

AMERICAN UNIVERSITY OF BEIRUT

OPTIMAL SPATIAL-SPECTRAL INPUT FOR
REAL-TIME HYPERSPECTRAL IMAGE
CLASSIFICATION

by

JAWAD SALAH HAIDAR

A thesis

submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Mechanical Engineering
of Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
September 2, 2022

AMERICAN UNIVERSITY OF BEIRUT

OPTIMAL SPATIAL-SPECTRAL INPUT FOR
REAL-TIME HYPERSPECTRAL IMAGE
CLASSIFICATION

by

JAWAD SALAH HAIDAR

Approved by:

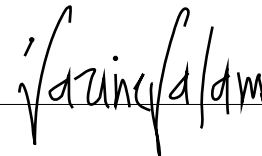
Samir
Mustapha

Digitally signed by Samir Mustapha
DN: cn=Samir Mustapha,
o=American University of Beirut,
ou=Mechanical Engineering,
email=sm154@aub.edu.lb, c=LB
Date: 2022.09.09 11:18:51 +03'00'

Dr. Samir Mustapha, Associate Professor

Advisor

Mechanical Engineering



Dr. Darine Salam, Associate Professor

Member of Committee

Civil and Environmental Engineering

Ali Tehrani

Digitally signed by Ali Tehrani
DN: cn=Ali Tehrani, c=FI, o=Aalto
University, ou=Bioproducts and
Biosystems,
email=ali.tehrani@aalto.fi
Date: 2022.09.09 13:42:00 +03'00'

Dr. Ali Tehrani, Associate Professor

Member of Committee

Chemical Engineering

Date of thesis defense: September 2, 2022

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Samir Mustapha for the continuous support of my thesis work, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Darine Salam, and Dr. Ali Tehrani, for their encouragement, insightful comments, and constructive feedback.

ABSTRACT

OF THE THESIS OF

Jawad Salah Haidar for Master of Engineering
Major: Mechanical Engineering

Title: Optimal Spatial-Spectral Input For Real-Time Hyperspectral Image Classification

Heavy metal contamination in soil is of great danger for the environment as well as for the human health. This is owed to their hazardous toxicity alongside with their ability be easily captured by crops and plants. Detection of soil contamination is a major step that aids in the soil rehabilitation process. Our goal in this project is to explore the potential of using hyperspectral imaging and deep Learning techniques to identify contamination in real time. The large size of the hyperspectral image and the relatively limited resources in onboard processors make the real-time classification a challenging task. In this work, we propose a novel approach for real-time hyperspectral image classification using optimal spatial-spectral input. The optimal input consisted of the main pixel with two of its spatial neighbors. Two novel deep learning models based on the optimized three pixel method were developed. The first is a Deep Neural Network (DNN) model focused on fast online classification and the second is a Recurrent Neural Network (RNN) model focused on offline classification with enhanced accuracy. These models were evaluated using four datasets, three agricultural datasets and the Sydney Bridge dataset. The DNN model achieved a maximum accuracy of 97% with an inference speed of 83,665 pixels per second while the RNN model achieved a maximum accuracy of 99% with an inference speed of 25,000 pixel per second. When compared, the DNN model is more suited for quick real-time applications while the RNN model is more suited for applications where accuracy is critical. Furthermore, in terms of latency, our approach maximized the

preprocessing speed compared to other spatial-spectral methods. The attained fast speed in preprocessing and inference is applicable for Real-time application.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	1
ABSTRACT	2
ABBREVIATIONS	8
1 Introduction	9
2 Literature	11
2.1 Hyperspectral Cameras	11
2.2 Hyperspectral Image Processing	13
2.2.1 Correction Methods	13
2.2.2 Dimensionality Reduction	14
2.2.3 Spectral Unmixing	14
2.3 Hyperspectral Image Classification	15
2.4 Drone and Real-time Classification	17
2.5 Paint Condition Assessment	18
2.6 Spectral and Spatial Features	19
3 Preliminaries	21
3.1 Recurrent Neural Network	21
3.2 Long Short Term Memory	22
3.3 Gated Recurrent Units	23
4 Methodology	26
4.1 Preprocessing	26
4.1.1 Data Reduction Experiment	26
4.1.2 Reflectance Transform	27
4.2 DNN	27

4.3	RNN	29
5	Results and Discussion	32
5.1	Agriculture Case Study	32
5.1.1	Dataset	32
5.1.2	Model Accuracy	32
5.1.3	Preprocessing and Inference Speed	33
5.2	Paint condition Case Study	35
5.2.1	Datasets	35
5.2.2	Classification Results	35
5.2.3	Input Experiments	37
6	Conclusion	43
	Bibliography	45

ILLUSTRATIONS

2.1	Figure 1-Types of hyperspectral cameras. (a) Whiskbroom (b) Push Broom (c) Spectral scanning (d) Snapshot [15]	12
3.1	Block diagram of the RNN cell with tanh activation function.	22
3.2	Block diagram of the Long short term memory cell.	24
3.3	Block diagram of the Gated Recurrent Unit that includes sigmoid and tanh activation functions.	25
4.1	Pixel transformation from irradiance into reflectance	27
4.2	Recurrent Neural Network with GRU units.	30
5.1	Pavia dataset classification results	33
5.2	Indiana dataset classification results	33
5.3	Salinas dataset classification results	34
5.4	confusion matrices for short range models	37
5.5	Short Range classification results	38
5.6	confusion matrices for Long range models	39
5.7	Long Range classification results	40

TABLES

2.1	Specifications for hyperspectral cameras	13
2.2	Absorption bands for gases found in air in micrometer	13
5.1	Benchmark for spatial-spectral methods	35
5.2	DNN model performance on short range images	36
5.3	RNN model performance on short range images	36
5.4	DNN model performance on Long range images	37
5.5	RNN model performance on Long range images	39
5.6	score for the three pixel input versus the five pixel input	41

ABBREVIATIONS

UAV	Unmanned Ariel Vehicle
DNN	Deep Neural Network
RNN	Recurrent Neural Network
PCA	Principle Component Analysis
HSI	Hyper Spectral Image
VNIR	Visible Near Infrared
SWIR	Short Wave Infrared
SVM	Support Vector Machine
MVS	Mixed Volume Simplex
MLP	Multy Layer Perceptron
KNN	K Nearest Neighbour
CNN	Convolution Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory

CHAPTER 1

INTRODUCTION

Heavy metal contamination in soil is of great danger for the environment as well as for the human health. This is owed to their hazardous toxicity alongside with their ability be easily captured by crops and plants. Additionally, they might penetrate to groundwater and thus polluting the drinking water [1], [2]. In Lebanon, the waste mismanagement and uncontrolled dumping practices in addition to limited water and soil resources lead to a drastic increase in heavy metal contamination in various lands across the country [3], [4]. Various studies have been conducted across Lebanon especially in the bekaa region where they found medium to high heavy metal contamination levels that necessitates action [5], [6].

The critical contamination issue necessitates an efficient detection and monitoring method. One conventional technique is lab based; where soil samples are collected form situ, then chemical analysis is conducted on them at the lab, finally spatial interpolation is applied to estimate the metal contamination in the full spatial region of the land. The downside of this method is time consumption and high cost of operation. A more efficient method that meet the standards of areas with large scale is hyperspectral method [7]. Hyperspectral images have a high spectral resolution with bands ranging from 20 to 500 nm, these types of images have enough information about the soil health and type. To detect soil contamination from hyperspectral images, machine learning algorithms should be applied to perform classification. Some of the most used classification methods are Support Vector Machine (SVM), random forest and artificial neural networks [8], [9].

These spectral cameras are mounted on Unmanned Aerial Vehicles (UAVs) that can maneuver and scan the field with eye bird view thus providing a good field of view in addition to the flexibility of varying the height of capture [10]. During the flight the hyperspectral images can be saved to memory and then analyzed offline on a

high-performance computer. The downside for this method is that the user will not get access to the results of classification until the flight is done. On the other hand, a real time onboard classification will give the user insights during the flight and thus providing the potential for instantaneous decisions [11]. The complexity in real time onboard classification arise from the fact that the hyperspectral image has a huge volume along with the limited computational power for onboard computers as well as limited transmission speed [12], [13].

Given the importance of performing hyperspectral image classification in real-time, our goal in this study is to develop an optimized deep learning model that can perform onboard hyperspectral classification. The models will be general and able to be used in any domain including soil contamination. To test the models two case studies will be conducted. The first will be conducted on a agricultural dataset to classify agricultural lands. The second case study will be performed on a structural health monitoring dataset. The latter dataset includes hyperspectral images of the Sydney bridge and is used for paint condition assessment.

CHAPTER 2

LITERATURE

2.1 Hyperspectral Cameras

One way to categorize Hyperspectral cameras is by the way they capture the spectral data. In this sense, they are divided into four categories: Push broom, whiskbroom, framing and windowing. Push broom captures the data line by line, with the lines being perpendicular to the direction of motion. Whiskbroom records the hyperspectral data pixel wise by sweeping perpendicular to the motion of movement by the aid of rotating mirror. On the other hand, the framing method stacks two dimensional images, to form the hyperspectral cube, by focusing on one wavelength per capture. In the case of windowing, no integration is established between the consecutive 2D images [14]. The four categories of hyperspectral cameras are shown in figure 2.1.

There are several methods to capture a hyperspectral image. The whiskbroom method scans pixel by pixel and uses a mirror that sweeps, perpendicular to the motion direction of the UAV, to form a scanned line. On the other hand, the push broom method scans a line of pixels at one time, this line forms a 2d array with a spectral dimension and a spatial dimension that is perpendicular to the motion. As the UAV moves these 2D arrays are stacked to form the 3D spectral image. A critical point to keep in mind is that the speed of the array capturing should be synchronized with the speed of the UAV. A third method uses a tunable filter that forms a 2d array for the whole scene for one band at a time. Lastly, the snapshot method, capture the entire 3D cube at one shot [16]. Hyperspectral cameras can be classified also by the spectral range that they cover. Tan et al. [9], used two hyperspectral cameras one to cover the near infrared range and the other to cover short wavelength infrared range. Using this camera, they were able to detect several heavy metal contaminations in soil such as As, Cr, Pb and Zn. On the other hand

