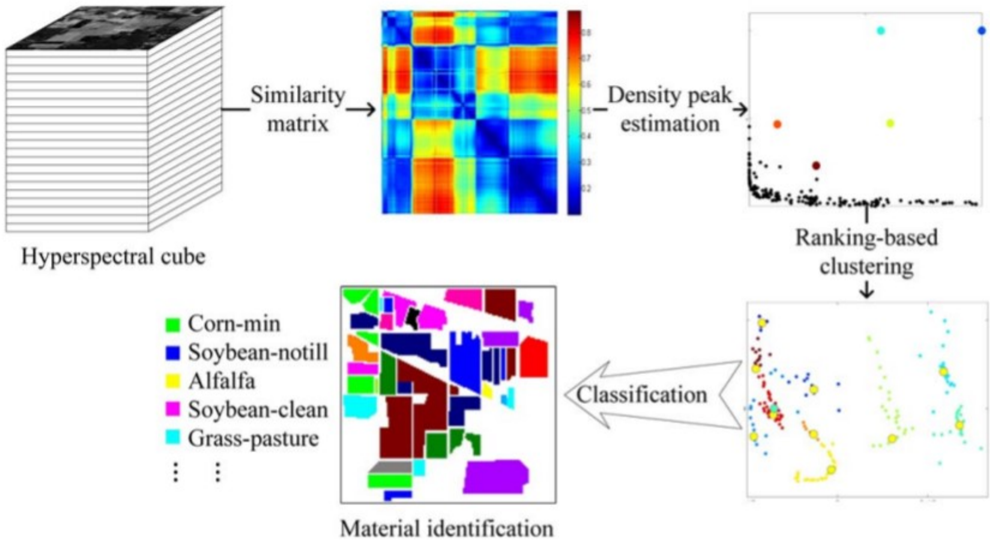


Figure 2.2: Overview of E-FDPC [22]

subset using a searching based method requires a defined criterion function, which defines what makes a solution the best solution, and requires a searching strategy, which determines how the strategy will seek for the optimal sets [10], [36]. Some examples of searching based methods are the linear prediction (LP) and particle swarm optimization (PSO). The LP band selection method chooses the bands to represent the hyperspectral image data by using the spectral bands that are the most unique [10]. The PSO algorithm is a heuristic learning algorithm that seeks to find the best solution by a series of algorithmic calculations that are iteratively updated. These calculations were created with inspiration from that of bird flocks, with the potential solutions treated as the moving parts and the location of the best know position in the subspace being chosen [36].

The clustering approach for band selection separates the hyperspectral image set into groups, or clusters, determined by a similarity metric, and a band from each of the clusters that best represents that group is the one chosen for the final band set. The use of Ward's linkage in [28] is an example of how clustering can be applied for band selection. In this application, the bands were grouped in attempts to get the lowest intracluster variance and highest intrercluster variance. Sparsity-based methods are developed with the notion that all of the bands of the hyperspectral image data can be sparsely represented in such a way that new information of the image data can be identified. The improved sparse subspace clustering (ISSC) model is a type of sparsity-based method for band selection in which the bands are chosen when spectral clustering is performed on the previously constructed similarity matrix. More on this technique will be provided in Section 3.1 [40]. Embedded learning-based methods uses a defined implementation, such as embedded learning or classification tasks, to be the basis of which the bands are selected. In these types of band selection methods, the bands are chosen that give the optimal solution to the problem given. An example of this method would be the use of a convolutional neural network (CNN) for the task of classification of the hyperspectral imagery, in which the CNN is used to extract deep features from the set and then used on AdaBoost SVM classifier for choosing the most unique and informative spectral bands. Band selection methods categorized as hybrid scheme-based methods combine multiple methods to find the optimal band set. An example of this method is [24] when the genetic algorithm and support vector machine were both used to find the bands that gave the best classification accuracy. Mutual information was also used in this selection to help reduce the use of highly correlated bands.

2.3 Classification Models

To test the effectiveness of the band selection techniques, the task of classification is used. In this, the reduced data sets are applied to a particular classification model, which in turn seeks to evaluate the different components of the scene within the image data. This was chosen as the test due to its reflection of a real-time use with hyperspectral image data. There are numerous classification models available, but this thesis focuses on the use of an artificial neural network (ANN) and a support vector machine (SVM). These models were selected because of their high achieving results from various related hyperspectral band selection studies, such as in [33], [46], [38].

An ANN is a type of machine learning structure that was developed using concepts that relay to the human neural system with how its components interact. Just as the neural system is composed of several neurons that signal to one another, the ANN is composed of layers with neurons that receive inputs and send outputs to communicate with the other neurons. An ANN are made up of three types of layers: input layer, hidden layer, and output layer. The neurons within the layers are what connect the layers together. Each neuron has a numeric weight, which can be a fixed value or change with iterations, that determines its contribution to the ANN structure.

Figure 2.3 shows an example of a simple ANN.

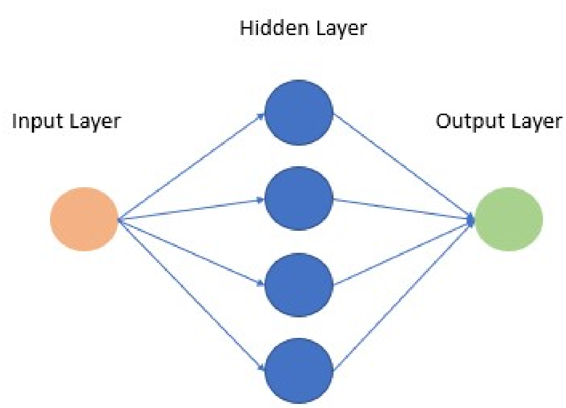


Figure 2.3: Example of a simple ANN structure

An advantage to using ANNs is that there are numerous studies in which they have shown great performance and classification abilities. However, ANNs tend to require large amounts of data for training and testing its models, with some ANNs requiring at least thousands of inputs to even receive passible responses. This classification model can also be computationally expensive depending on its makeup, with more complex networks needing more computational power.

The SVM is a supervised machine learning classification model that uses algorithmic calculations to determine the labels of the given data. It takes labeled training data and its corresponding classification labels and creates a model that places the given data points into one of the classification labels. SVM does this by dividing the given data into subcategories separated by a hyperplane, also known as the decision boundary. The hyperplane is said to be its best when it is able to correctly identify majority of the data to its respective class. SVM, though first introduced for binary classification [8], has been extended to include multi-class classification capabilities. Two approaches that can be used for multi-class classification are the "one vs. all" and "one vs. one" techniques. The "one vs. all" technique takes all of the dataset and separates it into subsets that place each classifier in comparison with the remaining classifiers. The "one vs. one" approach goes through each of the individual categories and has them tested against all of the other categories [18]. Some advantages to using this classification model are that it is able to perform well in high dimensional spaces, even when the dimensional space is much larger than the given sample number. A few drawbacks to using SVM are that it can have a tendency to overfit when the number of features is greater than the number of samples and that it requires cross-validation for probability estimates. Further, the choice of the test kernel is often unknown a priori.

2.4 At The Edge Device

For this thesis, "at the edge" refers to the processing of the data on the device in the moment of collection. In some applications of data collection and processing, the data must be sent back and forth between a central station and the device. However, with this study, the expectation is that once the device has been trained, it can receive data for classification and determine its usefulness all within the device. If the information is determined to be useful, it can then be transferred back to the central station.

Figure 2.4: NVIDIA AGX Xavier Developer Kit and Standalone Module [12].

The NVIDIA Jetson AGX Xavier developer kit, as seen in Figure 2.4, is an AI computer that has a 512-core NVIDIA Graphics Processing Unit (GPU) with an 8-core ARM. Its module can run with a minimum power supply of 10W, has 32GB 256-bit LPDDR4x memory, and is only 105 mm x 105 mm, which makes it desirable for use on AI-powered autonomous machines, especially those such as unmanned aerial devices where weight and storage can be constrained and substantial for performance capabilities. Further, the Jetson AGX Xavier developer kit has been optimized for

AI-powered autonomous machines with its GPU performance of 32TOPS capability and NVIDIA AI tools enhanced for deep learning. Table 2.1 shows the attributes of the Jetson AGX Xavier module compared to other modules on the market. These attributes of the NVIDIA are the reason it was selected to act as the device for the at the edge results.

Table 2.1: Similar Devices and Specifications

Module	NVIDIA Jetson AGX Xavier	NVIDIA Jetson Xavier NX	NVIDIA Jetson Nano
Storage	32GB eMMC 5.1	16 GB eMMC 5.1	microSD capablities
Size	105mm x 105mm	45mm x 69.6mm	69.6mm x 45mm
GPU	512-core Volta GPU 64 Tensor cores 22 TOPS (INT8)	384-core Volta 48 Tensor cores	NVIDIA Maxwell 128 NVIDIA CUDA cores
CPU	8-core Carmel Arm v8.2 64-bit CPU 8MB L2 + 4MB L3	6-core Carmel Arm v8.2 64-bit CPU 6MB L2 + 4MB L3	Quad-core ARM Cortex-A57 MPCore processor
Memory	32GB 256-bit LPDDR4x 136.5GB/s	8 GB 128-bit LPDDR4x 1866MHz 59.7GB/s	4 GB 64-bit LPDDR4 1600MHz 25.6 GB/s

2.5 Related Work with Real-Time Applications

The study of hyperspectral image data has been a well-researched topic. Within this realm, there has also been a growing number of studies done with the intentions of providing research in which band selection techniques are applied to a hyperspectral image data set and then tested for real-time applications. The following section provides an overview of some similar work that has been done.

In [47], there was an experiment ran in which the band selection techniques were trained on a central processing unit (CPU) device and then implemented onto a NVIDIA GTX285 GPU system using C++. This research highlights the use of unsupervised band selection techniques for reducing the dimensions of hyperspectral image data, with a specific focus on endmember extraction algorithms. The algorithm selected for this study was the LP algorithm and GPUs were used to help process the algorithms due to its low computational cost even with high computational processing. The GPU was able to optimize the CUDA CUBLAS library to implement the algorithmic calculations. Figure 2.5 provides a block diagram of how the LP-based band selection algorithm was used with the GPU.

There has been similar experimentation done in which the band selection algorithm, the PSO algorithm, is initiated on a CPU device and then placed onto a GPU device to train and test the device. Its results are then printed back on the CPU [44]. Figure 2.6 shows the implementation of this placing on the model for training and testing.

Within [14], the Ant Colony Optimization-based Band Selection (ACOBS) algorithm, which builds off of the original ant colony optimization algorithm, was used to select the best band set. For the purposes of real-time comparisons, an NVIDIA GPU is employed to demonstrate real-time performance. The device was optimized to run in parallel, so the individual tasks of the algorithm were able to run at higher speeds. Figure 2.7 shows this experimentation.

A study was done in [42] in which the focus of effectively using hyperspectral image data for real-time devices shifted from the use of band selection techniques to finding an efficient method

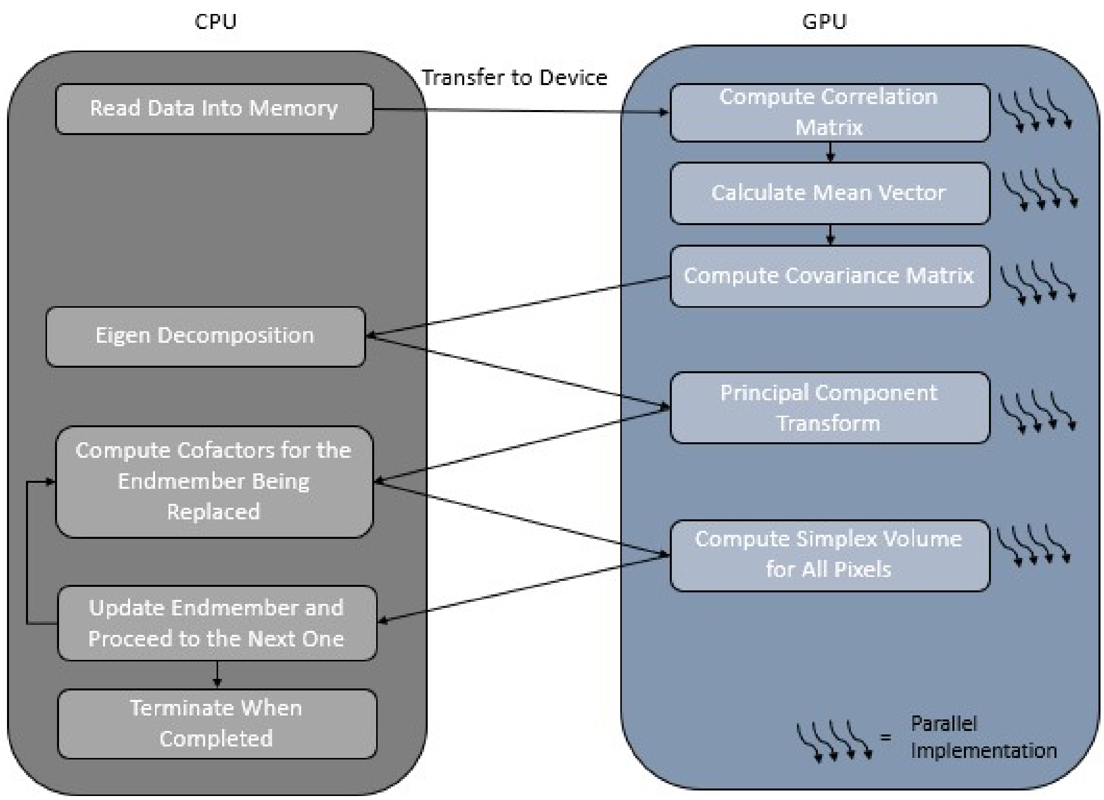


Figure 2.5: Parallel LP Band Selection Algorithm [47]

for spectral unmixing of the data set. Spectral unmixing is a method of estimating the endmembers, which are the distinctive attributes of the hyperspectral data cube. In this paper, an NVIDIA device is optimized to effectively run both its CPU and GPU components for a more efficient real-time analyzation of the data.

