DEEP CONVOLUTION NEURAL NETWORKS WITH RESNET ARCHITECTURE FOR SPECTRAL-SPATIAL CLASSIFICATION OF DRONE BORNE AND GROUND BASED HIGH RESOLUTION HYPERSPECTRAL IMAGERY

Abhinav Galodha[1]*, Rahul Vashisht[1], Nidamanuri R.R[1], Ramiya A.M[1]

[1]Department of Earth and Space Sciences, Indian Institute of Space Science and Technology (IIST),
Department of Space, DoS, ISRO, Trivandrum, India - (abhinavgalohda, rahulvashisht290)@gmail.com, (rao,ramiya)@iist,ac.in

Commission II, WG II/6

KEY WORDS: Unmanned Aerial Vehicle (UAV); Precision Agriculture; crop classification; Deep residual networks (ResNet); Hyperspectral image classification (HSI); Convolution Neural Networks (CNN);

ABSTRACT:

Drones have been of vital importance in the fields of surveillance, mapping, and infrastructure inspection. Drones have played a vital role in acquiring high-resolution images and with the present need for precision farming, drones have helped in crop classification and monitoring various crop patterns. With the recent advancement in computational power and development of robust algorithms to carry out deep feature learning and neural network, based learning such techniques have regained prominence in contemporary research areas such as classification of common 2-D and 3-D images, object detection, etc. In our research, we propose a deep convolutional neural network architecture (CNN) for the classification of aerial images captured by drones and high-resolution Terrestrial Hyperspectral (THS or HSI) which includes 6-layers and with weights optimized along with the input layer, the convolutional layer, the max-pooling layer, the fully connected layer, softmax probability classifier, and the output layer. We have acquired THS (using Cubert-GmbH data) and drone agricultural data of seasonal crops sowed during the months of March-June for the year 2017. Crop patterns include Cabbage, Eggplant, and Tomato with varying nitrogen concentrations in the region of Bangalore, Southern India. To study the influence and impact of CNN, the ResNets model has been applied. ResNets model and architecture are combined with a deep learning network followed by a recurrent neural learning network model (RCNN). The HSI input layer with corresponding ground truth data for the region is fed into the ResNets model with a spectral and spatial residual network for the 7*7*139 input Hyperspectral Imagery (HSI) volume. The network includes two spectral and two spatial residual blocks. An average pooling layer and a fully connected layer transform into a 5*5*24 spectral-spatial feature volume further to a single output feature vector. At present we use an RMSProp optimizer for error loss minimization which when applied to the drone data was able to achieve an overall accuracy of 97.16%. Similarly, for cabbage, eggplant and tomato acquired through the same method we achieved overall accuracy at 87.619%, 89.25%, and 80.566% respectively in comparison to ground truth labels. Drones and ground-based datasets equipped with good computational techniques have become promising tools for improving the quality and efficiency of precision agriculture today.

1. INTRODUCTION

Hyperspectral images help in determining and classifying every pixel corresponding to different land covers and landscapes by information gathered across band wavelengths in the electromagnetic spectrum. The increase in spectral and spatial resolution of hyperspectral images (HSI's) pertains to two major obstacles any remote sensing user faces. Also, when the number of training samples is relatively smaller it concerns the number of features extracted, which further causes the well-known problem of curse of dimensionality (also called as the Hughes Phenomenon) (Chang 2003).

Firstly, identification accuracy diminishes with rising in the dimensionality of training and validation data and with the introduction of hundreds of spectral channels.

Secondly, spatial resolution for identifying small objects is enough but raises the problem of high correlation between neighboring pixels.

Classifying each pixel with a certain land-cover or landscape feature is the resultant of hyperspectral image analysis which includes image segmentation, object recognition, land-use and land-cover classification, target detection, and so on. Image labeling is a mandatory task for remotely sensed hyperspectral image datasets. Under image labeling, a class label is assigned to each label, where each pixel ranges across a set of spectral channels. There are three main approaches for hyperspectral image classification: pixel-based, spectral-spatial-based, and object-based Ding et al. (2020). In a pixel-based approach, information is stored under each pixel for various features throughout each of these spectral channels which are thus, used to perform classification. In spectral-spatial, the neighboring unknown and unlabeled pixels are also used for classification. This part of image classification comes under transfer learning. The object-based method classifies objects on the basis of color, texture, shape, and spectral signatures profile underclass labels defined from one class into a cluster of one category (Guo et al., 2018).

The relevant methods which are based on the spectral-spatial approach are found to be more accurate compared to other methods as the challenge of dimensionality is taken well into consideration and also, pixels are spatially related. The spectral-spatial methodology has a diversified technique that includes clustering (k-means, ISODATA), label propagation, active learning, supervised (SVM), semi-supervised, and deep convolutional neural networks. Individual manual labeling is a time-consuming process. Thus recent research has shifted towards the development of classification models where few ground truth labels are enough to classify (Acquarelli et al., 2017).