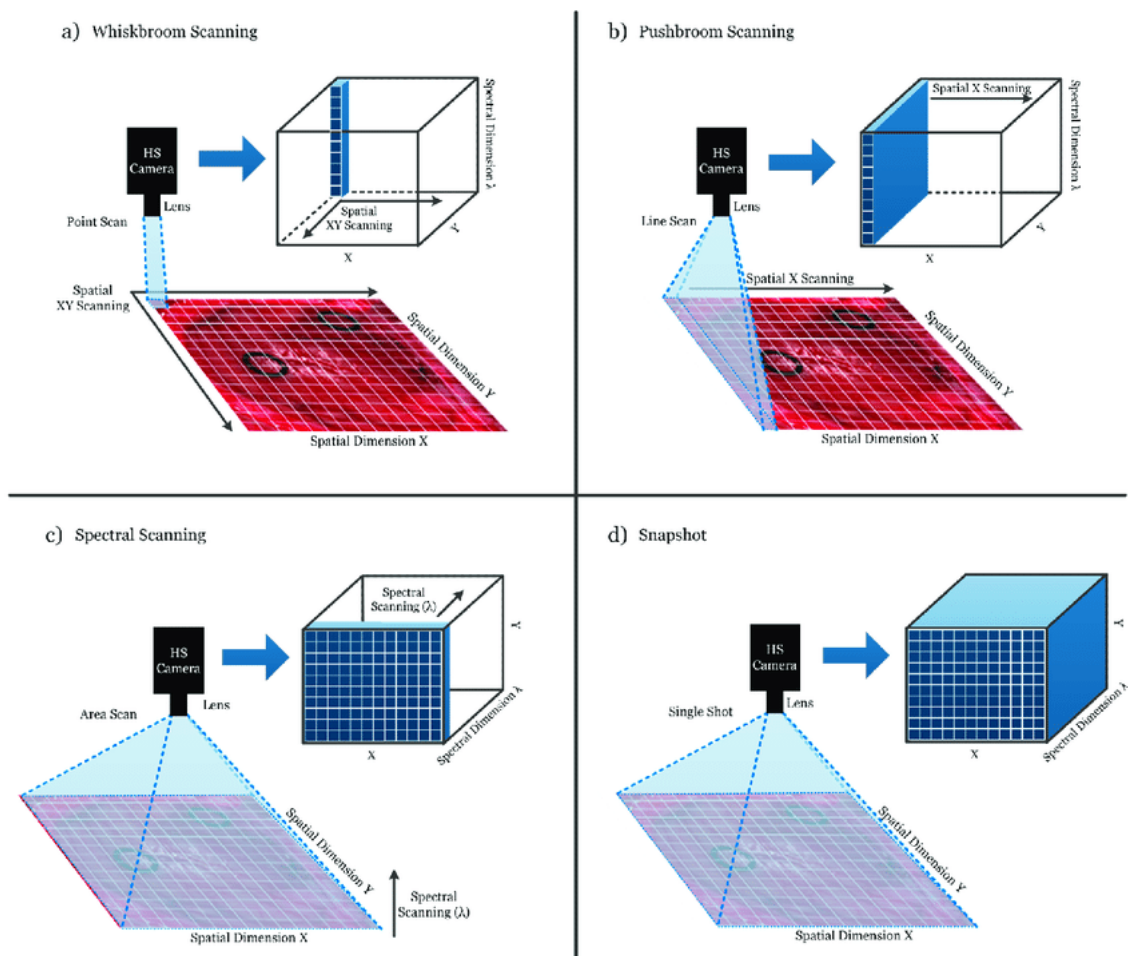


Figure 2.1: Figure 1-Types of hyperspectral cameras. (a) Whiskbroom (b) Push Broom (c) Spectral scanning (d) Snapshot [15]

Table 2.1: Specifications for hyperspectral cameras

	Application	Weight (Kg)	Spectral Range (nm)	Spectral Channels	Spatial Pixels	Power (W)
VNIR-1800	Airborne/field	5	400-1000	186	1800	30
SWIR-384	Airborne/field	5.7	930-2500	288	384	30
Mjolnir V-1240	UAV	<4	400-1000	200	1240	50
Mjolnir S-620	UAV	<4	970-2500	300	620	50

Table 2.2: Absorption bands for gases found in air in micrometer

Water Vapor	1.88 , 1.38 , 0.94
Oxygen	0.76
Carbon dioxide	2.08

the authors in [17] , used a violet near infrared spectral camera that was mounted on a UAV. A detailed classification of different hyperspectral cameras is found in table 2.1. VNIR-1800 is a hyperspectral camera that covers the visible and near infrared region. Whereas the SWIP-384 covers the short-wave infrared spectrum. The Mjolnir V-1200 and Mjolnir S-620 are usually mounted on UAV. The former covers the visible near infrared whereas the later covers the short-wave infrared region.

2.2 Hyperspectral Image Processing

2.2.1 Correction Methods

Atmospheric noise is the contribution of different atmospheric factors to the spectral signal [18]. The process of transferring the radiance gauged by the camera into the real reflectance of the land/surface is called atmospheric correction. Table 2.2 shows the absorption band of different atmospheric gases that affect the spectrum. Some of these gases are water vapor, oxygen, and carbon dioxide. The algorithms for hyperspectral image classification should account for these bands by removing or filtering them from the spectral curve. The atmospheric correction approaches can be divided into three different approaches. The empirical approach that includes several methods as Internal Average Reflectance (IAR), empirical line and cloud shadow. The radiative transfer modeling simulates the scatter and absorption effects of aerosols and gases using theoretical modelling.

2.2.2 Dimensionality Reduction

There are three types of dimensionality reduction techniques. Feature selection works on eliminating bands that do not contribute to giving the pixel a unique representation, in other words, keeping the bands that identifies the spectral signature of the pixel. The feature selection method works when the user already knows where the spectral signature of a certain material in the electromagnetic spectrum is. The feature selection method is also valid in the case of unsupervised learning by selecting the bands that maximizes the distance between the different clusters of the image. Another method is called the feature extraction, in this method the objective is to transform the input data to a new space that reduces the dimensions of the input vectors and at the same time keeping the most crucial information. One main algorithm for feature extraction is called the principal component analysis. This algorithm projects the spectral bands into a smaller N dimensional space while preserving the variance in the data. The N dimensions are the number of principle components, the first principal component captures the maximum data variation. Additionally, there is the physical indices calculation techniques that creates features with physical interpretations like the normalized difference vegetation index, normalized difference vegetation index and the floating Algae Index [1].

2.2.3 Spectral Unmixing

The solar radiation and ambient environment cause disturbances in the measured reflectance [19]. The spectral mix is common especially in the case where the image is captured at a very high altitude since in this case each pixel will cover a huge area in the land. Additionally, the lack of high spectral resolution causes a geographic mixing, in other words the spatial dimension of one pixel is large enough to include several materials per pixel. Endmembers stands for the different sources that formed the spectra of a pixel. On the other hand, the abundance is the percentage of each endmember. In geographic mixing different materials are found in one pixel but they are not mixed. This kind of mixing can be resolved using Linear Mixing Models. Whereas in the case of intimate granual mixture the materials are mixed, this issue is usually solved using nonlinear mixing models. The last type of mixture is called the intimate laminate mixture, where different materials are stacked above each other. The estimation of the count of endmembers in the whole image is the first step taken in spectral unmixing. The techniques in this matter are split into three: the supervised where the number of endmembers is chosen to be the first PCA

components that span most of the variance. The semi-automatic that subtract the correlation and covariance matrices, then counts the non-zero elements and sets this number to be the count of endmembers. The unsupervised that deducts the value from the correlation between signal and noise [1].

2.3 Hyperspectral Image Classification

Turner et al. [17], used two different airframes to capture hyperspectral images for a field. The first airframe included a micro-hyperspec, Inertial Measurement Unit (IMU), Global Positioning System (GPS) and a machine vision camera whereas the second airframe contained a Nano-hyperspec, IMU and GPS only. The results of the experiment showed that machine vision cameras and SFM can be used as an alternative to determine the orientation of the UAV rather than the convention of using an IMU. The authors in [9], got 60 soil samples and determined the concentration of pb, Cr and Cu) . Two hyperspectral cameras VNIR-1600 and SWIR-384 were used to capture hyperspectral images followed geometric and radiation corrections, and due to special resolution roughness, they used the minimum volume simplex to perform linear unmixing. The MVS is considered as a fast process to apply linear unmixing for the hyperspectral data [20]. Finally, they performed several classification models and found out that the random forest performed the best. The authors in [7], stated that one of the most challenging tasks in heavy metal detection using a hyperspectral camera is to derive a lab mathematical model, using samples of soil, that can be applied to detect heavy metals in a real world scenario as capturing the images from a UAV. The challenge arises due to the fact that lab is a controlled environment and does not account for random noise that might be encountered by a UAV. For this reason, they developed an innovative method that derives the ratio between dry and moisture soil spectral reflectance. Consequently, this has enhanced the potential of using lab models to map lands with very large areas. Furthermore, using the combination of Boruta model and stepwise regression, the best spectral variables were chosen. They have tested their method to detect three different metals: Cd, As, and Hg. The approach to predict the contamination of heavy metals inside soils follows the following steps. The first step is to collect the soil samples from the area of study. Then the spectra of the samples are measured and preprocessed. At this stage, there is three different options to the independent variables that will act as an input to the prediction model, either to extract the spectral indices or select the information variables or to take the full spectra as the

input from the other side, the label for the inputs, is taken from the real concentrations of the samples. Then a regression relation is derived, using machine learning tools, between the independent variables and the labels to form a regression model. Finally, new samples can be introduced to the model to predict their concentrations [21]. The authors in [1], divided their machine learning model into three parts, Input layer that includes the high-resolution airborne image labeled at 1068 points. The labeling process was established by gathering 1068 soil samples and measuring their concentration in lab, the coordinates for each label was extracted using GPS. Second part was the feature extraction layer, the features they have chosen can be divided into three sets: spectral set, distance, and gradient from pixels to rivers and factories, and the third set is the distance and gradient to waste storage areas. The third part is the prediction part, at this stage they have performed several machine learning models (MLP, SVM, Random Forests, Extreme Random Forest) to classify the test pixels into the three different risk levels (Low, Medium, High). Additionally, the researchers in this paper found that the spectral related features have the least influence on prediction. In the context of hyperspectral data two main issues arise, from one side the small sample size and the high dimensional feature vectors from the other. Solving the former issue is done using ensemble methods to generalize the data, and the latter issue is solved by performing the Competitive Adaptive Reweighted Sampling feature selection method. Tan, Wang, Chen, *et al.* [9], started by applying competitive reweighted sampling to select the best representing features for the spectral signatures, then the feature selected data were the input for a stacking method, the stacking method started with the cross validation of the data with three independent prediction models (decision forest, KNN, SVM), the three outputs of the prediction models were the input to the extremely randomized trees. Yu, Fang, and Zhao [22], used PCA to choose the most effective wavelengths in the spectrum range of the Hyperspectral camera. The spectral response of several metals is not unique, in such cases the detection will rely on correlation of the metals with organic matters that might be found in soil. Other metals do not absorb the spectral waves, detecting their presence is attained by their covariation with other detectable metals. Additionally, metals in soil mostly exist in the compound not the ion form [2].

2.4 Drone and Real-time Classification

Three main types of UAVs: fixed wings, VTOL and rotorcraft. Such drones can be used in various civil infrastructure applications like Ariel inspection, ariel survey, search and many others [23]. Wyder, Chen, Lasrado, *et al.* [24] developed an autonomous UAV that detects and takes down other UAVs in regions with no GPS signals. They generated 58647 images. 10,000 of them were generated in Air-Sim simulator. The images were changed to gray scale to reduce computational power and then trained using YOLO detection algorithm. Using the Jetson TX2 as the onboard hardware they achieved 77% of accuracy at 8 frames per second. The authors in [25], worked on road inspection using low power drone with real time classification. They used MobileNet-V2 that is a CNN used for onboard mobile platforms. The framework has good accuracy while running on low-power platforms. For training, they used 21,890 images divided into 14 classes of road asset items. They achieved 81% accuracy with 7.4 fps and consuming 1.9 watts for the onboard jetson Nano. Waleed, Mukhopadhyay, Tariq, *et al.* [26] worked on outdoor ceramic insulator condition monitoring using drone with raspberry Pi and Odroid XU4 as onboard computers. In the training stage 2973 images were used for three classes: health, broken and dirty. They performed their application using both onboard and offboard high performing computer. For offboard, SVM combined with CNN is used to determine the region of interest. On the other hand, the combination of SVM and CNN are computationally heavy to be applied on onboard, so they used a light process which is single shot multi-box detector. To improve the performance, they used mobilenetv2 and consequently achieved a processing time of image per 2 seconds. One of the main architectures used for onboard computer vision applications is Mobile Nets. The authors in [27], have developed this architecture using lightweight model by utilizing depth wise separable convolutions. It can be used for object detection, geo-localization and other computer vision applications. MobileNetV2 is version two of the previously mentioned model, it includes depth wise convolution network with 32 filters. The convolution layers are followed by 19 residual bottleneck layers. This model outperforms the mobileNetv1 with accuracy of 74.7 74.7%. But the parameters in this model are 6.9 million whereas in mobileV1 they are only 4.2 million [28]. Using the Indian pines hyperspectral image dataset, the jetson TX2 showed a better performance to energy ration over the Xeon hardware during the training phase [29]. The boards were not tested in the inference phase to check their performance for real time applications. The au-

thors in [30], benchmarked the jetson TX1, Jetson Nano against the jetson Xavier by testing their computational capabilities to perform classification for both hyperspectral image and 3D point clouds. For 3D point cloud computation, the boards were benchmarked in the inference phase as well as in the training phase whereas for the HIS classification the boards were tested only in the training phase. They concluded that Xavier outperforms both jetson TX1 and jetson Nano and that it is suitable for Realtime edge computing. A real time target detection algorithm for hyperspectral images using FPGA was developed by the researchers in [36]. The Authors used the CRD algorithm that shrinks the dimensions of the data as its first step to lower the computation. they achieved faster processing time compared to 3.40 GHz CPU. The authors in [8], introduced a hardware accelerator to enhance the classification computation for hyperspectral images. The authors applied support vector machine and their proposed processor was implemented on FPGA. A pixel classification of 0.1 ms with high accuracy of 99.7% was achieved.