Another study was done in which the orthogonal subspace projection (OSP) algorithm was used for the task of automatic target detection. The implementation expanded to include a OSP cluster version and a OSP GPU version, and the target detection performance was compared between the two methods [31]. The clustering based OSP provided was beneficial for applications in which the data had already been transmitted and received, but the GPU based OSP allowed for close to real-

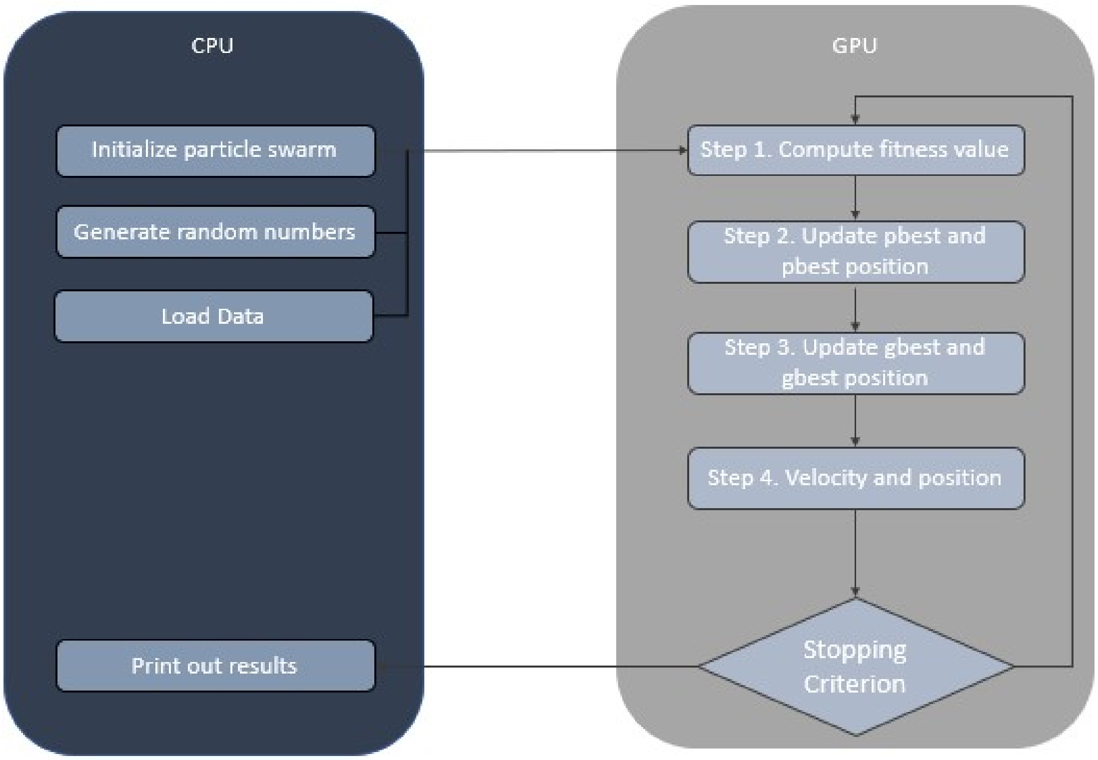


Figure 2.6: Overview of CPU-GPU System for the PSO-Band Selection [44].

time collection and anamoly detection. For this research, the studied show that the classification accuracy for both techniques were the same values, so the comparison was done for the calculation run times.

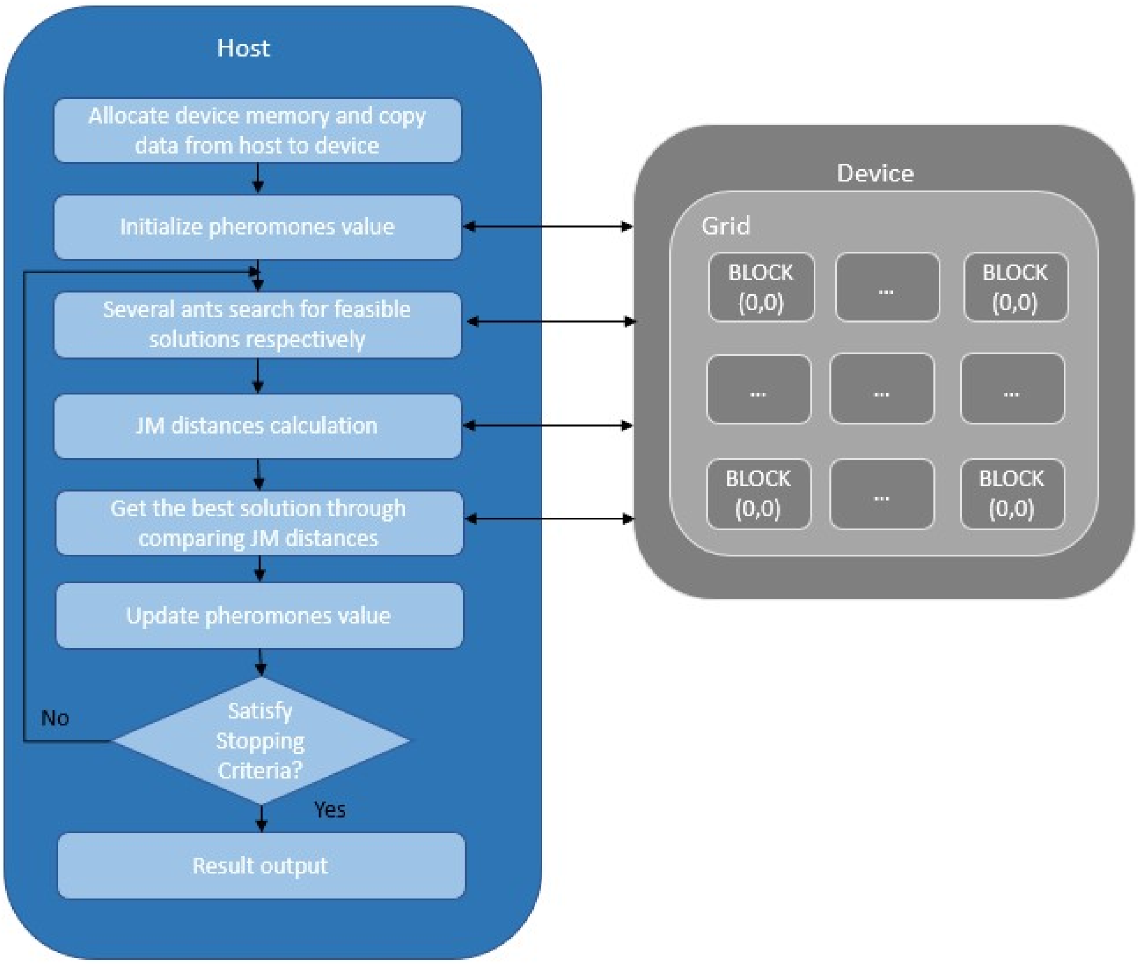


Figure 2.7: Overview of the ACOBS CPU-GPU Implementation [14]

CHAPTER III

OVERVIEW OF BAND SELECTION PROCESSING PIPELINE

In this section, unsupervised band selection techniques are used on the input hyperspectral image data to get a resulting reduced band selected hyperspectral image data. Being unsupervised makes the band selection techniques appealing candidates because it promotes a more generalizable set of bands to be selected. That is, incorporating the classification performance into the strategy for band selection identification is susceptible to being biased towards the data distribution used by the supervised band selection algorithm during the learning/band selection phase. Thus, unsupervised techniques have the appeal of not requiring labeled data/ground truth information available to be applicable. While using supervised band selection techniques can be a highly effective strategy for situations in which the training data available adequately represents a specific domain that is of interest, the identified bands would likely be less robust when transferred to a new area/region of interest. The following band selection approaches are discussed: Fast Volume-Gradient-based Band Selection (VGBS) [15], Improved Sparse Subspace Clustering (ISSC) [40], Maximum-Variance Principal Component Analysis (MVPCA) [7], and Normalized Cut Optimal Clustering MVPCA (NC-OC-MVPCA) [43]. In this preliminary section, the focus will be on the initial testing of these band selection techniques on MATLAB, as well as testing a varying amount of bands for selection.

3.1 Band Selection Techniques

The band selection techniques chosen for this thesis research were selected due to their promising results from their respective research papers and for their reproducibility.

3.1.1 Fast Volume Gradient Band Selection

The VGBS method [15] is an unsupervised technique that selects bands on the basis of their volume gradient. It builds off the structure of the original VGBS algorithm, and it is considered a greedy-based method due to its function of taking or removing a band from the prospective band set to determine its effect on optimizes the objective function. Instead of calculating the volume of each band in the entire set, the VGBS calculates the volume gradient of the whole parallelotope. It then removes those bands that pertain to the largest gradient values. With the removal of these bands, there is likely to be minimal loss in spectral information, since the volume of the simplex is typically most effected by the redundant bands. To calculate the volume of the simplex, use

$$V_m = \frac{1}{(m-1)} V_{m-1} h \tag{3.1}$$

where m represents the number of bands within the set, V_m is the volume of the simplex, V_{m-1} is the volume of the subsimplex, and h represents height between the remaining band and the subsimplex. The derivative is then taken from both sides. This equation can be further reduced since the number of bands selected is chosen before calculation and thus remains the same throughout the calculation. The resulting equation can be shown as

$$V_{m-1} = (m-1)\frac{dV_m}{dh} \tag{3.2}$$

This equation shows that the bands can be selected based on the gradient of the simplex volume over the calculation of all of the subvolumes, since the subvolume is of the same proportion to that of the gradients of the remaining bands' volume. A disadvantage to the VGBS is that it can be sensitive to noise.

3.1.2 Improved Sparse Subspace Clustering

The ISSC [40] technique seeks to select bands building off of the concepts used in sparse subspace clustering. Within sparse subspace clustering, the method acts off the notion that the entire data set can be fragmented into smaller components, i.e., subspaces, that separate the data points into the subspaces by similarity. The original sparse subspace clutering method uses the L_1-norm problem to create the sparse coefficient vectors that represent the band vectors [11]. The ISSC method differs from the original algorithm by using the L_2-norm optimization problem instead of the L_1-norm optimization method. Each band vector can be defined by the following:

$$y_i = YZ_i + e \quad Z_{ii} = 0 \tag{3.3}$$

where y_i is the band vector, $Z_i = [Z_i, 1Z_i, 2...Z_i, N]^T$ is the coefficient band vector of y_i, and e is the error term with a bounded norm. The equation that encompasses all of the band vectors is represented as such:

$$Y = YZ + E, \quad diag(Z) = 0 \tag{3.4}$$

where Z and E are the matrix representations of the sparse coefficients and errors of all of the band vectors. Using the L_2-norm problem, (equation) can be optimized using this equation:

$$\hat{Z} = argmin||Y - YZ||_2^2 + \beta||Z||_2^2 \tag{3.5}$$

where $||Z||_2$ represents the L_2-norm. To solve the above-mentioned equation, the ISSC algorithm uses the least square regression algorithm. The resulting sparse coefficient matrix $\hat{Z}$ is then used to create the similarity matrix W, which is constructed using the following:

$$W_{AS_{ij}} = (\frac{\hat{Z}_i \cdot \hat{Z}_j}{||\hat{Z}_i||^2 \times ||\hat{Z}_j||^2})^2$$

The following algorithm is then used to create the distribution compactness (DC) plot, which was designed to help reduce errors by mathematically calculating the needed size of the band subset.

$$DC = \int_{\hat{Z}} p(\hat{Z}) d\hat{Z} \approx 1_N^T W 1_N = \sum_{i=1}^{N} \lambda_i \{1_N^T u_i\}^2$$

The $log(\lambda_i \{1_N^T u_i\}^2)$ is used to choose the appropriate cluster numbers. Then, the row vectors of the band with the closest to the mean vector is chosen to represent the cluster.