Furthermore, 3D CNNs were included to extract deep spectral-

*Corresponding author.

spatial features from raw hyperspectral imagery thus producing good classification outcomes. Since 2015, the proposed deep residual neural network ResNet has shown an advancement to the convolutional neural networks that allow to update gradients and weight functions with deep convolutional architecture Chen et al. (2014). We propose here a Hyperspectral Imagery feature extraction method using convolutional neural networks combined with residual neural network (ResNet) based pixel-wise classification. Major highlights of this paper include :

1. Use of ResNet's for high spatial dimension feature extraction and spectral channel-based HSI classification.
2. To carry out validation, batch normalization, and gradient descent loss optimization for overcoming overfitting, underfitting, and falling accuracy results.
3. The check the significance of comparative residual block to learn spatial and spectral representations separately through which we can distinct and discriminate features that are finally extracted.
4. To predict and visualize the thematic classified map on the basis of ground truth labeled data and carry out the validation assessment.
5. To develop the identification and classification procedure for crop species and crop management intensity, which is a prerequisite for the generation of Spatio-temporal and is explicit of the thematic crop maps of selected transects in Bangalore, southern India.

Hyperspectral remote sensors capture reflected radiation of targets throughout the optical spectrum in hundreds of spectral bands with a very narrow spectral bandwidth (FWHM) thus capturing the specific interactions of the spectrum, matter, and energy. Several studies have used hyperspectral measurements in support of crop management such as crop type identification, crop stress or damage, growth status evaluation. We have acquired Terrestrial Hyperspectral (THS) ground-based and Unmanned Aerial Vehicle (UAV) / Drone-based very high-resolution hyperspectral imagery using sensor FireflEYE 185 - Cubert GmbH with the spatial resolution in the order of 8nm, spectral channels varying from 450-980nm sampled at 4nm bandwidth, with 139 spectral channels, and power consumption limited at 12V / 8W for the region Gandhi Krishi Vignana Kendra (GKVK), University of Agricultural Sciences, Bangalore, southern India. The acquired agricultural data comprises seasonal crops sowed from March-June for the year 2017. Crop patterns include Cabbage, Eggplant, and Tomato with varying nitrogen concentrations. Considering above mentioned challenges, recent work has worked in the direction to apply deep learning models to extract distinctive and discriminating features. Convolutional neural networks (CNN) extracts feature including edges, color, shape and integrates spatial and spectral features (Palsson et al., 2018).

The performance of assessment, prediction, and classification is expected to reduce if supervised learning is used when compared to active or transfer-based learning due to available labeled pixels such that no overlap between training and test pixels occurs. This learning process is mainly part of the pixel-based method. Recently, CNN's have n-number of applications in pattern recognition and computer vision tasks. The number of papers assures CNN can deliver state-of-the-art results using specialized inputs for HSI-based classification (Ozdemir 2016). 2D-3D CNN's showcased feature extraction spatially and on spectral channels that promise the best probable classification outcome. Small-sized 2D and 3-D input HSI cubes are studied for carrying out prediction and classification. These models generate thematic maps which process raw HSI but the challenge to overcome is that as the network goes deeper, the CNN model starts losing its significance. This is where architectures like AlexNet, VGG, GoogLeNet, ResNet have a keen role to play Yue et al. (2015). Here, we have employed ResNet with a 2-D CNN model to train, validate, test, predict, classify and finally perform accuracy assessment on a raw HSI cube data set.

To tackle this challenge inspired by [3] we propose a semi-supervised spatial and spectral residual architecture with multiple consecutive blocks taking HSI characteristics into the account. With captivation of spatial and spectral residual blocks, we extract discriminative features from HSI cubes which regard an extension to 2D CNN along with feature maps being extracted with Conv3D layered batch normalization Li et al. (2017). The total parameters include 257,229 and trainable parameters include 256,589 thus non-trainable parameters are 640 which is substantially very less.

Recent studies in face of these problems have tried to encompass supervised deep learning models to extract/segment out features for remotely sensed data for classification. For this, we are using Rectified Linear Unit (ReLU) as the activation function (Zhu et al., 2017).

$$f(x) = x^{+} = max(0,x) \tag{1}$$

ReLU solves the gradient vanishing problem and stops the inactive neurons. The other reason that it is used is because of how efficiently it can be computed compared to more conventional activation functions like the sigmoid and hyperbolic tangent, without making a significant difference to generalization accuracy (Acquarelli et al., 2017).

2. STUDY AREA AND DATASET

The study area chosen lies in southern India in the state of Karnataka in the metropolitan city of Bangalore. The dataset we have used is scanned using Cubert Hyperspectral FireflEYE SE S-185 sensor with a spectral resolution of 8nm with channel bandwidth ranging from 450 - 980 nm with the 139 spectral channels. This sensor has been mounted to UAV as well as Terrestrial Imager (THS) to scan crop patterns in the region of Bangalore, Gandhi Krishi Vignana Kendra (GKVK). Crops have been sown from the period of March-June, 2017 which include crop varieties such as cabbage, eggplant and tomato.