2.5 Paint Condition Assessment

Before the spread of more advanced techniques, previous paint condition assessment applications relied on traditional machine learning methods. One such study trained a multi-class Support Vector Machines classifier with a linear kernel to assess the paint condition on civil structures based on spectral input data [31]. Despite the classifier being able to identify certain classes accurately, it achieved a fair overall classification accuracy of 75% and suffered from confusion between classes [31]. Even recently, decision tree models were trained with the CatBoost algorithm as degradation models that could predict the retroreflectivity of waterborne paints based on the significant variables that are believed to affect the performance of these paints [32]. Their models achieved acceptable levels of accuracy but require further tweaking and tuning of hyperparameters to enhance performance to more desirable levels [32]. These studies highlight some of the drawbacks of traditional machine learning techniques which include the many model parameters that need to be tuned to the specific application to get desirable performance and shortcomings when dealing with high dimensional data and data that exhibit nonlinear properties. For that reason, recent paint assessment studies have become increasingly more reliant on advanced machine learning techniques, mainly deep learning neural networks. Deep learning models can be trained to represent the high dimensional features in huge amount of data which traditional models are unable to describe while often being

coupled with imaging data [33]. Recently, a deep learning-convolutional neural network model was trained on images of fatigue induced damage in structural steel fuse members for structural steel damage condition assessment which resulted in an overall 99.87% testing accuracy [33]. The study developed a general procedure for implementing the deep learning model for structural condition assessment applications which involves a large dataset size, a quantitative damage index to label the images, guided backpropagation for training, and saliency maps for the visual evaluation of results [33]. A particular interest for researchers has been the monitoring of wind turbine conditions and recently, a data-oriented approach has been developed to utilize digital data for the monitoring and maintenance planning of surface protective coating systems of large onshore wind turbines [34]. The approach involves segmentation of reference areas on the turbine surface and then condition assessment of the segmented reference areas for screening and classification of damage regions using computer vision and deep learning networks [34].

2.6 Spectral and Spatial Features

Spectroscopy, particularly near infrared spectroscopy, has had several practical applications, mainly in the fields of agriculture, food science and physical chemistry which have been widely reported in literature [35]. Recently, forensic science has also benefited from infrared spectroscopy which was evaluated for its ability to discriminate among paint samples such as spray paint and automotive paint in forensic investigations by identifying necessary spectral features [36]. However, spectroscopy alone is limited in its application due to its failure to give information on spatial distribution of the spectra obtained [37]. To form hyperspectral imaging, spectroscopy was combined with conventional imaging to give spatial information of the spectra within a two-dimensional image plane [37]. In an aforementioned study using spectral data to assess paint condition on civil structures, the data used to train the model was a set hyperspectral images taken of the Sydney Harbour Bridge such that a single training instance is the spectrum of a single pixel in an image [31]. Despite the clear advantages of this method including the visualization and localization of the paint condition on the structure through paint condition rating maps, the spatial information of hyperspectral imaging data was not used to train the classification model which gives pixel-wise predictions [31]. Spatial dependency is complementary to spectral behavior and is naturally another useful information source in addition to spectra and so incorporating spatial information offers the pos-

sibility to boost the pixel-wise analysis [38]. Spectral-spatial feature networks are deep networks that can make use of both spectral and spatial data in hyperspectral images by extracting both features simultaneously and using them for predictions [39]. One such study utilized a three-dimensional deep Convolutional Neural Network with hyperspectral image data for spatial-spectral resolution of hyperspectral images by learning end-to-end with mapping between low and high spatial resolution hyperspectral images with high accuracy [40]. A proposed 3-D deep CNN for hyperspectral image classification showed excellent performance under the condition of limited training samples and outperformed other models that did not utilize both spectral and spatial features including a single dimensional and two-dimensional CNN [41]. However, spectral-spatial feature networks in the form of full 3-D deep convolutional neural networks are easily prone to overfitting and have the highest computational costs and processing times [41].

As mentioned previously, several methods were proposed for real-time classification using only spectral input. Adding spatial information enhances the performance of the model but adds a computational burden that might prevent it from the real-time application. In this work, we propose an optimal input that will include the spatial information by only adding two of the adjacent pixels. Such input will be applicable to two different models: DNN and RNN. For this reason, we will pass this input to the two models and benchmark them against each other and against previous literature. The objectives of this thesis can be summed up as follows :

- Develop an optimal spectral-spatial input for hyperspectral image classification. The input will enhance the model performance with minimal computational burden.
- Utilize the optimal input to construct a light weight DNN and RNN models that require minimal computational power and can be embedded on onboard processors .
- Prove that the models are not biased by evaluating them using different datasets. Two case studies will be conducted, the first on agricultural dataset and the second on a paint condition dataset.

CHAPTER 3

PRELIMINARIES

There exist three main implementations for RNN models. The standard RNN, RNN with GRU units and RNN with LSTM units. In this section, a detailed explanation of each type with its mathematical formulation will be explained. By the end, the type with the most advantages will be used.

3.1 Recurrent Neural Network

An RNN is a type of Artificial Neural Networks. It differs from the standard ANN in its relation to the past input. The output is not only related to the current input as in the case of the ANN but also to the previous inputs [42], [43].

There are several types of RNN. One to one, with a single input and single output, that acts as a standard RNN. One to Many which takes a single input and generates several outputs and this is used mainly in music generation and image captioning. Many to Many that takes several inputs and generate several outputs which is very useful for language translation tasks. Finally, there is the many to one type that we will use in our paper since its major use case is in classification tasks. The ability to benefit from past inputs lies in the concept of the hidden state. The activation units in the current state are a function of the current input and the hidden state that transforms information from the previous time step. Let $x = [x^1, x^2, \dots, x^k]^T$ be a sequence sample of length k. As seen in figure, the hidden state and the current input will be feed to equation 3.1 to get the next hidden state.

$$h(h^{t-1}, x^t) = \sigma(W_h x^t + U_h h^{t-1} + b_h) \quad (3.1)$$

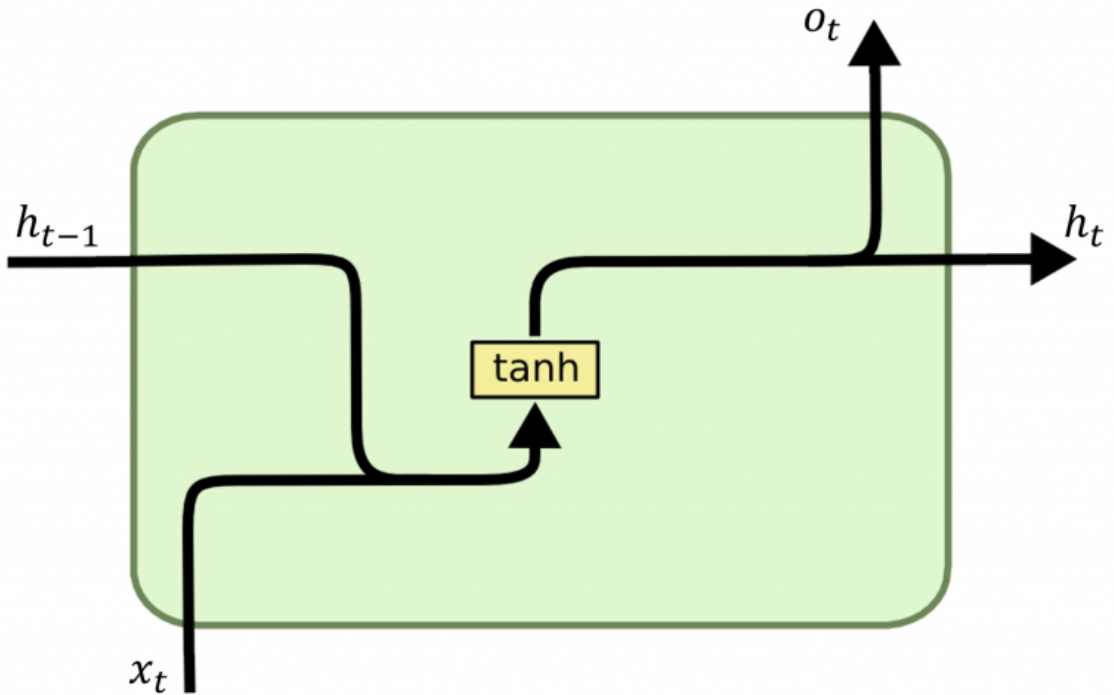


Figure 3.1: Block diagram of the RNN cell with tanh activation function.

Additionally the output h_t is feed to equation 3.2 to get the output y_t .

$$y(h^t) = W_y h^t + b_y \quad (3.2)$$

where W_h and b_h stands for the weights and bias of the input to hidden transformation. Similarly, W_y and b_y stands for the weights and bias of the hidden to output transformation. U_h is the weight matrix for the hidden to hidden transform and σ is the activation function.

Although RNN is able to connect information from the past using hidden states but when it comes to long sequences RNN face the problem of vanishing gradients [44]. In other words, during backpropagation there is a high probability for the gradients to zero out due to long sequence.

3.2 Long Short Term Memory

The solution to the vanishing gradients is in LSTM that was first introduced by [45]. Rather than relying on the hidden state, it also creates a memory cell c_t that helps in preserving long term memory. At each time step a candidate memory cell

\tilde{C}_t .

$$\tilde{C}_t = \tanh(W_{ci}x_t + W_{ch}h_{t-1} + b_c) \quad (3.3)$$

This candidate memory updates the memory cell in accordance with the input and forget gate.

$$c_t = i_t \odot \tilde{C}_t + f_t \odot c_{t-1} \quad (3.4)$$

The forget gate f_t specifies the degree to which the past memory should be forgotten. Whereas the input gate i_t specifies to what extent the candidate memory should be represented in the current memory.