3.1.3 Maximum-Variance Principal Component Analysis

The MVPCA technique [7] is a ranking-based approach that ranks the spectral bands according to their variances. A covariance matrix is first computed from the given hyperspectral image data. Then, a loading factor matrix is created by doing eigenvalue decomposition over the covariance matrix. This loading factor matrix is what determines the ranking of the spectral bands, with higher variances being ranked over those with lower variances. Disadvantages to using this method are that noisy bands tend to have higher variances and that the spectral bands with higher variances can often be highly correlated, which can lead to repetitious information representing a small portion of the dataset being selected to incorrectly represent the entire dataset [7].

3.1.4 Normalized Cut Optimal Clustering MVPCA

The NC-OC-MVPCA approach was introduced in [43] as a clustering-based band selection method with the capability of finding an optimal solution. The NC-OC-MVPCA uses the normalized cut (NC) criterion as its objection function. With the NC, the space is assumed to be a graph of $G = (V, W)$ with V being the node set and W being the similiarity matrix. The individual entries of this similiarity matrix are created by using

$$w_{ij} = exp(-\frac{\|x_i - x_j\|^2}{\sigma_i \sigma_j}) \tag{3.6}$$

where w_{ij} is $ijth$ entry of the similarity matrix, $\sigma_i = \|x_i - x_m\|^2$ is the local scaling parameter, x_m is the mth neighbor of x_i This similarity matrix is then mapped onto the function

$$f_{na}(X_i^j) = \frac{1}{K} \frac{\sum_{k=i}^{j} \sum_{l=i}^{j} w_{kl}}{\sum_{k=i}^{j} \sum_{l=1}^{L} w_{kl}} \tag{3.7}$$

where K is the number of partitions of the graph G. MVPCA is used to rank the bands. Dynamic programming is then done, using the NC objection function and ranked bands, to select the predefined number of bands. This concept of dynamic programming, presented in [20], [23], is a method in which a convoluted problem is reduced into a deconstructed problem set of several subproblems that are feasible to resolve. Once this point is reached, the problems are then incorporated back together until the original convoluted problem is solvable. This band selection technique can also be sensitive to noisy bands [43].

3.2 Dataset

The Indian Pines dataset is hyperspectral image data provided by Purdue University [4]. This set depicts a test site in Indiana, and it was collected by use of an Airborne Visible Infrared Imaging

Spectrometer (AVIRIS) sensor on June 12, 1992. It is comprised of 145 by 145 pixels with 224 spectral bands and 16 classes. Purdue University reduced the data set to 220 bands, removing bands 1, 33, 97, and 161. The Computational Intelligence Group (CIG) provided a further reduced data set of 145 by 145 pixels and 200 bands that removed the bands from the original dataset that were corrupted due to water absorption [17]. The bands removed were bands 104-108, 140-163, and 220. Figure 3.1 shows the ground truth image of Indian Pines, and Table 3.1 shows the pixel make up of the Indian Pines Groundtruth.

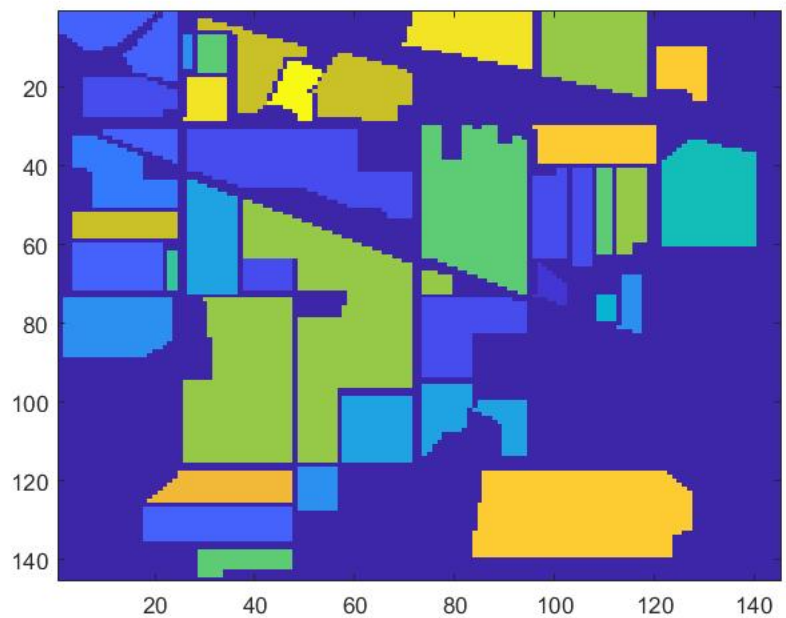


Figure 3.1: Indian Pines Groundtruth [4].

Table 3.1: Sample Numbers of Indian Pines Groundtruth classes

Number	Class	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93

3.3 MATLAB Utilization

In this section, there is discussion of the approaches that were utilized for per-pixel classification of the hyperspectral image data in its original form, i.e. the full datacube, and for classification of the the reduced spectral band set hyperspectral image data. For the preliminary implementations, the spectral band values chosen for the reduced hyperspectral image data were 25, 50, 75, and 100. These spectral band quantities were selected with the assumption that a wide range in band values that make up a large percentage of the data will significantly alter the classification performance of the reduced data sets. Both of the classification models, ANN and SVM, in this implementation were run on an Intel(R) Core(TM) i7-8565U CPU 1.80GHz and on MATLAB R2021a. Figure 3.2 shows a general overview of the process in which the classification models are trained and tested. The hyperspectral image data spectral band values is down-selected using one of the four band selection techniques presented previously. The resulting reduced image data is then split into training and testing data, which were used to train the classification model, and then the trained model is tested on the test dataset.

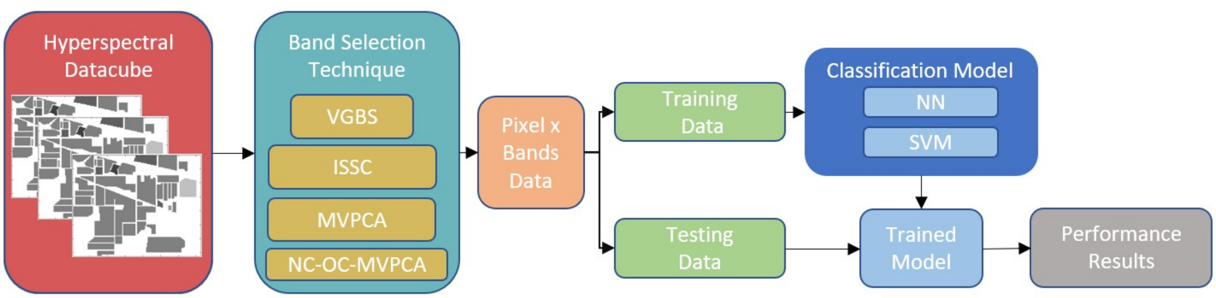


Figure 3.2: Preliminary Work Overview

Multiple ANN configurations/architectures were considered for the classification task, and the following architecture was empirically found to provide acceptable results for this investigation. The ANN is composed of six layers: one input layer, three fully connected layers, a soft max layer, and an output layer. The input layer has nodes corresponding to the number of input values, e.g. 25 nodes are used when the reduced hyperspectral image data set are comprised of 25 spectral bands. The fully connected layers are comprised of 25, 25, and 17 nodes. The values of the first two layers were chosen arbitrarily - the intention was to have low values to avoid overfitting but also high enough to encourage feature learning. The final hidden layer is made up of 17 nodes. This is done to account for the 16 classes of the Indian Pines datsaset and for the background information. Experiments were conducted ten times, and the performances reported herein are the average of these experiments. For clarity, each experiment randomly selected 79.90% of the data for training, and the remaining 20.1% was used for testing. This allows for a better test of the robustness/generalizability of the bands selected by the technique being considered and ensures the inference timings are consistent and not interfering with the operating system allocating cores for various processes. Figure 3.3 provides a detailed overview of the ANN training and testing process.

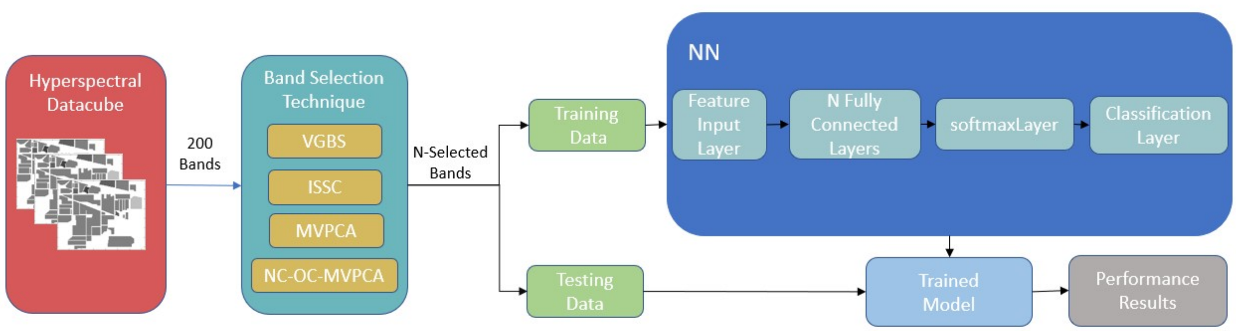


Figure 3.3: Initial Implementation Process

The SVM model for this classification task used the radial basis function (RBF) as its learning parameter. The RBF kernel can be represented by the following equation:

$$K(X_1, X_2) = exp(-\frac{||X_1 - X_2||^2}{2\sigma^2}) \tag{3.8}$$

where σ is the variance and $||X_1 - X_2||$ is the Euclidean distance between the two points. This SVM performed multi-class classification by using the one-against-one method. Thus, there are $k(k-1)/2$ classifiers created, where k is the number of total classes in the set. This value is selected to account for the comparison of each of the classifiers against each other, while removing cases where it would overlap or compare against itself. The data is decided to be in the class in which it has the highest probability of being a part. [18] The hyperspectral image data was separated into 10% of the data for training, and the remaining 90% was used for testing.

3.4 Results

Table 3.2 displays the classification accuracy results from the tests run using the trained ANN on MATLAB, and Table 3.3 shows the percent difference of the classification accuracies of the reduced spectral band hyperspectral data compared to those of the full hyperspectral image data tested on the ANN. Figure 3.4 shows the progression of the classification accuracy as the spectral band values increase.

As can be seen from the figures and table, the ISSC and NC-OC-MVPCA band selection techniques performed the best out of the techniques. The ISSC technique performed better than the full hyperspectral data outperformed the other selection techniques at every band selection value, with the exception of the 75 band selection. The NC-OC-MVPCA produced fairly similar accuracies to those of the ISSC technique and was able achieve a comparable classification accuracy

Table 3.2: NN Classification Accuracy Results

Technique	25 Bands	50 Bands	75 Bands	100 Bands	200 Bands
Full Matrix	-	-	-	-	**65.73**%
VGBS	60.63%	63.26%	65.85%	67.16%	-
ISSC	**65.85**%	**67.76**%	67.94%	**68.85**%	-
MVPCA	56.28%	61.33%	65.99%	66.11%	-
NC-OC-MVPCA	65.59%	67.13%	**68.04**%	68.25%	-

Table 3.3: NN Percent Difference between Band Selection Techniques and Full Hyperspectral Image Data

Technique	25 Bands	50 Bands	75 Bands	100 Bands
VGBS	-8.07%	-3.83%	0.18%	2.15%
ISSC	**0.18**%	**3.04**%	3.30%	**4.63**%
MVPCA	-15.49%	-6.92%	0.40%	0.58%
NC-OC-MVPCA	-0.22%	2.11%	**3.45**%	3.76%

value to the full hyperspectral data with as little as 25 bands. The VGBS and MVPCA algorithms needed 75 bands to get accuracies similar to that of the full hyperspectral image data, but these techniques were able to output accuracies within 15% of the full hyperspectral data with only 25 bands selected. Overall, the band selection techniques did not perform with much dissimilarity from the original full band set, and some were able to outperform the original hyperspectral data with a small percentage of the bands.