$$i_t = \sigma(W_{ii}x_t + W_{ih}h_{t-1} + b_i) \quad (3.5)$$

$$f_t = \sigma(W_{fi}x_t + W_{fh}h_{t-1} + b_f) \quad (3.6)$$

The last gate is the output gate o_t and is formulated as follows :

$$o_t = \sigma(W_{oi}x_t + W_{oh}h_{t-1} + b_o) \quad (3.7)$$

Finally, the activation h_t is calculated using the o_t and c_t as follows:

$$h_t = o_t \tanh(c_t) \quad (3.8)$$

3.3 Gated Recurrent Units

The GRU is a variant of LSTM that was first introduced in 2014 [46]. Compared to LSTM this variant has this is a simplified version with less parameters and this feature makes it easier to train plus needs less input data to be able to generalize. In terms of HSI classification, it is proven that is more suitable to use GRU over LSTM [47]. There are two main differences between the GRU and LSTM. The first one is that the former has only two gates, update gate and reset gate, instead of three gates. Additionally, GRU doesn't have a cell state. The equations are as follows :

$$z_t = \sigma(W_{zi}x_t + W_{zh}h_{t-1} + b_z) \quad (3.9)$$

$$r_t = \sigma(W_{ri}x_t + W_{rh}h_{t-1} + b_r) \quad (3.10)$$

$$\tilde{h}_t = \tanh(W_{hi}x_t + W_{rh}(r_t * h_{t-1}) + b_h) \quad (3.11)$$

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \quad (3.12)$$

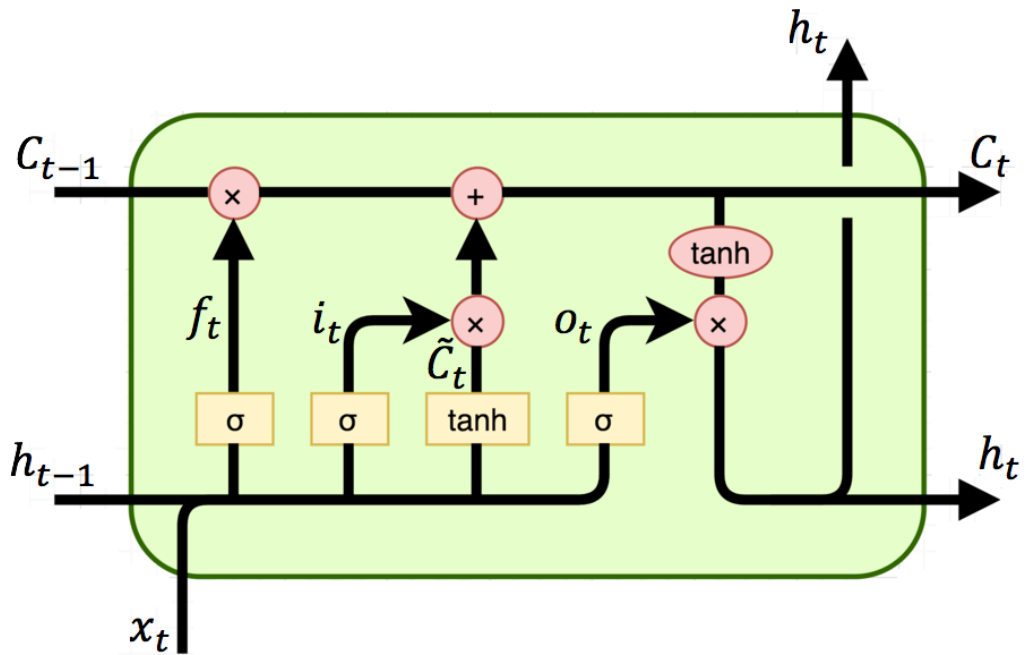


Figure 3.2: Block diagram of the Long short term memory cell.

All the W s stand for the weight matrices. The role of the update gate shown in equation 3.9 is to decide how much should be preserved from the previous memory. Whereas the reset gate shown in equation 3.10 decides the information that should be removed.

The fact that the standard RNN might encounter the zero gradient issue, in addition, to the fact that the GRU can generalize with less data compared to LSTM, the GRU will be used to perform the classification task.

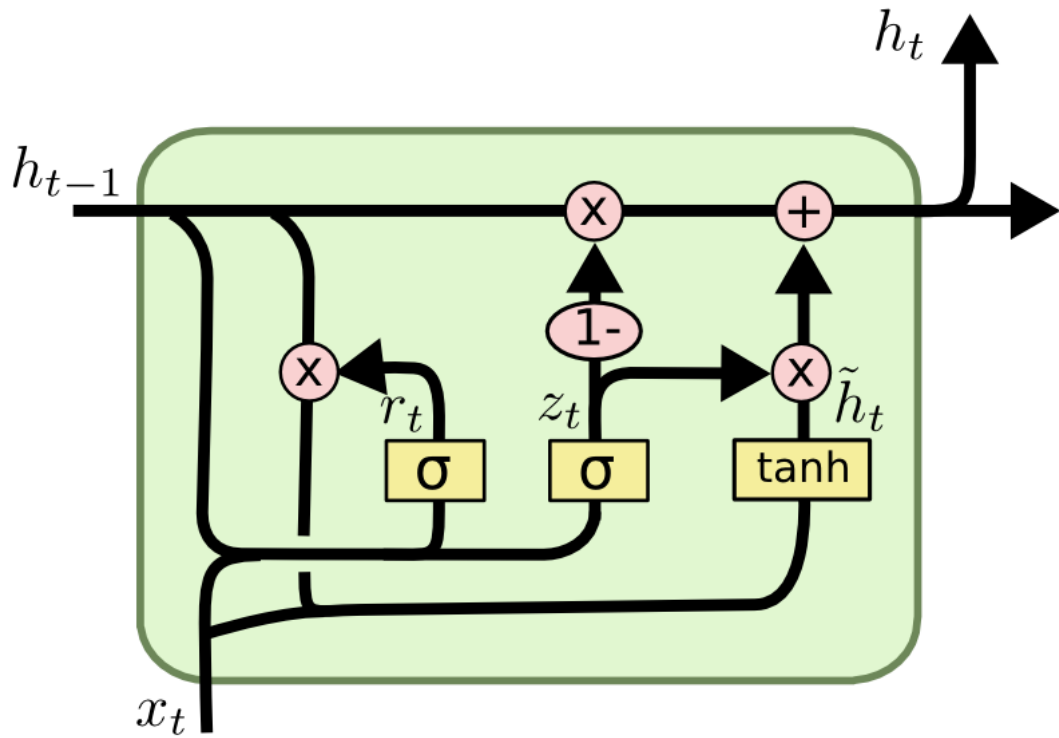


Figure 3.3: Block diagram of the Gated Recurrent Unit that includes sigmoid and tanh activation functions.

CHAPTER 4

METHODOLOGY

4.1 Preprocessing

The data will pass through two preprocessing stages. The first consists of reducing the dimensionality of the input spectrum. The data reduction will be performed using principle component analysis. This reduction will decrease the computational burden while preserving the important features that are found in the spectrum. The second will transform the input intensities into reflectance. This is a crucial step since reflectance is a material property and is robust to brightness changes.

4.1.1 *Data Reduction Experiment*

Several techniques are used to reduce the data size of hyperspectral images like Principle Component Analysis , Kernel PCA, Non-Negative Matric Factorization [48]. In this paper, PCA is adopted. PCA is based on projecting the data feature vectors into N-dimensional space that covers the most variability in the data. Reducing the N value enhances the speed of the algorithm during classification with a less computational load on the memory. However, reducing N might affect the accuracy of the model since we might lose some important features in our feature vector. Therefore, a thorough check should be completed to understand the relationship between N and the resulting accuracy. Although the reduced data can result in reduced memory consumption as well as speeding up the inference speed of the deep neural model, it delays the preprocessing time. Our target is to maximize the inference speed while minimizing the preprocessing latency. For that, the following implementation for PCA in a real-time data acquisition scenario will be implemented. The PCA can be divided into two major steps, the principal component extraction, and the data projection. The PC extraction is the most time-consuming step, for that we learn

the principal components offline from the training images. Then with each incoming frame, the data is directly projected into the principal component space.

4.1.2 Reflectance Transform

The intensity values captured by the camera are dependent on the scene radiance. Thus, capturing the same object but at a different location with different scene radiance will result in a different intensity value. In case we used such intensity values for classification, our deep learning model will fail. To compensate for this issue, we will transform the spectrum from the intensity space into the reflectance space. The reflectance is a material identity that is independent from the scene radiance [49] and thus it will be a great feature for classification. This transformation will be done using the gray-World method that approximates the scene radiance by the spatial mean for each layer of the hyperspectral image [50]. Following this approximation, each layer will be divided by its scene radiance to get the reflectance spectrum as shown in equation 4.2. The scene radiance is estimated as

$$L(\lambda) = \text{mean}(I(u, \lambda)) \quad (4.1)$$

Where lambda is the wavelength, u is the spatial location, and I is the intensity value. The transformation is illustrated in the following equation:

$$S(u, \lambda) = \frac{I(u, \lambda)}{L(\lambda)} \quad (4.2)$$

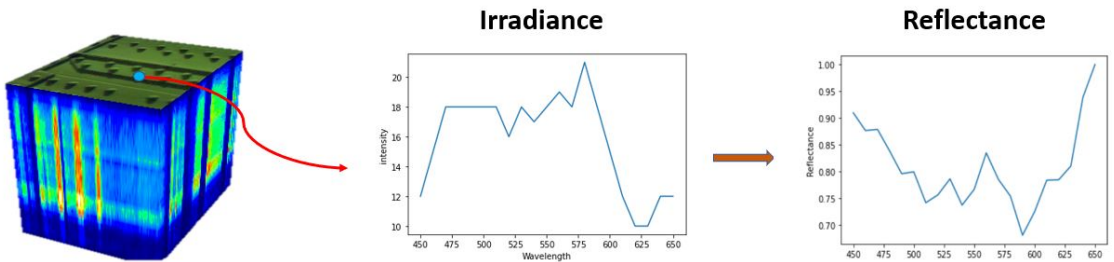


Figure 4.1: Pixel transformation from irradiance into reflectance

4.2 DNN

A spatial-spectral model will be built by adding the optimal number of spatial pixels. The model is a deep neural network model that consists of several hidden

dense layers. The activation function that is used in each hidden layer is the Relu function. Thus, the input for each node in the hidden layers is a weighted linear combination of all the previous node values in addition to a bias term. Then the Relu function will output zero if the combination is negative otherwise it outputs the value as it is.

The input for this model is the main pixel that needs to be classified with two of its neighboring pixels, specifically the left and the right one. This input looks as follows:

$$\vec{x} = [left\ pixel\ middle\ pixel\ right\ pixel] \quad (4.3)$$

As an example of the spatial information, imagine that the right and left pixels belong to class one, then there is a high probability for the middle pixel to belong to class one as well. Since each pixel will have N spectral dimension, the total number of neurons in the input layer will be 3xN. The output layer will consist of number of neurons equal to the number of output classes that it is dependent on the dataset used. In all datasets the output layer will include a softmax activation function that calculates the probability for each output class. These probability outputs are then fed with the ground truth probabilities to the cross entropy activation function that is shown in equation 4.5. To optimize this model, the stochastic gradient descent algorithm was utilized.

$$y_{predicted} = softmax\left(\sum_{i=1}^N a_i w_i + b_0\right) \quad (4.4)$$

$$Loss = \sum_{i=1}^{classes} y_i \cdot \log(y_{predicted}) \quad (4.5)$$

To ensure that the model did not overfit to the noise, dropout and regularization were utilized. The dropout was performed after the last hidden layer by dropping 20 percent of the neurons during the training phase. Additionally, ridge regularization was performed on three of the hidden layers. The lambda value was set to 0.01 in the three regularized layers. The developed method requires three consecutive pixels to do the prediction. An additional experiment that will be conducted is to vary the number of pixels needed under the constraint of belonging to the same horizontal line. Each of these inputs will be tested based on the accuracy and speed. Adding more pixels to the model will decrease the inference speed and increase the pre-processing delay, however, it might affect the accuracy of the model. In case there was a noticeable enhancement in the accuracy with minor decrease in speed,

the new number of pixels will be adopted as the recommended input.

4.3 RNN

The concept of the optimized input of three consecutive pixels was also applied using RNN as well. It is appealing to use RNN sequential model given that the input is sequential in term of time. Additionally, the middle pixel class is dependent on the class of the adjacent pixels, this concept can be easily learned by an RNN given that it includes hidden states that act as memory.

$$\vec{x} = \begin{bmatrix} x_1 & \cdots & x_k \end{bmatrix} \quad (4.6)$$

$$\vec{z} = \text{sigmoid}(W_z \cdot [h_{t-1}, x_t]) \quad (4.7)$$

$$r_t = \text{sigmoid}(W_r \cdot [h_{t-1}, x_t]) \quad (4.8)$$

$$\tilde{h}_t = \text{tanh}(W \cdot [r_t * h_{t-1}, x_t]) \quad (4.9)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (4.10)$$

The designed network is formed by three GRU units where the input for each GRU is the reflectance spectrum of one of the three pixels. Two layers of these triplet GRUs will be stacked on top of each other as seen in figure 4.2. The output of the bottom GRUs will be connected to the inputs of their respective top GRUs. Adding the layer will help the model learn more complex features. Concerning the order of the three pixels we will have two different implementations. The first one is connecting the three pixels in the same order as the GRUs are connected whereas the second implementation is connecting the last pixel to the second GRU and the last GRU to the next pixel. The intuition behind the latter implementation is to put the main pixel to be predicted, in our case the middle pixel, at the last GRU to be directly connected to the output. Otherwise, it will be transferred to the output through the hidden layer. Thus, by doing so as if we are giving a higher weight the pixel to be predicted. The downside of this implementation is that we no longer can predict the pixel as it comes, in other words, as the first pixel is received by the sensor it is forwarded through the first GRU but after that the algorithm waits until the two pixels are received and then forwards them. The implementation with the best accuracy to speed ratio will be chosen. The hidden states for all the GRU units are initialized as zero vectors. As shown in equation 4.6 , the pixel spectrum

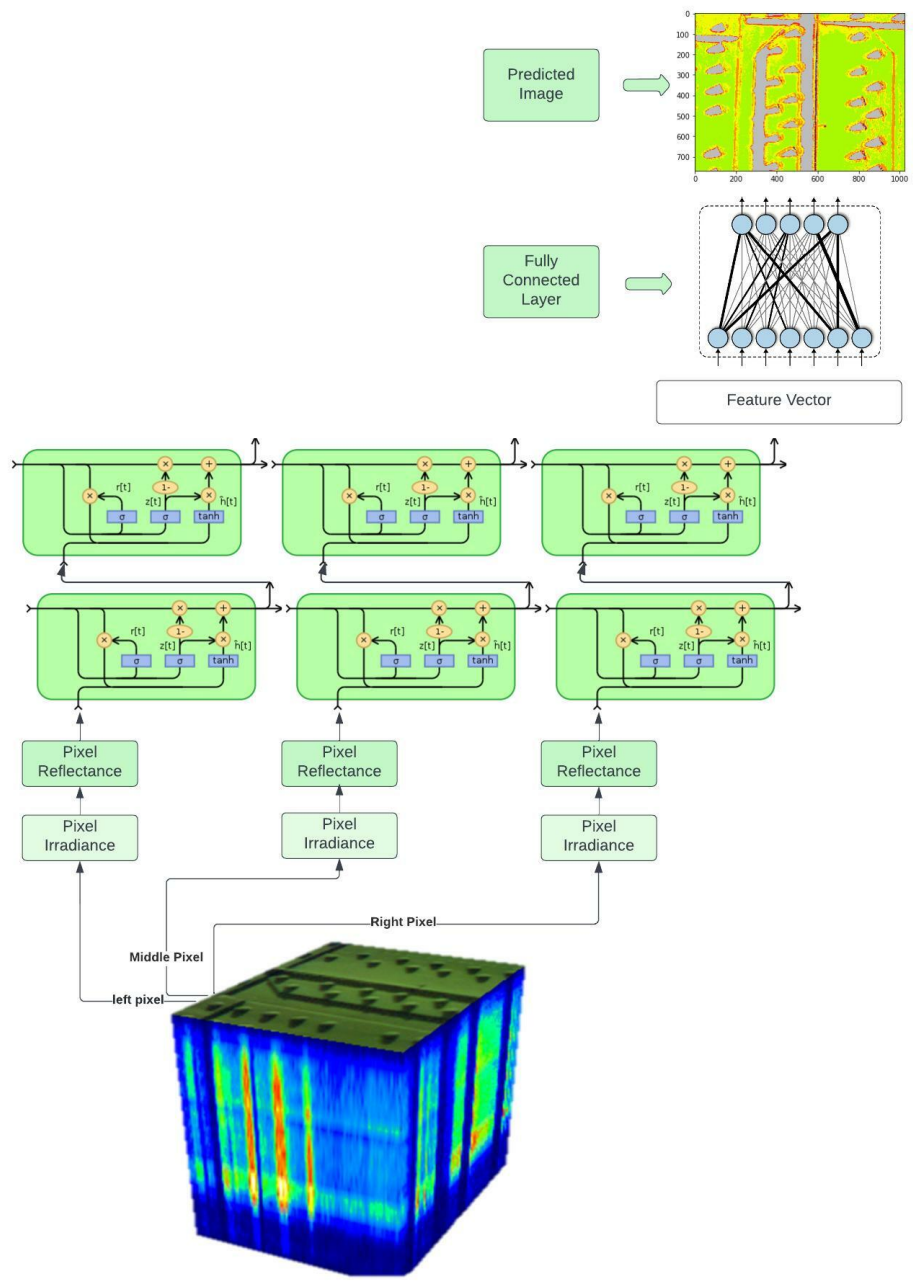


Figure 4.2: Recurrent Neural Network with GRU units.

is denoted by x vector where k refers to the number of bands. The reset gate in equation 4.8 and the update gate in equation 4.7 are denoted by r and z respectively, W_z , W_r and W stands for the weight's matrices. In terms of output, the activation of the last GRU is fed to a fully connected layer that consists of one hidden layer consisting of 100 units and output layer with number of units equivalent to the number of classes. The output layer is feed into a SoftMax activation function that calculates the probability of each class. A cross entropy loss function takes these probabilities to calculate the loss. The reason for choosing GRU units over the traditional RNN unit is to solve the vanishing gradient issue that might occur while using traditional RNN [51].

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Agriculture Case Study

5.1.1 *Dataset*

Three agricultural datasets are used. The first Dataset is the well-known Pavia University that was collected using the AVIRS sensor [52]. It includes a 610x610 sized hyperspectral image with 102 spectral bands. The spectral bands' wavelength ranges from 430 nm to 860 nm. There are 9 outputs for this dataset that covers various urban land classes. The second Dataset is called Indian Pines that consists of 224 spectral bands and 16 different classes of vegetation. The third is the Salinas dataset that includes 512x217 sized hyperspectral images with 224 spectral bands covering 16 different classes. The data in each dataset is split into training (80%) and testing samples (20%). The time taken by the sensor to acquire this image is 6.5 seconds. This acquisition time is important to determine whether the classification is real-time and thus our goal is to achieve preprocessing and classification for the whole image without exceeding the time-span of the acquisition .

5.1.2 *Model Accuracy*

For the Pavia dataset including the spatial information by adding the left and right pixel as an input has achieved a model accuracy of 92%. What is worth mentioning is that the model consisted of only 5705 parameters, thus we were able to incorporate the spatial information while keeping our model lightweight. This is a major advantage of our method compared to previous literature where incorporating the spatial information led to relatively large models, for instance, the hybrid 3D-2D CNN [52], included the spatial information by extracting patches of 25x25 pixels,

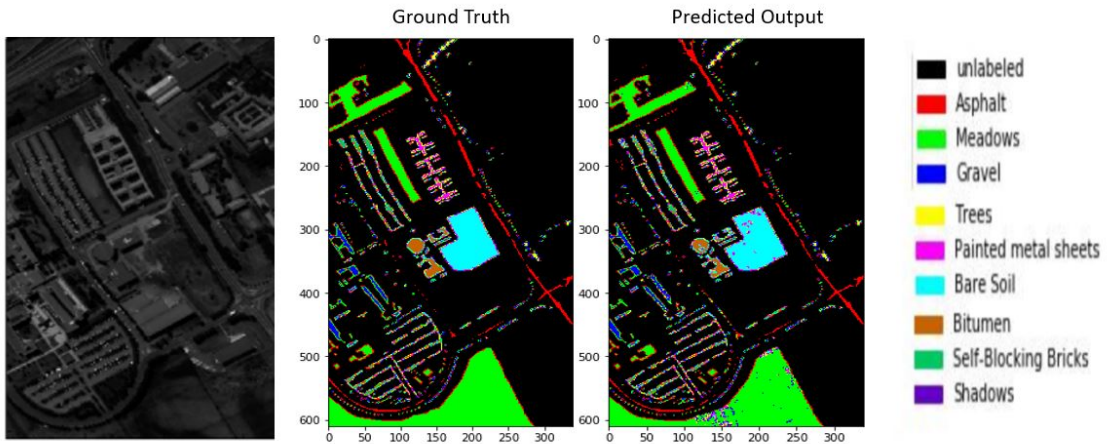


Figure 5.1: Pavia dataset classification results

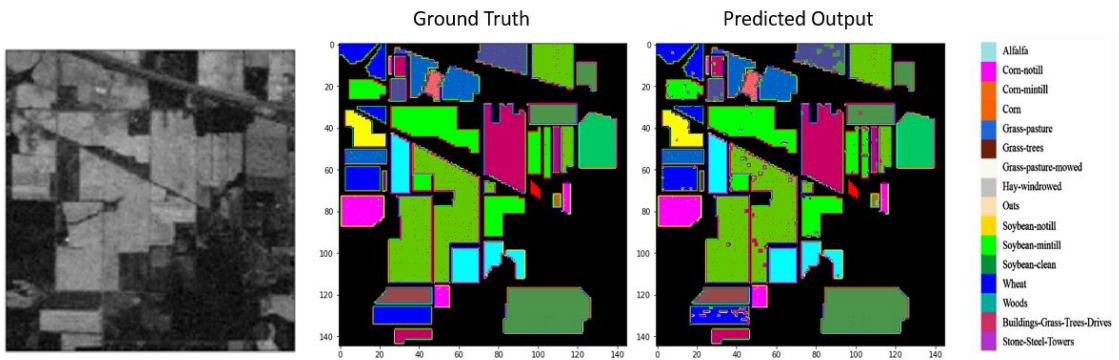


Figure 5.2: Indiana dataset classification results

and this led to a model consisting of 5 million parameters. Additionally, the stacked spectral-spatial vector input method [48], consisted of 28629 parameters. For the sake of enhancing our model’s ability to extract general spectral-spatial features, we trained the spectral-spatial model on the duplicated dataset. This enhanced our model accuracy by reaching a value of 96%. The results are shown in figure 5.1. Furthermore, our model achieved high accuracy in the other two datasets. As the model achieved an accuracy of 96% and 97% with the indian pines and salinas datasets respectively, the prediction results can be seen in figure 5.2 and 5.3 .