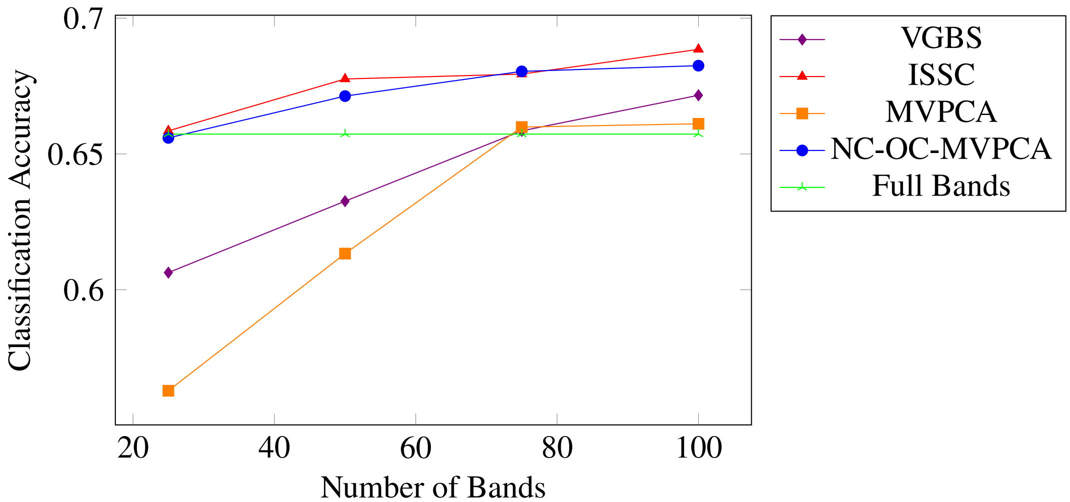


Figure 3.4: Comparison of Classification Accuracy amongst Techniques

Table 3.4 shows the run times of the trained ANN on MATLAB, and Table 3.5 shows the percent difference of the run times of the reduced spectral band hyperspectral data compared to those of the full hyperspectral image data tested on the ANN. Figure 3.5 shows the progression of the run times as the spectral band values increase.

Table 3.4: NN Run Time of Band Selection Techniques

Technique	25 Bands	50 Bands	75 Bands	100 Bands	200 Bands
Full Matrix	-	-	-	-	**0.0424s**
VGBS	0.0317s	0.0355s	0.0335s	0.0360s	-
ISSC	0.0304s	**0.0330s**	**0.0335s**	0.0411s	-
MVPCA	**0.0261s**	0.0344s	0.0355s	0.0386s	-
NC-OC-MVPCA	0.0290s	0.0344s	0.0338s	**0.0356s**	-

Table 3.5: NN Percent Difference of Band Selection and Full Matrix Run Times

Technique	25 Bands	50 Bands	75 Bands	100 Bands
VGBS	28.84%	17.54%	23.27%	16.20%
ISSC	33.04%	**24.85**%	**23.43**%	3.05%
MVPCA	**47.49**%	20.82%	17.68%	9.27%
NC-OC-MVPCA	37.30%	20.74%	22.42%	**17.44%**

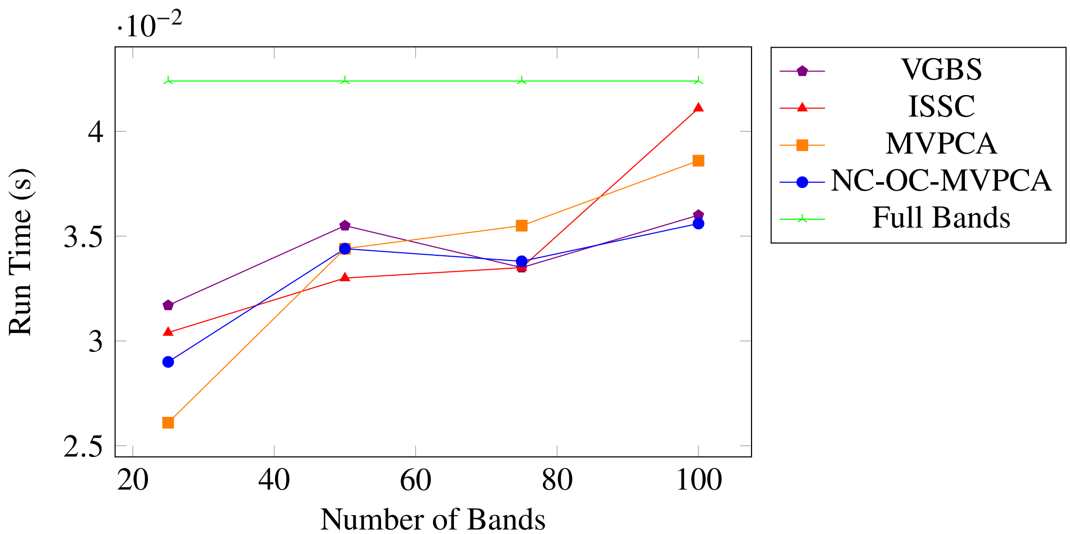


Figure 3.5: Comparison of Performance Run Time amongst Techniques

All of the band selection techniques were able to test the ANN classification model faster than the full hyperspectral data, despite the band selection values chosen. The speeds, with the exception of ISSC and MVPCA with 100 bands selected, were a substantial percentage faster, with some of the techniques running 30-40% faster than the full hyperspectral data. On average, the ISSC algorithm produced the quickest run time. The VGBS and NC-OC-MVPCA had consistent competitive run times regardless of the number of bands selected. While MVPCA started as the fastest technique, it gradually became one of the slowest.

Table 3.6 presents the classification accuracy results from the tests run using the trained SVM on MATLAB, and Table 3.7 shows the percent difference of the classification accuracies of the reduced spectral band hyperspectral data compared to those of the full hyperspectral image data tested on the SVM. Figure 3.7 shows the progression of the classification accuracy as the spectral band values increase.

Table 3.6: SVM Classification Accuracy Results

Technique	25 Bands	50 Bands	75 Bands	100 Bands	200 Bands
Full Matrix	-	-	-	-	**61.52**%
VGBS	**62.00**%	**65.26**%	**63.66**%	62.38%	-
ISSC	58.93%	61.76%	61.28%	61.40%	-
MVPCA	59.60%	62.21%	63.42%	**62.40**%	-
NC-OC-MVPCA	59.74%	60.02%	60.81%	61.31%	-

For the SVM classification model, the VGBS on average performed the best, even with 100 bands it was only 0.02% less accurate than the MVPCA, and it was able to achieve comparable classification accuracy to the full hyperspectral image data matrix with just 25 bands. The ISSC only required 50 bands for it to achieve similar classification results, and it took the MVPCA somewhere between 50 and 75 bands to do so. The NC-OC-MVPCA band selection technique was

Table 3.7: SVM Percent Difference of the Classification Accuracy of the Band Selection Techniques and Full Hyperspectral Data

Technique	25 Bands	50 Bands	75 Bands	100 Bands
VGBS	**0.77**%	**5.89**%	**3.42**%	1.38%
ISSC	-4.30%	0.39%	0.39%	0.19%
MVPCA	-3.18%	-1.11%	3.04%	**1.42**%
NC-OC-MVPCA	-2.94%	-2.47%	-1.17%	0.35%

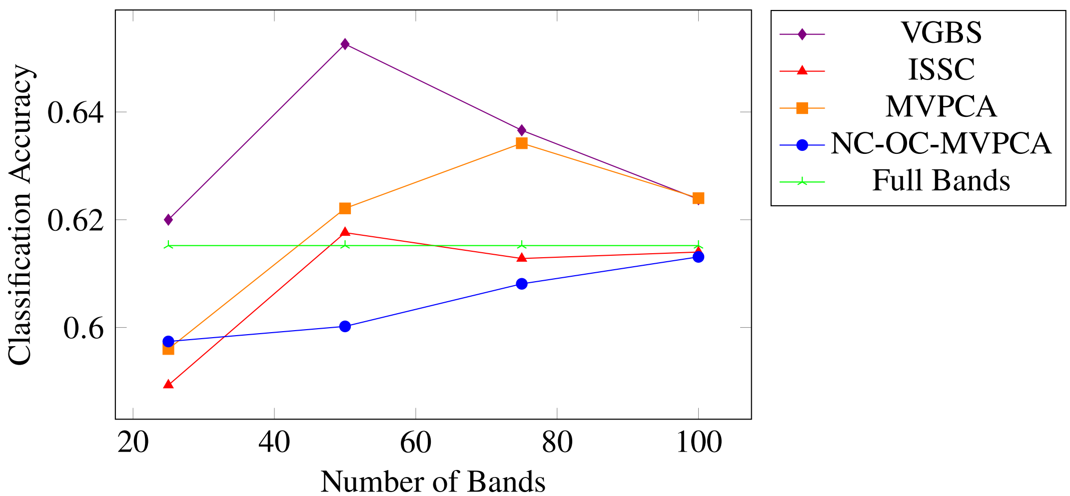


Figure 3.6: Comparison of Classification Accuracy amongst Techniques

on average the worst performer in terms of classification accuracy, but its results were not too far off with a 3% percent difference being the maximum difference for the technique.

Table 3.8 presents the run times from the tests run using the trained SVM on MATLAB, and Table 3.9 shows the percent difference of these run times between those of the reduced spectral band hyperspectral data and those of the full hyperspectral image data tested on the SVM. Figure

3.7 shows the run times for each of the band selection techniques over the selection of 25 to 100 band values. For this graph, the full hyperspectral data was not included due to its high run time which caused the graph to not be as readable.

Table 3.8: Run Time of Band Selection Techniques

Technique	25 Bands	50 Bands	75 Bands	100 Bands	200 Bands
Full Matrix	-	-	-	-	**38.2892s**
VGBS	6.5906s	16.5212s	17.3957s	21.7341s	-
ISSC	**3.8681s**	7.2948s	**10.1671s**	**13.3822s**	-
MVPCA	3.8897s	**7.0858s**	11.0509s	14.3912s	-
NC-OC-MVPCA	4.2615s	7.3915s	10.3041s	13.4619s	-

Table 3.9: Percent Difference of Band Selection and Full Matrix Run Times

Technique	25 Bands	50 Bands	75 Bands	100 Bands
VGBS	141.26%	79.43%	75.04%	55.16%
ISSC	**163.30%**	135.99%	**116.07%**	**96.41%**
MVPCA	163.11%	**137.54%**	110.41%	90.73%
NC-OC-MVPCA	159.94%	135.28%	115.18%	95.95%

Similar to that of the NN, all of the performance times with the band selection techniques provide for run times that are much quicker than the hyperspectral full spectral bands. The ISSC

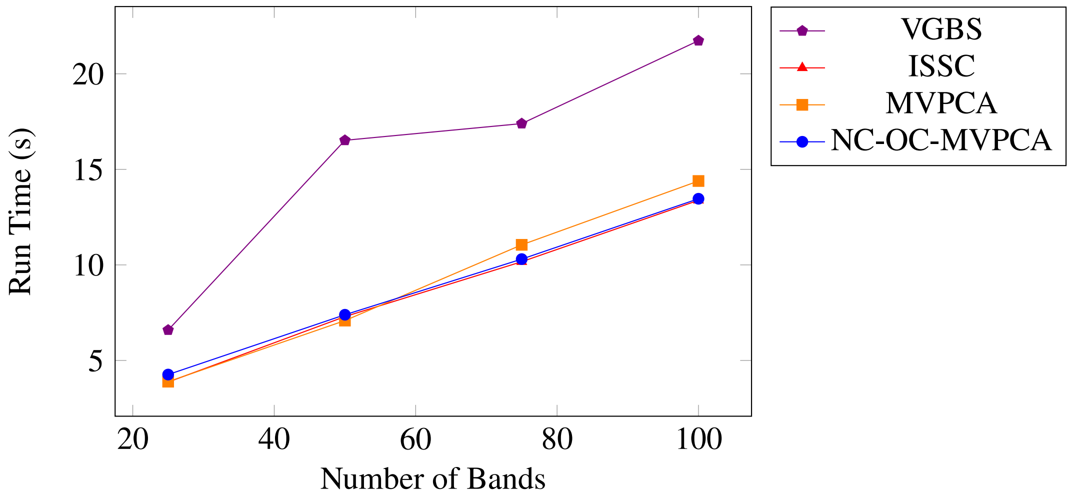


Figure 3.7: Comparison of Performance Run Times amongst Techniques

was the highest performer on average with a faster run time of up to 163.30%. Whereas, the VGBS was the slowest of the techniques, with its speed decreasing to a less than 100% percent difference from the full hyperspectral set within only 50 bands selected. The MVPCA and NC-OC-MVPCA performed comparably to the ISSC algorithm, with only fractions of a second difference.

CHAPTER IV

AT THE EDGE APPLICATION

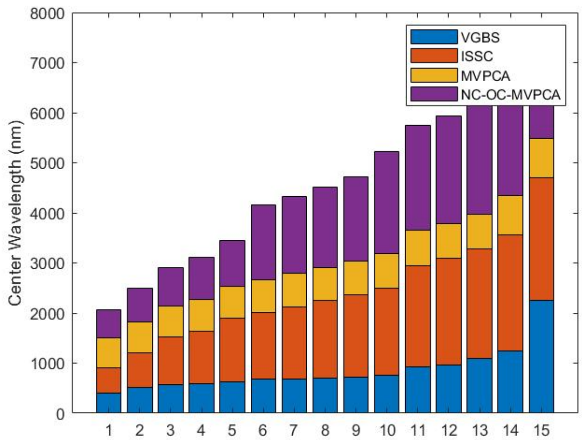
The focus of this section is to introduce a device that presents results for at the edge implementations and to compare the classification performance produced by MATLAB. This section also includes an implementation process similar to that from the previous section but with a narrower range of band selection values.

4.1 Implementation

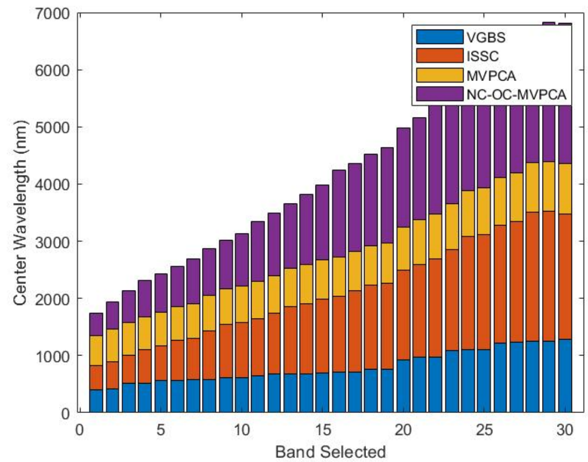
With the initial testings done for this thesis, the band values that were selected, 25-100 bands, spanned a large percentage of the data, 12.5% to 50% [5]. After reading more into current hyperspectral band selection research in which a more selective range is applied, such as [46] with 5-40 bands, [25] with 10-50 bands, [39] with 5-60 bands, and [38] with 5-60 bands selected, the spectral band values for the following implementations were selected to be 15, 30, 45, and 60.

Figure 4.1 shows the diverse selection of the four band selection techniques when tasked with selecting 15, 30, 45, and 60 bands. The x-axis represents each of the bands that were selected by the techniques, and the y-axis represents the center wavelength of the selected band in nanometers.

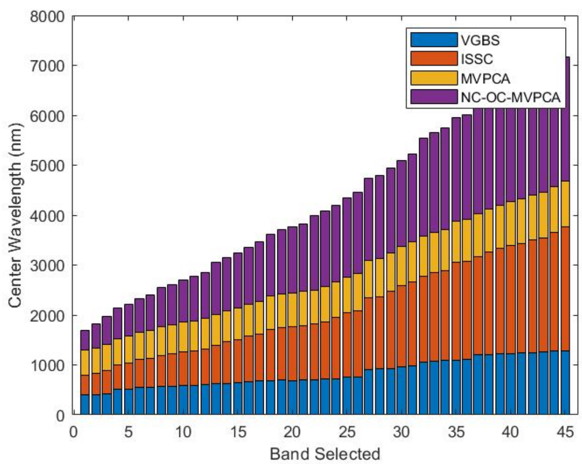
These new band selection values were used to create down-selected hyperspectral data that was then split into training and testing data. The split between the two types of data was done by randomly selecting 80% of the data to be training and the remaining 20% to be the testing data.



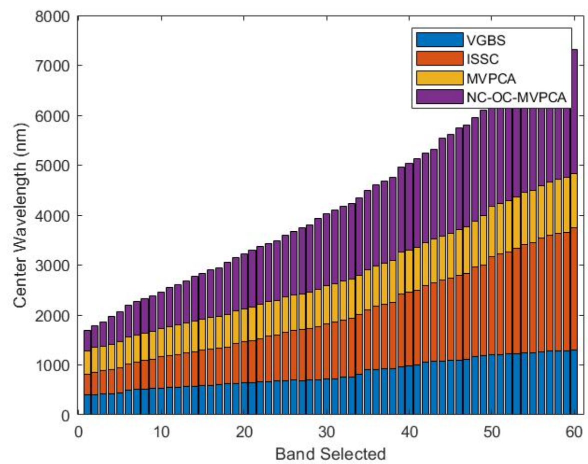
(a) 15 Bands Selected



(b) 30 Bands Selected



(c) 45 Bands Selected



(d) 60 Bands Selected

Figure 4.1: The Wavelengths Associated with the Different Bands Selected

Once the training data input was created, it was placed onto the chosen classification model, either ANN or SVM, and used to create a trained classification model.

Figure 4.2 gives a visual representation of the process in which the training model is trained on MATLAB and tested on MATLAB and the NVIDIA Jetson AGX Xavier, which will be referred to as NVIDIA for the remainder of this chapter.

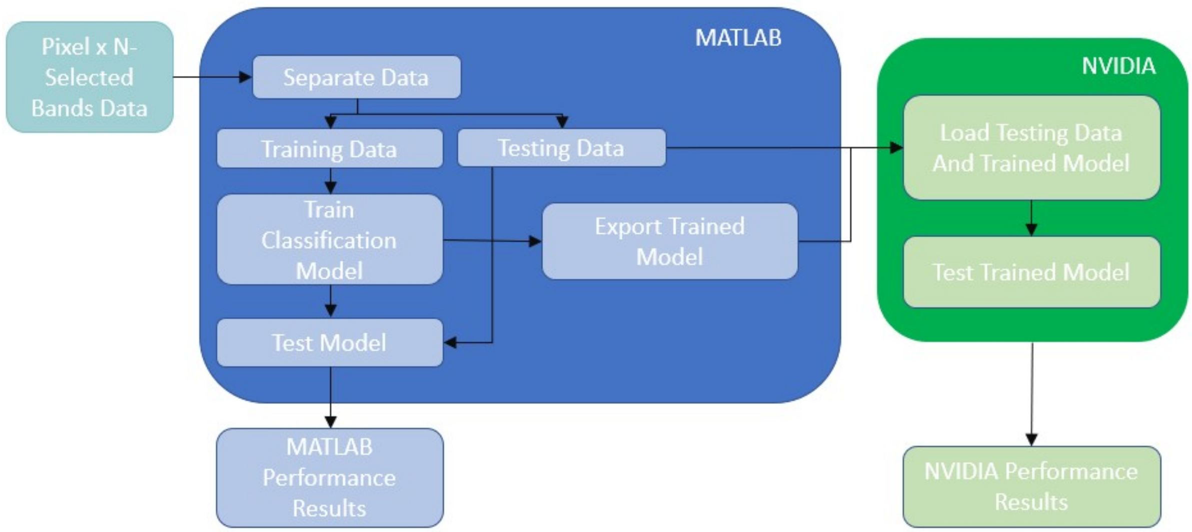


Figure 4.2: Block Diagram of Real Time Application

For placing the trained ANN model onto the NVIDIA, Open Neural Network Exchange (ONNX) was utilized. ONNX is an open-source platform that seeks to expand the uses of artificial intelligence research beyond the use of a single platform and has a community of users that have developed different resources for other users to be able to place models onto other platforms [3]. The models on MATLAB were created to be DAGNetworks to fit within the ONNX conversion on MATLAB. Once the models were converted to the correct format, they were placed on the

NVIDIA. There are numerous Ubuntu compatible platforms that can then be used to test the placed models. For the realm of this research, the focus was limited to using Python applications. The ONNX-runtime program was run in Python to test the given models. The responding results were then recorded for comparison to the MATLAB implementations of the same classification models.

For placing the SVM model onto the NVIDIA, the models were converted from a MATLAB structure to a C structure. This structure was compatible with python, which allowed for the process to run similarily to that of the ANN model process for the remainder of the testing.

Once the models were placed, the models were tested on both MATLAB and the NVIDIA for comparison of classification performance. The same testing data was used to evaluate both of the applications in order to have a comparable reference for classification performance. The performance was recorded in terms of classification accuracy and run times based on the testing of these trained models.

4.2 Results

4.2.1 Classification of Reduced Spectral Band Numbers on MATLAB

Table 4.1 shows the classification accuracy results from the tests run using the lower band selected values on the trained NN on MATLAB, and Table 4.2 shows the percent difference of these new lower band selection values' classification accuracies compared to that of the full hyperspectral image data matrix. Figure 4.3 shows the progression of the classification accuracy as the spectral band values increase.

From these results, it is evident that the ISSC and NC-OC-MVPCA band selection techniques were the best performing techniques in terms of classification accuracy. VGBS was able to perform well compared to the full hyperspectral image data matrix when almost 60 bands were selected.

Table 4.1: NN Classification Accuracy of Band Selection Techniques

Technique	15 Bands	30 Bands	45 Bands	60 Bands	200 Bands
Full Matrix	-	-	-	-	**65.73%**
VGBS	57.44%	61.15%	63.13%	65.77%	-
ISSC	62.95%	**67.04**%	**67.15**%	67.82%	-
MVPCA	55.83%	56.80%	60.05%	61.70%	-
NC-OC-MVPCA	**63.40**%	65.54%	67.07%	**68.23**%	-

Table 4.2: NN Percent Difference of the Classification Accuracy of the Band Selection Techniques and Full Hyperspectral Data

Technique	15 Bands	30 Bands	45 Bands	60 Bands
VGBS	-13.468%	-7.22%	-4.04%	0.05%
ISSC	-4.32%	**1.97**%	**2.13**%	3.13%
MVPCA	-16.29%	-14.57%	-9.04%	-6.33%
NC-OC-MVPCA	**-3.61**%	-0.29%	2.02%	**3.73**%

The MVPCA band selection technique performed the worst of the band selection techniques, with it not being able to reach a comparable classification accuracy within the 60 band limit.

Table 4.3 presents the run time results from the tests run using the trained NN on MATLAB, and Table 4.4 shows the percent difference of the run times of the reduced spectral band hyperspectral data compared to those of the full hyperspectral image data tested on the NN. Figure 4.4 shows the progression of the run times as the spectral band values increase.