5.1.3 Preprocessing and Inference Speed

For Real-time application, the sum of the preprocessing time and the inference time should be less than the time taken to acquire a hyperspectral frame. The preprocessing in spatial-spectral models consists of PCA and spatial-spectral input array

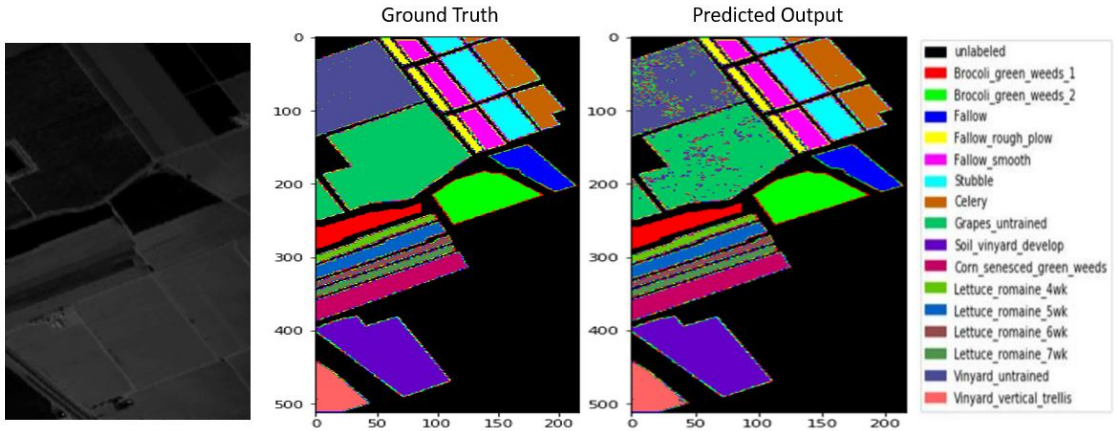


Figure 5.3: Salinas dataset classification results

formation. In our case, we were able to minimize the combined latency to 2.6 seconds. Accordingly, we achieved real-time classification for Hyperspectral images. In the following subsections, our method was benchmarked with other spatial-spectral models in terms of latency and throughput. For fair benchmark, all the experiments are conducted on the K80 Tesla GPU provided by google colab. The time taken to apply the standard PCA approach, on the Pavia University frame, was tested to be 2.6 seconds. This time was split into 2.5 seconds for extracting the principal components and 0.1 seconds for projecting the pixels to the reduced dimension. The Realtime approach learns the principal components offline and only applies the projection as new frames are acquired. Thus, the PCA latency decreased from 2.6 seconds to only 0.1 seconds per frame. In the spectral-spatial model, there is a major step in preprocessing that consumes time. The step of filling the rows of the input array by the three neighboring pixels. Since our method consists of taking an optimized spatial input, this step is faster than other methods in the literature. In our case each row in the input matrix is filled with three pixels only, compared to at least 9 pixels in the case of stacked vector and 25x25 patch based layers in Hybrid 3D-2D. As a result of this reduced input, our method took only 2.5 seconds, whereas the vector method took about 4.1 seconds, and the patch method took at least 5.9 seconds. The throughput was tested using a batch size of 1000. Our lightweight architecture led to a fast inference rate. We have achieved a throughput of 1,200,000 pixels per second. This fast throughput outperformed both stacked vector and patch-based input having the corresponding throughput of 239,128 and 1500 pixels per second respectively. A detailed benchmark is found in table .

In a nutshell, our approach achieved comparable accuracy with other spectral-

Table 5.1: Benchmark for spatial-spectral methods

	Our Method	Stacked Vector	Patch
Input (Pixels)	3	At least 9	25x25
Accuracy (%)	96	98.7	99.97
Preprocessing Latency (sec)	2.7	4.2	>6
Inference (Pixel/sec)	1,200,000	239128	1500
Parameters	5705	28629	5,122,176

spatial models while decreasing the preprocessing latency by one fold and increasing the inference speed by 5 folds. In addition to the real time speed achieved by our model, the small number of parameters makes it a perfect fit for onboard processors.

5.2 Paint condition Case Study

5.2.1 *Datasets*

The Dataset consists of 768 by 1024 images for the Sydney bridge. The distance from the camera to the surface differs from image to another. Accordingly, the images are either considered as short-range images taken from a distance less than 5 meters, or long-range images with a distance ranging between 10 and 40 meters. The wavelength ranges from 450 to 650 nm with a resolution of 10 nm. Consequently, the number of bands will be 21 distributed along the previously mentioned spectral range. The labels of the short-range images include paint condition levels 1, 3 and 4 in addition to the background label. Whereas the long-range images include the four levels of paint condition in addition to a shadow and a sky label. For both the long range and short-range datasets, the labeled data was split into training and testing with a split ratios of 80 % and 20 % respectively.

5.2.2 *Classification Results*

For short-range images, four images are used to test the performance of the models. With each pixel in the images as a single data sample, the test portion of these images consisted of 11016 samples of level one, 1619 samples of level 3, 46049 samples of

Table 5.2: DNN model performance on short range images

	Precision	Recall	f1-score	support
Level 1	1	1	1	11016
Level 3	0.90	0.99	0.94	1619
Level 4	1	1	1	46048
Background	1	1	1	1235
Accuracy			0.97	59918
Weighted Average	0.97	1	1	59918

Table 5.3: RNN model performance on short range images

	Precision	Recall	f1-score	support
Level 1	1	1	1	11016
Level 3	0.96	0.95	0.96	1619
Level 4	1	1	1	46048
Background	1	1	1	1235
Accuracy			0.99	59918
Weighted Average	1	1	1	59918

level 4 and 1235 samples of the background class. For the DNN model, the lowest accuracy is for the prediction of level 3 at 90% whereas all the other classes are predicted with a 100 percent accuracy. For this model, the average accuracy over all the classes was 97 percent. On the other hand, the RNN with GRU units significantly outperform the DNN in Level 3 with an accuracy of 96% for this level and the same performance as the DNN for the other levels. These results are depicted in tables 5.2 and 5.3. In general, the RNN model outperforms the DNN model with an average accuracy of 99%. The predictions of the images for the two models are shown in figure 5.5 and the confusion matrices for the predictions of the two models are shown in figure 5.4. Compared to literature that used the same dataset, our models achieve an unprecedented accuracy over the work of [31] who used Support Vector Machines to reach an overall accuracy of 75.16% on short range images.

As for Long-range images, four images are used for testing the performance of the models. The test portion of these images consisted of 12032 samples of level one, 3564 samples of level 2, 5559 samples of level 3, 2275 samples of level 4, 1024 samples of the background class and 1101 samples of the sky class. The DNN model outperforms the RNN model in level four, the RNN model outperforms the DNN model in levels one and three, and both models have the same performance in predicting level 2, sky and the shadow. The confusion matrices for the predictions of the two models can be shown in figure 5.6. In general, the RNN model, with an average accuracy of 97%, outperforms the DNN model with an average accuracy

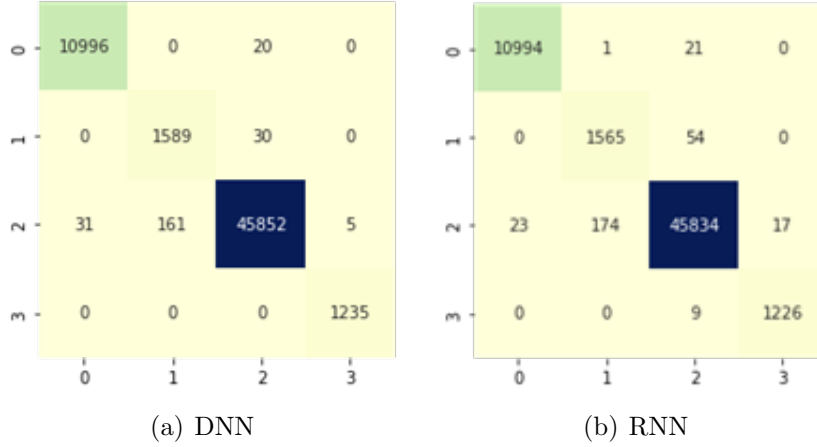


Figure 5.4: confusion matrices for short range models

Table 5.4: DNN model performance on Long range images

	Precision	Recall	f1-score	support
Level 1	0.95	0.98	0.97	12032
Level 2	0.93	0.93	0.93	3564
Level 3	0.94	0.90	0.92	5559
Level 4	0.98	0.99	0.98	2275
Background	1	1	1	1024
Sky	1	1	1	1101
Accuracy			0.96	25555
Weighted Average	0.96	0.96	0.96	25555

of 96%. Compared to previous work, both model accuracies outperform the SVM method in [31] that achieved an accuracy of 85.54 % for the long range images.

The speed is one of the most important attributes to our model as our goal is to perform paint analysis in real time. In that sense, the DNN model, with a speed of 83665 pixels per second, outperforms the RNN model with a speed of only 25,000 pixels per second which is approximately 3.4 times faster. Given that the accuracy of the RNN model outperforms that of the DNN model whereas the RNN model proves to be faster in terms of inference speed, the RNN model is recommended for for offline applications where the accuracy is the most critical metric whereas the DNN model is recommended for online scenarios.

5.2.3 Input Experiments

As mentioned in section ??, several experiments are conducted to test the possibility of enhancing the optimal input. An ideal enhancement is where the accuracy of the

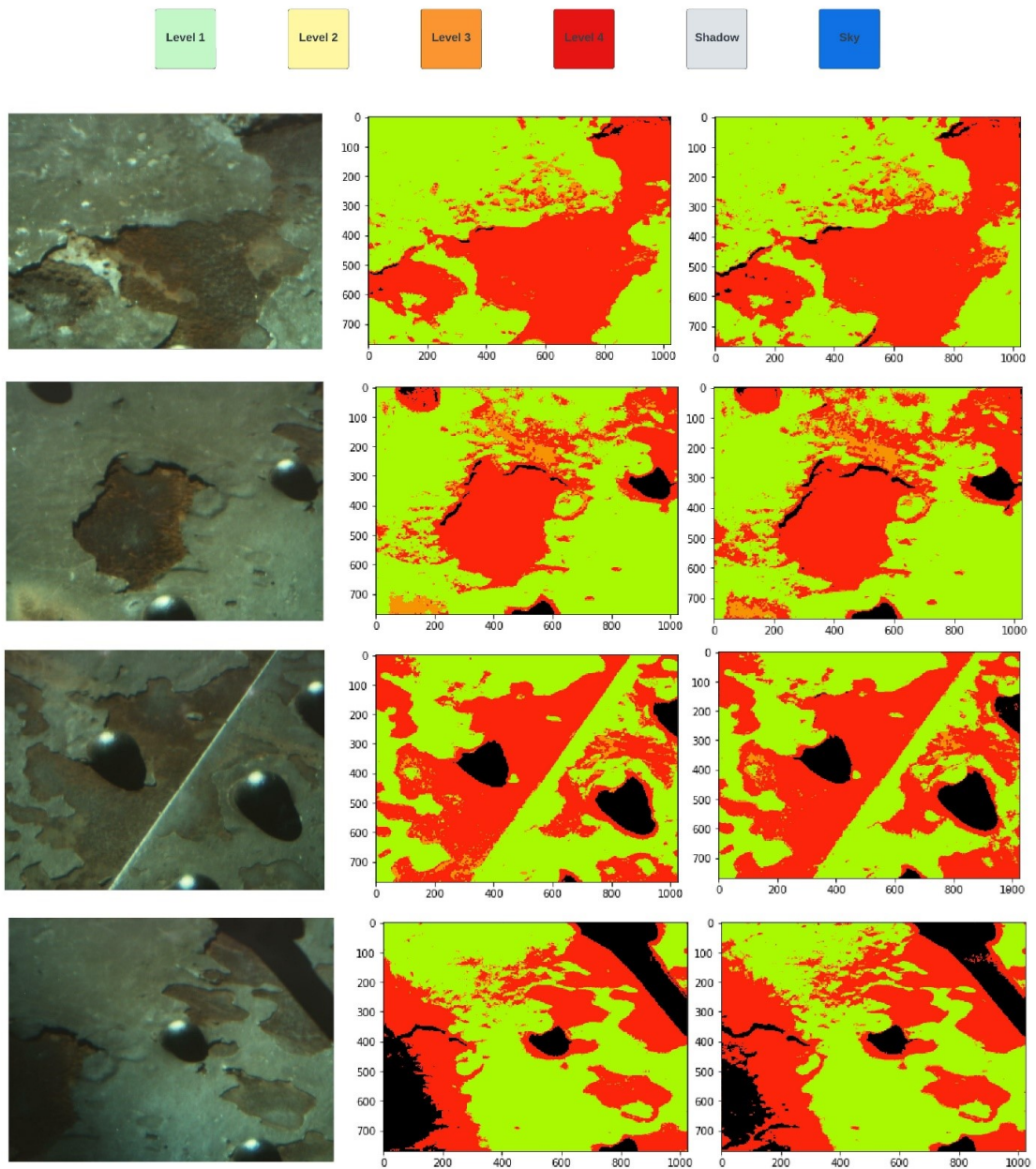


Figure 5.5: Short Range classification results

Table 5.5: RNN model performance on Long range images

	Precision	Recall	f1-score	support
Level 1	0.98	0.98	0.97	12032
Level 2	0.93	0.95	0.93	3564
Level 3	0.97	0.90	0.93	5559
Level 4	0.97	0.99	0.98	2275
Background	1	1	1	1024
Sky	1	1	1	1101
Accuracy			0.97	25555
Weighted Average	0.97	0.97	0.97	25555