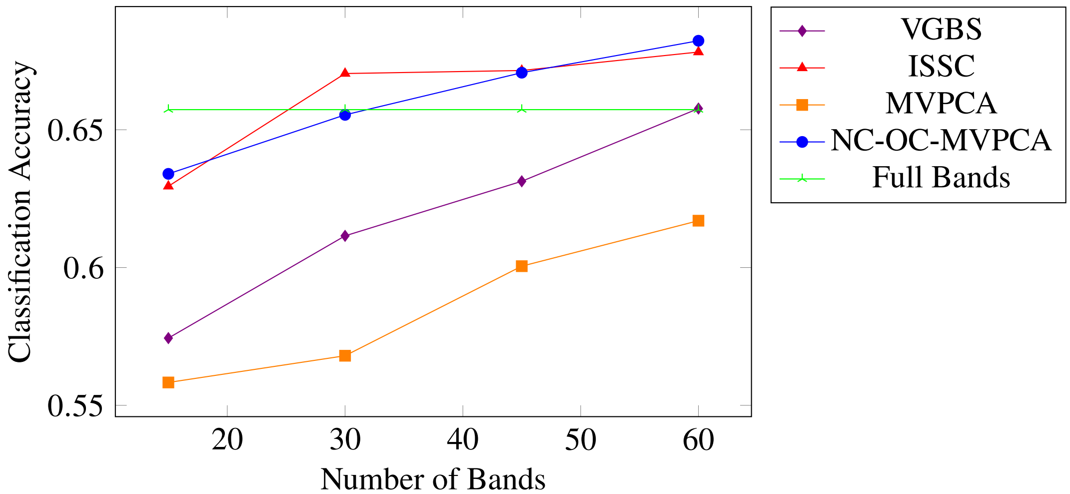


Figure 4.3: Comparison of Classification Accuracy amongst Techniques

Table 4.3: Run Times of Band Selection Techniques

Technique	15 Bands	30 Bands	45 Bands	60 Bands	200 Bands
Full Matrix	-	-	-	-	**0.0424s**
VGBS	**0.0271s**	0.0310s	0.0349s	**0.0309s**	-
ISSC	0.0312s	0.0274s	**0.0322s**	0.0367s	-
MVPCA	0.0322s	0.0326s	0.0327s	0.0366s	-
NC-OC-MVPCA	0.0288s	**0.0250s**	0.0338s	0.0332s	-

As can be seen from the figures, all of the band selection techniques run at shorter run times than the full hyperspectral image data set, which is to be expected with such low values chosen for the band sets. But what makes the results significant is the percentage in which the speed is increased by. Most of the percentages are between 25-45%, which is a significant amount when less than 30% of the bands are being selected for classification of the entire set. Table 4.5 presents

Table 4.4: Percent Difference of Band Selection and Full Matrix Run Times

Technique	15 Bands	30 Bands	45 Bands	60 Bands
VGBS	**43.94%**	31.10%	19.20%	**31.44%**
ISSC	30.28%	42.89%	**27.38%**	14.29%
MVPCA	27.32%	25.73%	25.71%	14.72%
NC-OC-MVPCA	38.06%	**51.45%**	22.56%	24.20%

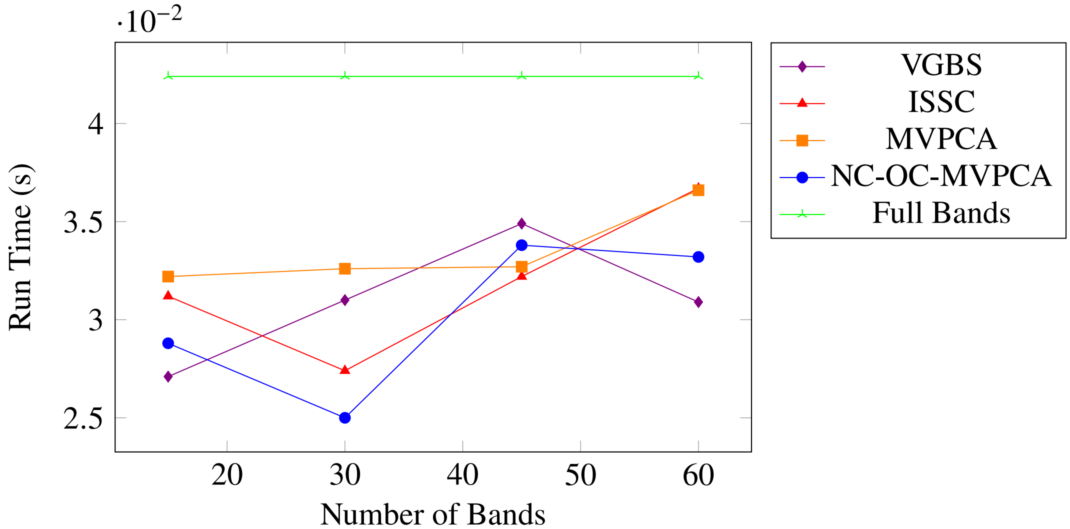


Figure 4.4: Comparison of Performance Run Times amongst Techniques

the classification accuracy results from the tests run using the trained SVM on MATLAB, and Table 4.6 shows the percent difference of the run times of the reduced spectral band hyperspectral data compared to those of the full hyperspectral image data tested on the SVM. These band selected values are also of the new chosen smaller values. Figure 4.5 shows the progression of the run times as the spectral band values increase.

Table 4.5: Classification Accuracy of Band Selection Techniques

Technique	15 Bands	30 Bands	45 Bands	60 Bands	200 Bands
Full Matrix	-	-	-	-	**61.52%**
VGBS	**62.95%**	**62.24%**	**64.94%**	**65.59%**	-
ISSC	58.86%	59.22%	61.26%	61.31%	-
MVPCA	58.60%	60.40%	61.69%	62.93%	-
NC-OC-MVPCA	58.85%	59.79%	59.98%	60.55%	-

Table 4.6: Percent Difference of the Classification Accuracy of the Band Selection Techniques and Full Hyperspectral Data

Technique	15 Bands	30 Bands	45 Bands	60 Bands
VGBS	**2.29%**	**1.15%**	**5.416%**	**6.40%**
ISSC	-4.43%	-3.82%	0.43%	0.35%
MVPCA	-4.87%	-1.83%	-0.27%	-2.25%
NC-OC-MVPCA	-4.44%	-2.86%	-2.54%	-1.60%

The VGBS band selection technique was the highest performing in terms of classification accuracy of all the band selection techniques. With only 15 bands, the VGBS is able to perform at slightly higher classification accuracy than the full set. The other band selection techniques all produce fairly comparable results, even though the ISSC is the only one of the three that reaches the higher accuracy within the 60 band value limit.

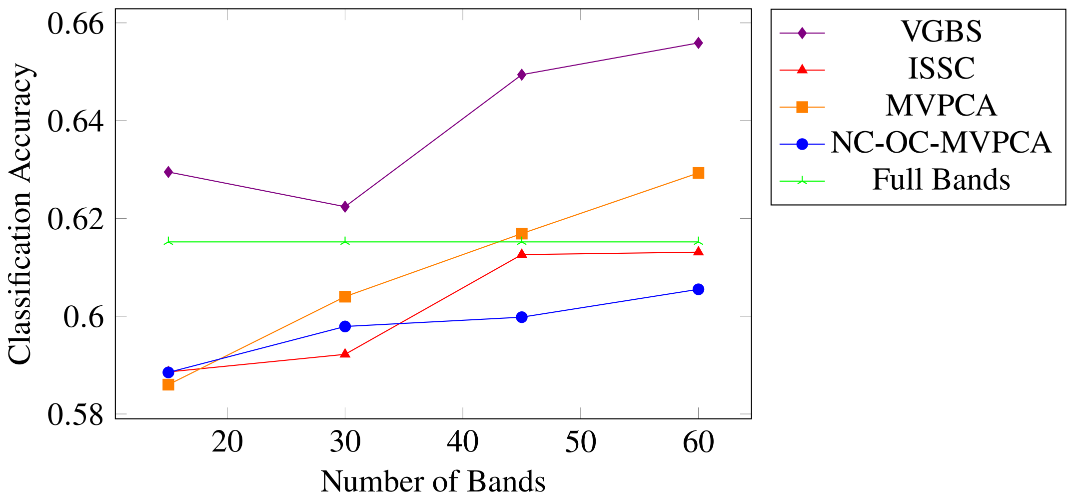


Figure 4.5: Comparison of Classification Accuracy amongst Techniques

Table 4.7 presents the run time results from the tests run using the trained SVM on NVIDIA, and Table 4.8 shows the percent difference of the run times of the reduced spectral band hyperspectral data compared to those of the full hyperspectral image data tested on the SVM. Figure 4.6 shows the progression of the run times as the spectral band values increase.

Table 4.7: Run Times of Band Selection Techniques

Technique	15 Bands	30 Bands	45 Bands	60 Bands	200 Bands
Full Matrix	-	-	-	-	**20.5540s**
VGBS	4.8212s	9.4018s	10.2021s	16.1602s	-
ISSC	2.7487s	4.8112s	6.1739s	8.9576s	-
MVPCA	**2.6887s**	**4.4769s**	**6.1373s**	9.7854s	-
NC-OC-MVPCA	2.9497s	4.7123s	6.1702s	**8.6029s**	-

Table 4.8: Percent Difference of Band Selection and Full Matrix Run Times

Technique	15 Bands	30 Bands	45 Bands	60 Bands
VGBS	155.27%	121.14%	115.84%	81.28%
ISSC	173.21%	155.35%	144.46%	124.16%
MVPCA	**173.75%**	**158.13%**	**144.74%**	118.58%
NC-OC-MVPCA	171.39%	156.17%	144.49%	**126.62%**

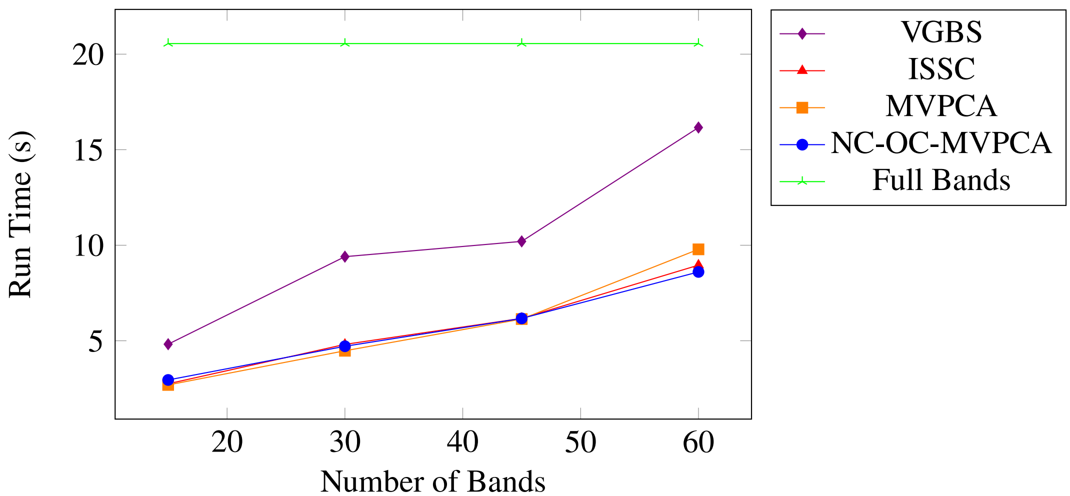


Figure 4.6: Comparison of Run Times amongst Techniques

4.2.2 Classification of Reduced Spectral Band Numbers on NVIDIA

The results from the implementations tested on MATLAB's classification accuracy are the same values as that of the NVIDIA tested models, so the remaining section will only present the run time values from the NVIDIA tests. Table 4.9 compares the run times of the band selection techniques placed on the ANN in NVIDIA. The following Table 4.10 gives the percent difference of the given performance run times compared to the full matrix placed onto the ANN. Figure 4.7 shows the progression of the run times as the spectral band values increase.

Table 4.9: Run Times of Band Selection Techniques

Technique	15 Bands	30 Bands	45 Bands	60 Bands	200 Bands
Full Matrix	-	-	-	-	**0.0467s**
VGBS	0.0045s	0.0047	**0.0053s**	0.0320s	-
ISSC	0.0043s	**0.0045s**	0.0058s	**0.0063s**	-
MVPCA	**0.0042s**	0.0049s	0.0059s	**0.0063s**	-
NC-OC-MVPCA	0.0043s	0.0046s	0.0057s	0.0065s	-

Table 4.10: Percent Difference of Band Selection and Full Matrix Run Times

Technique	15 Bands	30 Bands	45 Bands	60 Bands
VGBS	64.41%	59.50%	**48.49%**	22.08%
ISSC	68.21%	**63.12%**	40.82%	**31.61%**
MVPCA	**70.66%**	56.46%	38.40%	**31.61%**
NC-OC-MVPCA	67.28%	62.50%	42.08%	29.67%

Depending on the number of bands selected, a different band selection technique performed the best. All of the band selection techniques performed relatively close to the same, with all of the times being 22-70% faster than that of running the entire hyperspectral image data. This is

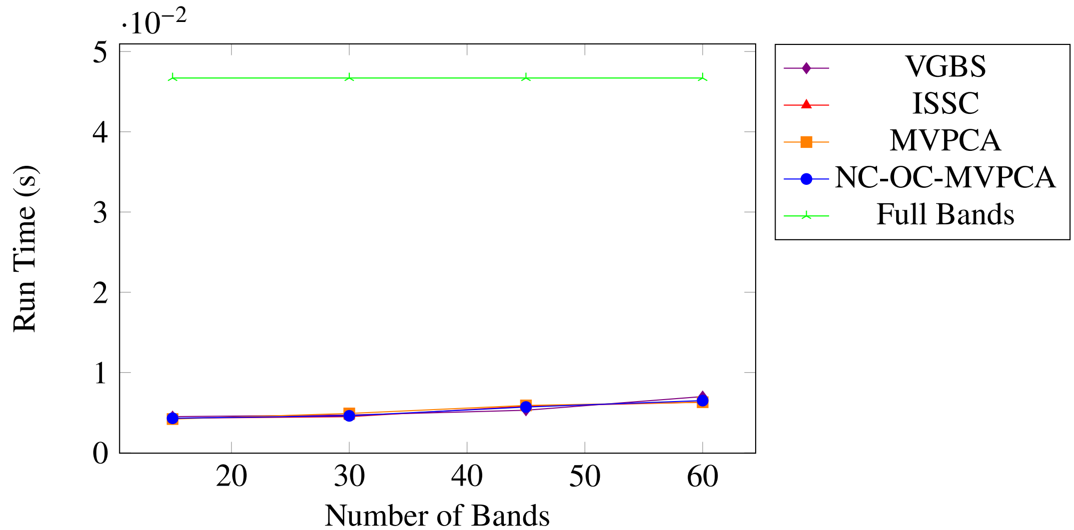


Figure 4.7: Comparison of Run Times amongst Techniques

significant because even in the case of using MVPCA for the selection of 15 bands, though the loss in classification accuracy is 16.29%, the gain of performance run time is 70%, which depending on the leniency on classification dependence, could be a good trade off. Another example could be use of the NC-OC-MVPCA with 15 bands, there is a 67.28% decrease in performance run time with just a loss of 3.61% in classification accuracy.

Table 4.11 compares the run times of the band selection techniques placed on the SVM in NVIDIA. The following Table 4.12 gives the percent difference of the given performance run times compared to the full matrix placed onto the SVM. Figure 4.8 shows the progression of the run times as the spectral band values increase.

As with the classification accuracries, the ISSC and NC-OC-MVPCA are the top performing band selection techniques. Although it is important to note that all of the band selection techniques perform relatively close to one another and follow the trend of increasing about a second with the

Table 4.11: Run Times of Band Selection Techniques

Technique	15 Bands	30 Bands	45 Bands	60 Bands	200 Bands
Full Matrix	-	-	-	-	**20.553994s**
VGBS	5.1868s	6.2416s	7.2431s	8.0562s	-
ISSC	5.1816s	**6.2117s**	7.3467s	**8.0473s**	-
MVPCA	5.5538s	6.5479s	7.5140s	8.1932s	-
NC-OC-MVPCA	**5.1597s**	6.2742s	**7.1305s**	**8.2428s**	-

Table 4.12: Percent Difference of Band Selection and Full Matrix Run Times

Technique	15 Bands	30 Bands	45 Bands	60 Bands
VGBS	119.40%	106.83%	95.77%	87.37%
ISSC	119.46%	**107.17**%	94.67%	**87.46**%
MVPCA	114.91%	103.36%	92.92%	86.00%
NC-OC-MVPCA	**119.74%**	106.45%	**96.98%**	85.50%

increase in 15 bands. These band selection techniques performed significantly faster than the full data set, with the ranges being from 85.50%-119.74%.

4.2.3 Comparison of the Run Times for MATLAB and NVIDIA

As mentioned before, the initial comparison for the MATLAB and NVIDIA results were going to be in terms of both classification accuracy and performance run time. However, the classification accuracies remained the same for both devices, so the focus of this comparison is based on the run

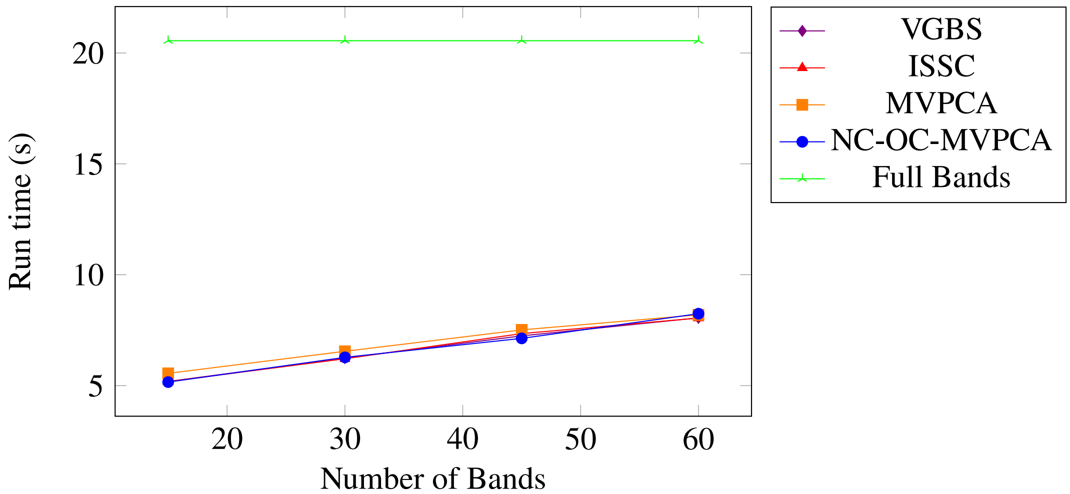


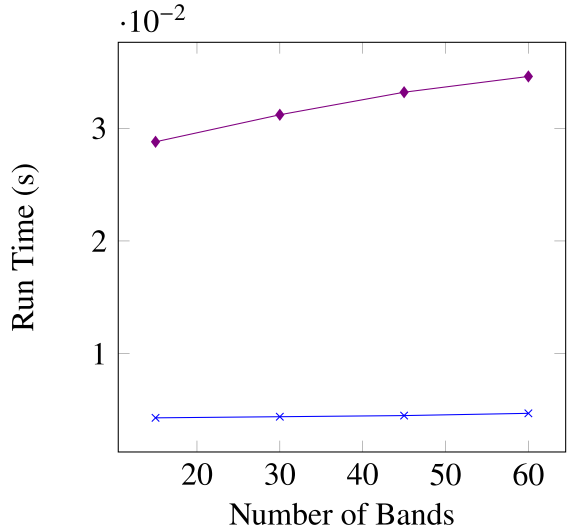
Figure 4.8: Comparison of Run Times amongst Techniques

times from the two. Figure 4.9 shows the comparison of the four band selection techniques when run on both of the devices.

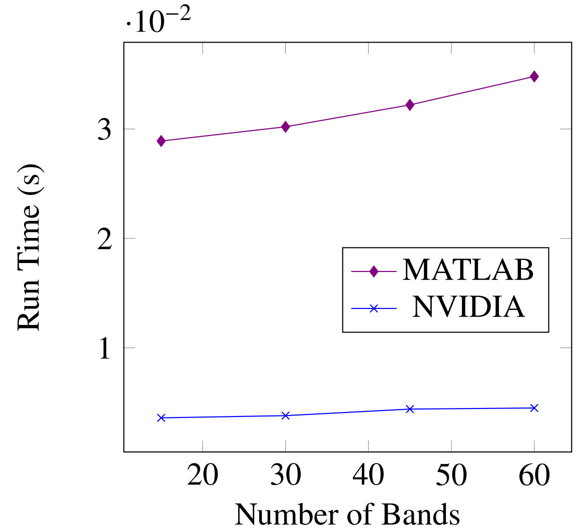
As can be seen from the devices, the NVIDIA ran faster than the MATLAB device despite the number of bands that were placed onto the device.

Figure 4.10 shows the comparison of the run times of the VGBS, ISSC, MVPCA, and NC-OC-MVPCA band selection techniques with increasing band selection values when trained through a SVM classification model and tested on the MATLAB and NVIDIA devices.

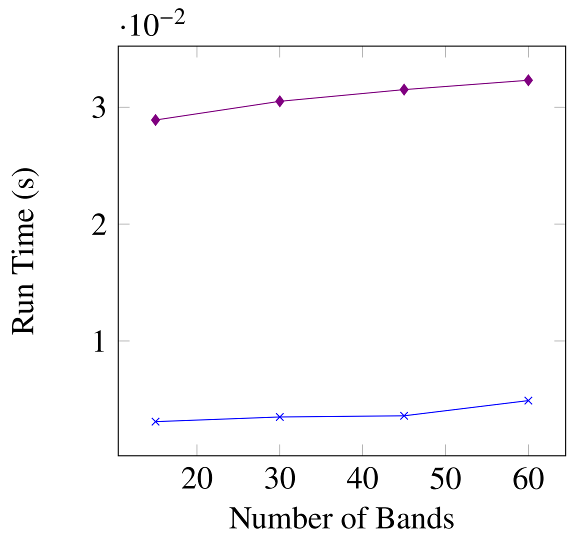
With these comparisons, it is evident that both of the devices perform at more comparable run times. In the VGBS appplication, NVIDIA ran at much faster speeds, regardless of the increase in band values, with the exception at 20 bands selected. Whereas, for both the ISSC, MVPCA, and NC-OC-MVPCA tests, the MATLAB implementation ran at a much quicker speed until after 50 bands were selected. The importance of the speed of the NVIDIA is not to conclude that it



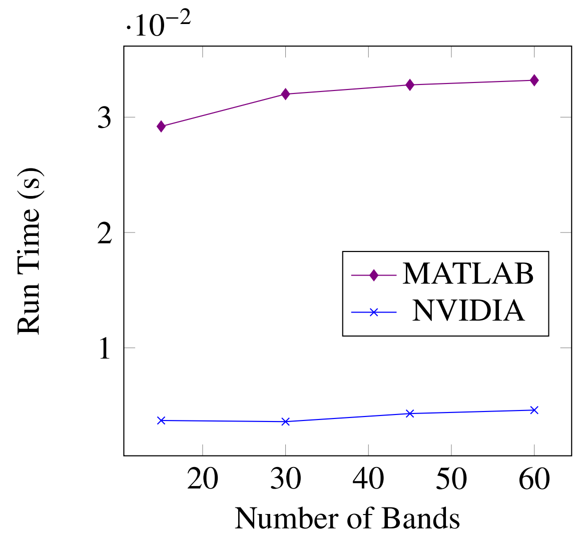
(a) Comparison of VGBS Run-Times



(b) Comparison of ISSC Run-Times

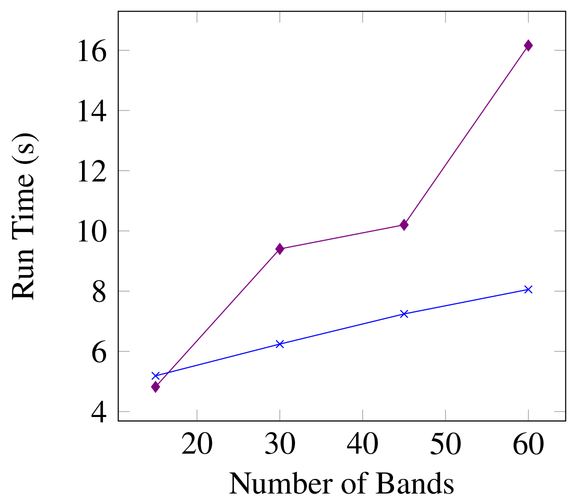


(c) Comparison of MVPCA Run-Times

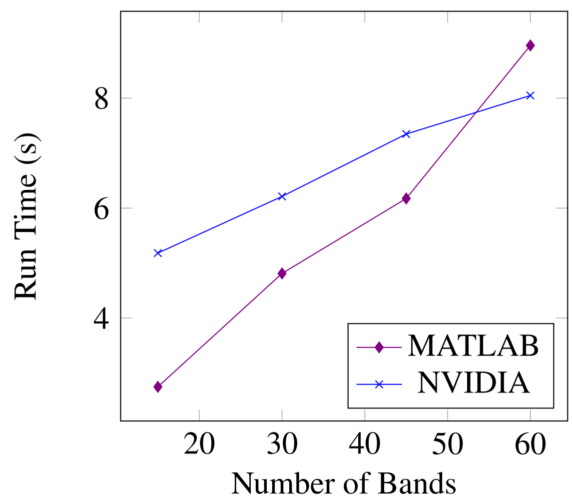


(d) Comparison of NC-OC-MVPCA Run-Times

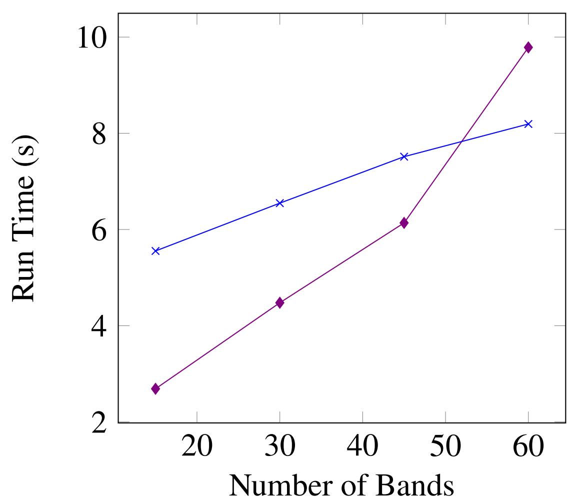
Figure 4.9: Run-Times for NN on NVIDIA and MATLAB