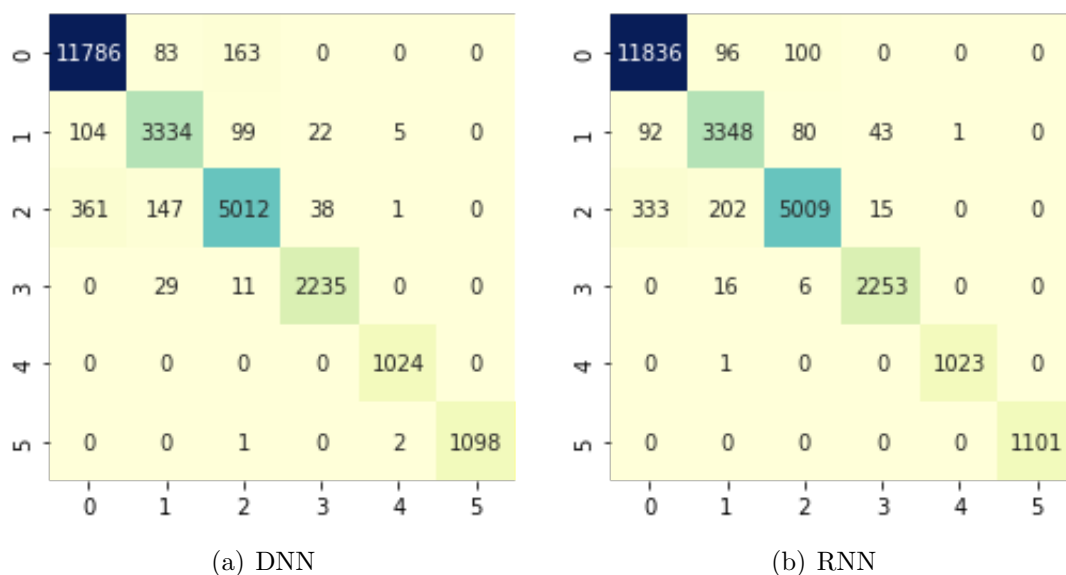


Figure 5.6: confusion matrices for Long range models

model and the inference speed are both increased while the situation where one of the two metrics increases and the other decreases to a lesser extent is considered an enhancement nonetheless.

The experiment of flipping the last two pixels is conducted on the short-range dataset only. Two models are trained such that the first model has the middle pixel connected to the middle GRU and the right pixel to the right GRU, whereas the second model has the middle pixel connected to the right GRU and the right pixel to the middle GRU, thus having the last two input pixels interchanged in the network. The results indicates that the two models perform the same in terms of accuracy with 99 %.The reason for the same model accuracy can be explained by the fact that most training instances of three neighboring pixels in the data belong

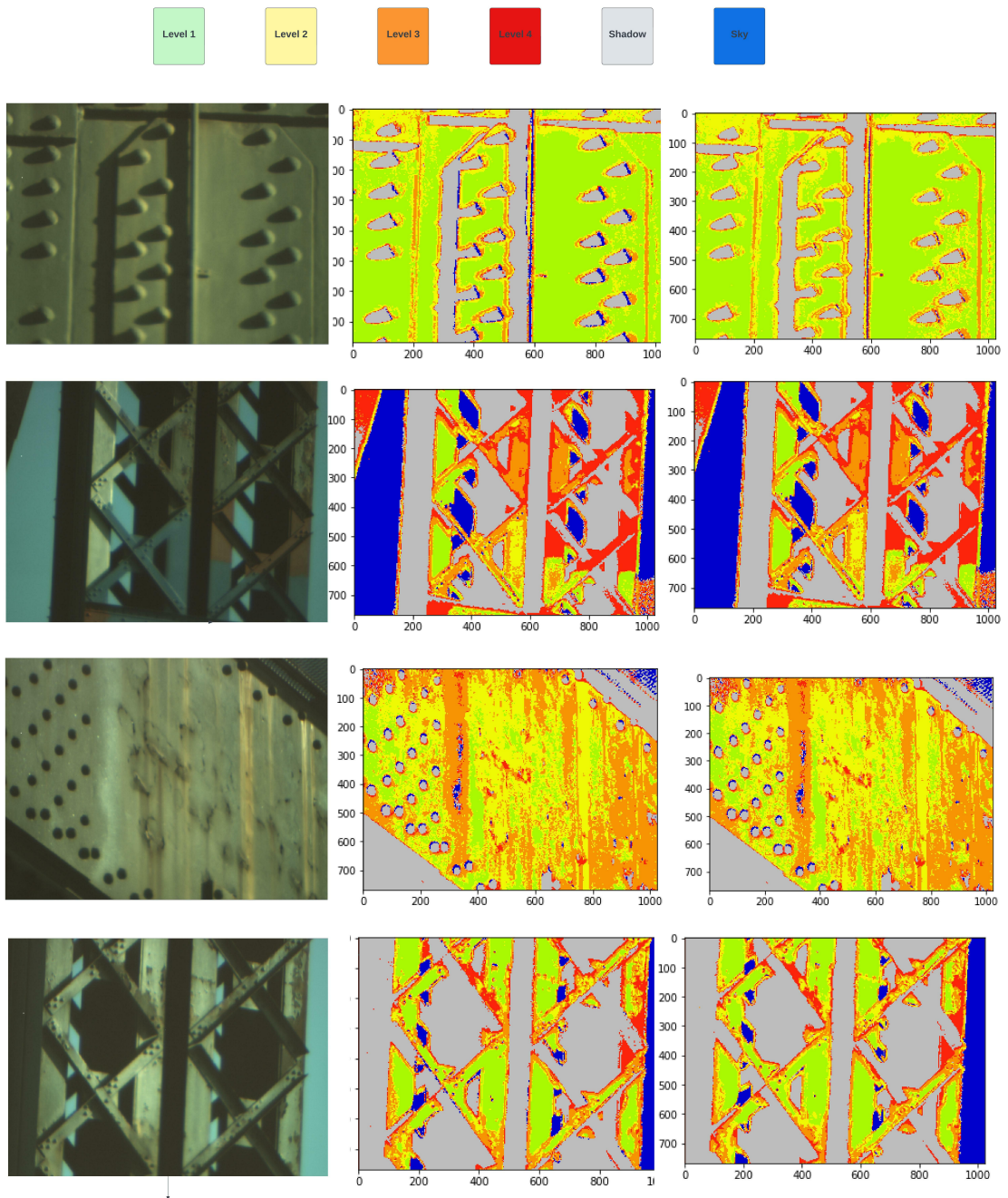


Figure 5.7: Long Range classification results

Table 5.6: score for the three pixel input versus the five pixel input

	Accuracy	Preprocessing Time (sec)	Inference Speed (Pixel/sec)	Inference Time (sec)	Score
Three Pixels	98	1.37	83665	0.01	-
Five Pixels	x	x	x	x	x

to the same class and thus the switching of the middle and right pixels results in no change between how both model schemes interpret the input data. Therefore, the two models are subjected to mostly the same input data during training which leads to the models performing the same in terms of accuracy. However, it is important to mention that the non-flipped or consecutive input model has an advantage over the flipped one in terms of speed as it can do the prediction at the same time when the input is received from the sensor without any delay which is the reason we choose the consecutive input scheme as the favored and best performing model.

For the sake of enhancing the accuracy of our model, various pixel inputs are passed under the constraint that the pixels are consecutive in a horizontal manner with each input pixel number receiving a score to compare. The score is a function of accuracy, preprocessing speed, and inference speed. Due to more interest in onboard applications, a relatively higher weight is given for speed over the accuracy. The score metric is as follows:

$$score = accuracy + \frac{w}{t_p + t_i} \quad (5.1)$$

Where t_p and t_i stands for the preprocessing time and inference time respectively. The w stands for the weight and will be set to 2. The inputs tested are three and five consecutive pixels.

The three pixel input works properly with the hardware of the system and gives a fast inference speed of 83665 pixel/sec with a short delay of 1.37 seconds during preprocessing. As for the five pixel method, the system it was run on was unable to process it despite a sufficient 12 GB of RAM which indicates a major disadvantage of this method in terms of computational load which makes it not applicable for online applications. These results can be seen in table 5.6. Thus the results of the two previously mentioned experiments indicate that the optimal input is indeed that consisting of three consecutive pixels.

There are three main types of hyperspectral cameras: the single-shot sensor, the whiskbroom sensor, and the push broom sensor. The models based on the horizontal

three pixel scheme can be proven to be one of the fastest spatial-spectral models in all the three sensor scenarios. Single-shot cameras save the 3D hyperspectral cube in one-shot so to achieve real time classification, parallel prediction is a must. The prediction was executed using a batch size of 1000 pixels in (cite previous work) and proved that the three pixel method is faster than other spatial-spectral approaches while using a one-shot sensor. Whisk-broom sensor scans the pixels line by line which has a great consequence on the prediction rate of the spatial-spectral methods. Since the spsp that are found in the literature uses a patch of $n \times n$ [48], [52], so they need to be in idle state until the sensor scans at least n lines before every prediction. On the other hand, the three pixel method can predict line by line by applying a batch prediction that covers the whole line. The third type is the push-broom sensor that scans the scene pixel by pixel. Like the case of whiskbroom sensor, the spsp method must wait for at least n lines before doing each prediction. On the other hand, the three pixel method must wait for only 3 pixels before each prediction.

The feature that made the optimal input superior in different sensor scenarios is taking the adjacent pixels in horizontal manner. Any other configuration, like vertical or diagonal would have caused a significant delay during prediction since the model should then wait for at least three lines before performing its predictions.

CHAPTER 6

CONCLUSION

Achieving real-time hyperspectral image classification is of great benefit given that numerous applications require immediate action such as surgical, structural monitoring and agricultural applications [53]. The complexity in real-time onboard classification arises because hyperspectral images have a large volume alongside the limited computational power for onboard computers [12].

Several approaches for real time hyperspectral classification have been done based on the spectral information as an input. Adding the spatial information enhances the model performance but at the same time adds a computational burden. In our work, we found the optimal spatial-spectral input that enhances the model performance with minimal additional computational cost. The optimal input consisted of three consecutive pixels. The role of the right and left pixel is to add spatial information that enhances the accuracy of the classification models.

Using this optimal input we proposed two novel deep learning models for the classification of hyperspectral images. A DNN architecture consisting of few hidden layers and an output layer with number of neurons equal to the classes to be classified. The second model is an RNN that consisted of two layers of GRUs. The choice of GRU was to solve the issue of vanishing gradients that is found in standard RNN architectures. The output of the two GRU layers was connected to a fully connected network to perform the required classification.

Several experiments were conducted to prove that this three pixel input is indeed the optimal one. One of the experiments was to try five consecutive pixels instead of three and pass them to the models. Such input drastically increased the computational power and lead to overflow of the Random Access Memory (RAM). Additionally, choosing the input pixels in horizontal manner made it efficient in all types of hyperspectral cameras. Since the horizontal input can be classified directly

as the pixel is received from the sensor in contrary to other methods in the literature that waits several lines before performing the classification.

Our RNN and DNN models outperformed previous approaches in terms of accuracy and speed. In the Pavia case study, this novel approach achieved an accuracy of 96% with real time preprocessing and classification speed. Furthermore, In the Sydney Bridge case study, the RNN architecture having an accuracy of 99 % outperformed the DNN that got an accuracy of 97 %. On the other hand the inference speed of the DNN was about four times faster than the RNN. For this reason, we recommend the DNN architecture in cases where the real time inference is required. On the other hand, we recommend our RNN in scenarios where the accuracy of the prediction is very critical. The fact that the optimal input gave high accuracy in both case studies proves that it is not biased and can easily generalize to new datasets.