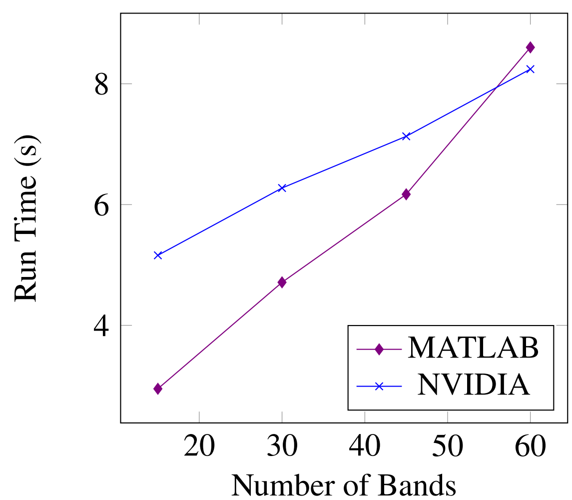
(a) Comparison of VGBS Run-Times



(b) Comparison of ISSC Run-Times



(c) Comparison of MVPCA Run-Times



(d) Comparison of NC-OC-MVPCA Run-Times

Figure 4.10: Run-Times for SVM on NVIDIA and MATLAB

performed "better" than the MATLAB tests, but to show that it can have a run time comparable to that of simulated tests, which can then attest to its feasibility of being used as a real-time device.

CHAPTER V

CONCLUSIONS

The overall purpose of this research was to test the use of band selection techniques and investigate their practicality in real-time at the edge applications. The classification accuracies and performance times reflect acceptable and achievable results, which confirms that the band selection techniques do have the ability to retain important information needed for the testing of the hyperspectral image set as a whole. For use of the ANN as a classification model, the classification accuracy remained fairly similar to that of the full hyperspectral image data, with exceptions in using MVPCA for certain values and with VGBS for 15 and 30 band selection values. There was also a significant decrease in run-time with the use of band selection techniques with decreases as high as 51.45% when tested on MATLAB and 70.66% when tested on NVIDIA with the lower band selected values. With the SVM classification model, there was an even smaller difference between the classification performance of the full hyperspectral image data compared to the band down-selected data. These outputs show that the resulting tests can reflect those of real-time data collection and processing. The NVIDIA Jetson AGX Xavier produced similar, or sometimes faster, run-times than the simulation-run tests, which does not correlate in terms of efficiency over the simulation-run tests but instead demonstrates the plausibility of the NVIDIA device acting as the edge device. With that established, the classification accuracies and run-times of the classification models placed on the NVIDIA Jetson AGX Xavier can also be seen as a further confirmation of the effectiveness of these down-selected models. For example, when 15 bands were selected for the ANN, all of the band selected models performed 64.41% to 70.66% faster than that of the original hyperspectral image data, which is a considerable increase in speed. In future work, this research could be expanded upon to include more band selection techniques and additional hyperspectral image data. This work could also be placed onto a UAV for documentation of performance as data is collected.

REFERENCES

[1] T. Alipourfard, H. Arefi, and S. Mahmoudi, "A Novel Deep Learning Framework by Combination of Subspace-Based Feature Extraction and Convolutional Neural Networks for Hyperspectral Images Classification," *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 4780–4783.

[2] J.-P. Ardouin, J. Levesque, and T. A. Rea, "A demonstration of hyperspectral image exploitation for military applications," *2007 10th International Conference on Information Fusion*, 2007, pp. 1–8.

[3] J. Bai, F. Lu, K. Zhang, et al., "ONNX: Open Neural Network Exchange,", https://github.com/onnx/onnx, 2019.

[4] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3,", Sep 2015.

[5] S. Butler, S. R. Price, A. Gurbuz, S. J. Price, and J. R. Fairley, "Investigation of band selection techniques to enable classification of hyperspectral imagery at the edge," *Real-Time Image Processing and Deep Learning 2021*. International Society for Optics and Photonics, 2021, vol. 11736, p. 117360L.

[6] C.-I. Chang, *Hyperspectral imaging: techniques for spectral detection and classification*, vol. 1, Springer Science & Business Media, 2003.

[7] C.-I. Chang, Q. Du, T.-L. Sun, and M. Althouse, "A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 6, 1999, pp. 2631–2641.

[8] C. Cortes and V. Vapnik, "Support-vector network," *Machine Learning*, vol. 20, 1995, pp. 273–297.

[9] I. Dopido, A. Villa, A. Plaza, and P. Gamba, "A Quantitative and Comparative Assessment of Unmixing-Based Feature Extraction Techniques for Hyperspectral Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, 2012, pp. 421–435.

[10] Q. Du and H. Yang, "Similarity-based unsupervised band selection for hyperspectral image analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 4, 2008, pp. 564–568.

[11] E. Elhamifar and R. Vidal, “Sparse subspace clustering,” *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.

[12] D. Franklin, “Jetson AGX Xavier,”, September 2021.

[13] V. S. K. G and V. S, “Clustering Based Band Selection for Endmember Extraction Using Simplex Growing Algorithm in Hyperspectral Images,” *Multimedia Tools Appl.*, vol. 76, no. 6, Mar. 2017, p. 8355–8371.

[14] J. Gao, Z. Chen, L. Gao, and B. Zhang, “GPU implementation of ant colony optimization-based band selections for hyperspectral data classification,” *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2016, pp. 1–4.

[15] X. Geng, K. Sun, L. Ji, and Y. Zhao, “A Fast Volume-Gradient-Based Band Selection Method for Hyperspectral Image,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, 2014, pp. 7111–7119.

[16] A. Gowen, C. O’Donnell, P. Cullen, G. Downey, and J. Frias, “Hyperspectral imaging – an emerging process analytical tool for food quality and safety control,” *Trends in Food Science and Technology*, vol. 18, no. 12, 2007, pp. 590–598.

[17] M. Grana, M. Veganzons, and B. Ayerdi, “Hyperspectral Remote Sensing Scenes,”.

[18] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, 2002, pp. 415–425.

[19] A. Ifarraguerri and M. Prairie, “Visual method for spectral band selection,” *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 2, 2004, pp. 101–106.

[20] R. P. Jefferis and K. A. Fegley, “Application of dynamic programming to routing problems,” *IEEE Trans. Syst. Sci. Cybern*, vol. SSC-1, no. 1, 1965, p. 21–26.

[21] S. Jia, Y. Qian, and Z. Ji, “Band Selection for Hyperspectral Imagery Using Affinity Propagation,” *2008 Digital Image Computing: Techniques and Applications*, 2008, pp. 137–141.

[22] S. Jia, G. Tang, J. Zhu, and Q. Li, “A Novel Ranking-Based Clustering Approach for Hyperspectral Band Selection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, 2016, pp. 88–102.

[23] R. Larson, “Dynamic programming with reduced computational requirements,” ” *IEEE Trans. Autom. Contro*, vol. AC-10, no. 2, 1965, p. 135–143.

[24] S. Li, H. Wu, D. Wan, and J. Zhu, “An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine,” *Knowledge-Based Systems*, vol. 24, no. 1, 2011, pp. 40–48.

[25] S. Li, Y. Zhu, D. Wan, and J. Feng, "Hyperspectral band selection from the spectral similarity perspective," *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, 2013, pp. 410–413.

[26] G. Lu and B. Fei, "Medical hyperspectral imaging: a review," *Journal of Biomedical Optics*, vol. 19, no. 1, 2014, pp. 1 – 24.

[27] L. Luft, C. Neumann, M. Freude, N. Blaum, and F. Jeltsch, "Hyperspectral modeling of ecological indicators – A new approach for monitoring former military training areas," *Ecological Indicators*, vol. 46, 2014, pp. 264–285.

[28] A. MartÍnez-UsÓMartinez-Uso, F. Pla, J. M. Sotoca, and P. García-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, 2007, pp. 4158–4171.

[29] D. Nakaya, Y. Tomiyama, S. Satori, M. Saegusa, T. Yoshida, A. Yokoi, and M. Kano, "Development of High-Performance Pathological Diagnosis Software Using a Hyperspectral Camera," *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 2018, pp. 217–220.

[30] J. Oehlschläger, U. Schmidhalter, and P. Noack, "UAV-Based Hyperspectral Sensing for Yield Prediction in Winter Barley," *2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2018, pp. 1–4.

[31] A. Paz and A. Plaza, "Cluster versus GPU implementation of an Orthogonal Target Detection Algorithm for Remotely Sensed Hyperspectral Images," *2010 IEEE International Conference on Cluster Computing*, 2010, pp. 227–234.

[32] J. Qin, K. Chao, M. S. Kim, R. Lu, and T. F. Burks, "Hyperspectral and multispectral imaging for evaluating food safety and quality," *Journal of Food Engineering*, vol. 118, no. 2, 2013, pp. 157–171.

[33] U. Seiffert, F. Bollenbeck, H.-P. Mock, and A. Matros, "Clustering of crop phenotypes by means of hyperspectral signatures using artificial neural networks," *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 2010, pp. 1–4.

[34] M. Shimoni, R. Haelterman, and C. Perneel, "Hypersectral Imaging for Military and Security Applications: Combining Myriad Processing and Sensing Techniques," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, 2019, pp. 101–117.

[35] R. Siche, R. Vejarano, V. Aredo, L. Velasquez, E. Saldaña, and R. Quevedo, "Evaluation of Food Quality and Safety with Hyperspectral Imaging (HSI)," *Food Engineering Reviews*, vol. 8, no. 3, 2015, p. 306–322.

[36] H. Su, Q. Du, G. Chen, and P. Du, "Optimized hyperspectral band selection using particle swarm optimization," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, 2014, pp. 2659–2670.

[37] P. Su, D. Liu, X. Li, and Z. Liu, "A Saliency-Based Band Selection Approach for Hyperspectral Imagery Inspired by Scale Selection," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 4, 2018, pp. 572–576.

[38] W. Sun and Q. Du, "Hyperspectral Band Selection: A Review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, 2019, pp. 118–139.

[39] W. Sun, G. Yang, and J. Li, "Robust Multi-Feature Spectral Clusteirng for Hyperspectral Band Selection," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 3800–3803.

[40] W. Sun, L. Zhang, B. Du, W. Li, and Y. Mark Lai, "Band Selection Using Improved Sparse Subspace Clustering for Hyperspectral Imagery Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, 2015, pp. 2784–2797.

[41] K. Tiwari, M. Arora, and D. Singh, "An assessment of independent component analysis for detection of military targets from hyperspectral images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 13, no. 5, 2011, pp. 730–740.

[42] E. Torti, G. Danese, F. Leporati, and A. Plaza, "A Hybrid CPU–GPU Real-Time Hyperspectral Unmixing Chain," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 2, 2016, pp. 945–951.

[43] Q. Wang, F. Zhang, and X. Li, "Optimal Clustering Framework for Hyperspectral Band Selection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, 2018, pp. 5910–5922.

[44] W. Wei, Q. Du, and N. H. Younan, "Fast supervised hyperspectral band selection using graphics processing unit," *Journal of Applied Remote Sensing*, vol. 6, no. 1, 2012, p. 061504.

[45] L. Wenjing and W. Xiaofei, "Overview of Hyperspectral Image Classification," *Journal of Sensors*, vol. 2020, Jul 2020.

[46] C. Yan, X. Bai, and J. Zhou, "Band weighting and selection based on hyperplane margin maximization for hyperspectral image classification," *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 1702–1705.

[47] H. Yang, Q. Du, and G. Chen, "Unsupervised hyperspectral band selection using graphics processing units," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 3, 2011, pp. 660–668.

[48] B. Zhang, "Hyperspectral remote sensing of vegetation growing condition and regional environment," *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 2010, pp. 1–4.