In the near future, we will receive a Hyperspectral Camera. This camera will be used to capture soil samples from across Lebanon. The aim will be to implement our model on an onboard computer such as jetson boards and perform a real-time onboard prediction for soil contamination. Shifting from the current models into models that can classify soil contamination is an easy task. To do this shift all what is needed is to train our models on a new soil contamination dataset.

And in case the models will be used for regression, for instance predicting the contamination concentration in a soil sample, minor changes should be made to the current models. The changes will be conducted in the last layer by substituting the softmax activation function by a relu function. Additionally, the cost function will be changed from the cross entropy into the Mean Squared Error. The majority of the work will be on collecting the data and performing accurate labeling.

BIBLIOGRAPHY

- [1] X. Jia, Y. Cao, D. O'Connor, *et al.*, "Mapping soil pollution by using drone image recognition and machine learning at an arsenic-contaminated agricultural field," *Environmental Pollution*, vol. 270, p. 116 281, 2021.
- [2] F. Wang, J. Gao, and Y. Zha, "Hyperspectral sensing of heavy metals in soil and vegetation: Feasibility and challenges," *ISPRS journal of photogrammetry and remote sensing*, vol. 136, pp. 73–84, 2018.
- [3] J. Borjac, M. El Joumaa, R. Kawach, L. Youssef, and D. A. Blake, "Heavy metals and organic compounds contamination in leachates collected from deir kanoun ras el ain dump and its adjacent canal in south lebanon," *Heliyon*, vol. 5, no. 8, e02212, 2019.
- [4] C. M. Haydar, N. Nehme, S. Awad, *et al.*, "Assessing contamination level of heavy metals in the lake of qaraaoun. lebanon," *Physics Procedia*, vol. 55, pp. 285–290, 2014.
- [5] J. El-Nakat, "Measurement of levels of heavy metal contamination in vegetables grown and sold in selected areas in lebanon," *Jordan Journal of Chemistry (JJC)*, vol. 4, no. 3, pp. 303–315, 2009.
- [6] T. Darwish, I. Jomaa, M. Awad, and R. Boumetri, "Preliminary contamination hazard assessment of land resources in central bekaa plain of lebanon," *Lebanese Science Journal*, vol. 9, no. 2, pp. 3–15, 2008.
- [7] Z. Liu, Y. Lu, Y. Peng, L. Zhao, G. Wang, and Y. Hu, "Estimation of soil heavy metal content using hyperspectral data," *Remote Sensing*, vol. 11, no. 12, p. 1464, 2019.
- [8] L. A. Martins, G. A. Sborz, F. Viel, and C. A. Zeferino, "An svm-based hardware accelerator for onboard classification of hyperspectral images," in *Proceedings of the 32nd Symposium on Integrated Circuits and Systems Design*, 2019, pp. 1–6.

- [9] K. Tan, H. Wang, L. Chen, Q. Du, P. Du, and C. Pan, “Estimation of the spatial distribution of heavy metal in agricultural soils using airborne hyperspectral imaging and random forest,” *Journal of hazardous materials*, vol. 382, p. 120 987, 2020.
- [10] A. Gholizadeh, M. Saberioon, E. Ben-Dor, and L. Borvka, “Monitoring of selected soil contaminants using proximal and remote sensing techniques: Background, state-of-the-art and future perspectives,” *Critical Reviews in Environmental Science and Technology*, vol. 48, no. 3, pp. 243–278, 2018.
- [11] Q. Du and R. Nekovei, “Fast real-time onboard processing of hyperspectral imagery for detection and classification,” *Journal of Real-Time Image Processing*, vol. 4, no. 3, pp. 273–286, 2009.
- [12] Y. Dua, V. Kumar, and R. S. Singh, “Comprehensive review of hyperspectral image compression algorithms,” *Optical Engineering*, vol. 59, no. 9, p. 090 902, 2020.
- [13] H. Li, H. Zheng, C. Han, H. Wang, and M. Miao, “Onboard spectral and spatial cloud detection for hyperspectral remote sensing images,” *Remote Sensing*, vol. 10, no. 1, p. 152, 2018.
- [14] M. B. Stuart, A. J. McGonigle, and J. R. Willmott, “Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems,” *Sensors*, vol. 19, no. 14, p. 3071, 2019.
- [15] M. Halicek, H. Fabelo, S. Ortega, G. M. Callico, and B. Fei, “In-vivo and ex-vivo tissue analysis through hyperspectral imaging techniques: Revealing the invisible features of cancer,” *Cancers*, vol. 11, no. 6, p. 756, 2019.
- [16] R. Arablouei, E. Goan, S. Gensemer, and B. Kusy, “Fast and robust pushbroom hyperspectral imaging via dmd-based scanning,” in *Novel Optical Systems Design and Optimization XIX*, SPIE, vol. 9948, 2016, pp. 7–17.
- [17] D. Turner, A. Lucieer, M. McCabe, S. Parkes, and I. Clarke, “Pushbroom hyperspectral imaging from an unmanned aircraft system (uas)–geometric processing workflow and accuracy assessment,” 2017.
- [18] X. Ceamanos and S. Valero, “Processing hyperspectral images,” in *Optical Remote Sensing of Land Surface*, Elsevier, 2016, pp. 163–200.

- [19] R. F. Kokaly and R. N. Clark, “Spectroscopic determination of leaf biochemistry using band-depth analysis of absorption features and stepwise multiple linear regression,” *Remote sensing of environment*, vol. 67, no. 3, pp. 267–287, 1999.
- [20] J. Li, A. Agathos, D. Zaharie, J. M. Bioucas-Dias, A. Plaza, and X. Li, “Minimum volume simplex analysis: A fast algorithm for linear hyperspectral unmixing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 9, pp. 5067–5082, 2015.
- [21] T. Shi, Y. Chen, Y. Liu, and G. Wu, “Visible and near-infrared reflectance spectroscopy—an alternative for monitoring soil contamination by heavy metals,” *Journal of hazardous materials*, vol. 265, pp. 166–176, 2014.
- [22] K. Yu, S. Fang, and Y. Zhao, “Heavy metal hg stress detection in tobacco plant using hyperspectral sensing and data-driven machine learning methods,” *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 245, p. 118 917, 2021.
- [23] W. W. Greenwood, J. P. Lynch, and D. Zekkos, “Applications of uavs in civil infrastructure,” *Journal of infrastructure systems*, vol. 25, no. 2, p. 04 019 002, 2019.
- [24] P. M. Wyder, Y.-S. Chen, A. J. Lasrado, *et al.*, “Autonomous drone hunter operating by deep learning and all-onboard computations in gps-denied environments,” *PloS one*, vol. 14, no. 11, e0225092, 2019.
- [25] S. Mohan, O. Shoghli, A. Burde, and H. Tabkhi, “Low-power drone-mountable real-time artificial intelligence framework for road asset classification,” *Transportation Research Record*, vol. 2675, no. 1, pp. 39–48, 2021.
- [26] D. Waleed, S. Mukhopadhyay, U. Tariq, and A. H. El-Hag, “Drone-based ceramic insulators condition monitoring,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.
- [27] A. G. Howard, M. Zhu, B. Chen, *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.
- [28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

- [29] J. M. Haut, S. Bernabé, M. E. Paoletti, R. Fernandez-Beltran, A. Plaza, and J. Plaza, “Low–high-power consumption architectures for deep-learning models applied to hyperspectral image classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 5, pp. 776–780, 2018.
- [30] S. Ullah and D.-H. Kim, “Benchmarking jetson platform for 3d point-cloud and hyper-spectral image classification,” in *2020 IEEE International conference on big data and smart computing (BigComp)*, IEEE, 2020, pp. 477–482.
- [31] C. P. Huynh, S. Mustapha, P. Runcie, and F. Porikli, “Multi-class support vector machines for paint condition assessment on the sydney harbour bridge using hyperspectral imaging,” *Struct. Monit. Maint*, vol. 2, no. 3, pp. 181–197, 2015.
- [32] M. R. Mousa, S. R. Mousa, M. Hassan, P. Carlson, and I. A. Elnaml, “Predicting the retroreflectivity degradation of waterborne paint pavement markings using advanced machine learning techniques,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2675, no. 9, pp. 483–494, Apr. 2021. DOI: [10.1177/03611981211002844](https://doi.org/10.1177/03611981211002844).
- [33] H. Liu and Y. Zhang, “Image-driven structural steel damage condition assessment method using deep learning algorithm,” *Measurement*, vol. 133, pp. 168–181, 2019, ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2018.09.081>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0263224118309163>.
- [34] A. W. Momber, T. W. Nattkemper, D. Langenkämper, *et al.*, “A data-based model for condition monitoring and maintenance planning for protective coating systems for wind tower structures,” *Renewable Energy*, vol. 186, pp. 957–973, Mar. 2022. DOI: [10.1016/j.renene.2022.01.022](https://doi.org/10.1016/j.renene.2022.01.022).
- [35] Y. Ozaki, C. Huck, S. Tsuchikawa, and S. B. Engelsen, Eds., *Near-Infrared Spectroscopy*. Springer Singapore, 2021. DOI: [10.1007/978-981-15-8648-4](https://doi.org/10.1007/978-981-15-8648-4).
- [36] B. Lavine, J. Almirall, C. Muehlethaler, C. Neumann, and J. Workman, “Criteria for comparing infrared spectra: A review of the forensic and analytical chemistry literature,” *Forensic Chemistry*, vol. 18, p. 100 224, May 2020. DOI: [10.1016/j.forc.2020.100224](https://doi.org/10.1016/j.forc.2020.100224).
- [37] D. Wu and D.-W. Sun, “Advanced applications of hyperspectral imaging technology for food quality and safety analysis and assessment: A review part

- i: Fundamentals,” *Innovative Food Science; Emerging Technologies*, vol. 19, pp. 1–14, Jul. 2013. DOI: [10.1016/j.ifset.2013.04.014](https://doi.org/10.1016/j.ifset.2013.04.014).
- [38] L. He, J. Li, C. Liu, and S. Li, “Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018. DOI: [10.1109/tgrs.2017.2765364](https://doi.org/10.1109/tgrs.2017.2765364).
- [39] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, “Deep learning for hyperspectral image classification: An overview,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019. DOI: [10.1109/tgrs.2019.2907932](https://doi.org/10.1109/tgrs.2019.2907932).
- [40] S. Mei, X. Yuan, J. Ji, Y. Zhang, S. Wan, and Q. Du, “Hyperspectral image spatial super-resolution via 3d full convolutional neural network,” *Remote Sensing*, vol. 9, no. 11, p. 1139, Nov. 2017. DOI: [10.3390/rs9111139](https://doi.org/10.3390/rs9111139).
- [41] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016. DOI: [10.1109/tgrs.2016.2584107](https://doi.org/10.1109/tgrs.2016.2584107).
- [42] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” *arXiv preprint arXiv:1506.00019*, 2015.
- [43] K. Kawakami, “Supervised sequence labelling with recurrent neural networks,” Ph.D. dissertation, Technical University of Munich, 2008.
- [44] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.
- [45] Z. Yu and G. Liu, “Sliced recurrent neural networks,” *arXiv preprint arXiv:1807.02291*, 2018.
- [46] K. Cho, B. Van Merriënboer, C. Gulcehre, *et al.*, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [47] L. Mou, P. Ghamisi, and X. X. Zhu, “Deep recurrent neural networks for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

- [48] 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.
- [49] B. Hapke, *Theory of reflectance and emittance spectroscopy*. Cambridge university press, 2012.
- [50] G. Buchsbaum, “A spatial processor model for object colour perception,” *Journal of the Franklin institute*, vol. 310, no. 1, pp. 1–26, 1980.
- [51] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [52] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, “Hybridsn: Exploring 3-d-2-d cnn feature hierarchy for hyperspectral image classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277–281, 2019.
- [53] D. Madroñal, R. Lazcano, R. Salvador, *et al.*, “Svm-based real-time hyperspectral image classifier on a manycore architecture,” *Journal of Systems Architecture*, vol. 80, pp. 30–40, 2017.