Figure 1: Data collections using Terrestrial Hyperspectral Imager and Drone mounted with hyperspectral sensor

The hyperspectral camera that is being used is mounted on a UAV platform and can is also mounted on the terrestrial laser scanning (THS) device. It has a wavelength range between 450-980 nm with 139 spectral channels. The resolution of the imagery is very fine ranging from 8 nm across band wavelength at 532 nm sampled at every 4 nm. Panchromatic image resolution is 1 Megapixel for every 2500 spectra per cube. The weight of the hyperspectral camera is 490 grams. Power consumption is 8 watts

with a 12 volts power supply. The spatial dimensions range between 1000 * 1000 with spectral channels ranging at 139. The camera can capture snapshots at sensor resolution 2:1 with cube resolution 50*50*139 with a maximum of 15 frames per second with measurements ranging from 1-1000 msec.

3. PROPOSED ARCHITECTURE AND RELATED WORK

The deep learning framework comprises hierarchical layers of nonlinear neurons that need to be triggered for creating feature maps out of big/large frames of labeled images. CNN is a popular semi-supervised learning technique that currently has shown the power of feature extraction in pattern recognition and computer vision tasks. Generally, CNNs contain convolutional layers, pooling layers, fully connected layers, and logistic or softmax-based multi-class regression layers. There are other techniques for hyperspectral image classification in response to CNN's using different architectures for single-layered pixels, vectored layered pixels, the patch of labeled/unlabeled pixels, the 3D cube of sample pixels, and so on. Most hyperspectral images require Principal Component Analysis (PCA) to reduce redundant data/ remove noise from the image data set. For dimensionality reduction, PCA plays a vital role in carrying out analysis of hyperspectral images (Acquarelli et al., 2017).

Cross-validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (called the validation dataset or testing set). The goal of cross-validation is to define a dataset to "test" the model in the training phase (i.e., the validation set), to limit problems like overfitting. Other techniques break down the sample dataset into training and test set by defining a small fractional ratio (Chang 2003).

Other such methods include Stacked Auto-Encoders (SAE) with 2D patch and PCA, contextual deep CNN, CNN with 1D pixel cubes, a deep CNN with 1D pixel spectra (Palsson et al., 2018).

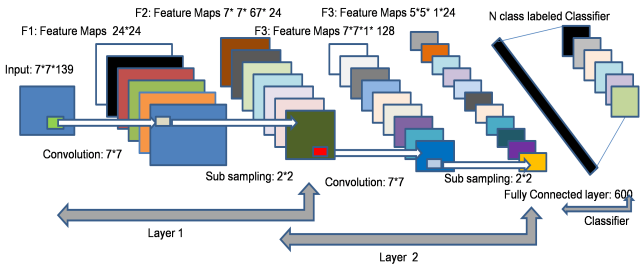

Figure 2: Architecture of 2D Deep Convolutional Neural Network on hyperspectral imagery

The representative trait of DCNN's in contrast to CNN-based models is their spectral-spatial structure, imposing coarse sparsity and hierarchy by reducing a number of hyper-parameters.The formulation of convolutional layers is:

$$F^j = ReLU(W^{j-1} * F^j + B^j) \tag{2}$$

In equation(2) F(ĵ) represents output of jth layer in the model W(ĵ-1) is kth convolutional filter bank,B(ûj) denotes bias of jth layer and ReLU(.) is an activation function.

To do the batch normalization, to regularize, and to speed up the training process the batch normalization layers need to be inserted into ResNet architecture to normalize and scale feature maps into intermediate batches thus, it helps to prevent the overfitting problem and thus, also smoothly converges towards global minima and does not require re-setting the hyper-parameters every time. The batch normalization is defined as below (Ioffe et al., 2015):

$$X^k = (X^k - E(X^k))/(VAR(X^k)) \tag{3}$$

In equation (3) X(î) denotes ith dimensional feature norm for X , E(.) represents the expected value and VAR(.) is the variance of the feature vector.

4. WORKFLOW AND FLOW-DIAGRAM

The below workflow will give a brief outline of how the whole process is being carried out. We will discuss in the coming section, how to deal with preprocessing of high-resolution imagery using PCA and wavelet denoising techniques to remove unnecessary noise and remove the redundant band information. The creation of training, validation, and test samples is the next step in extracting features as feature maps. The model which is used is ResNet pre-trained with 1.5 million images Duchi et al. (2011). The labeled categories with information are gathered for different class labels, these categories point to different classes. This comparison happens in the prediction phase of the model. The validation, training, accuracy, and errors are predicted as a function of different hyperparameter tuning. The cross-entropy function, cost/loss function, actual and predicted output is determined Prasad et al. (2008). In general terms, cross-entropy is equal to predicted label vector output - ground truth label vector. In terms of formula it is as mentioned below:

$$Error = \frac{1}{2} * (Y_{actual} - Y_{predicted})^2 \tag{4}$$

The validation group keeps a check-up on the cross-validation after the model has been trained and helps to predict the unknown

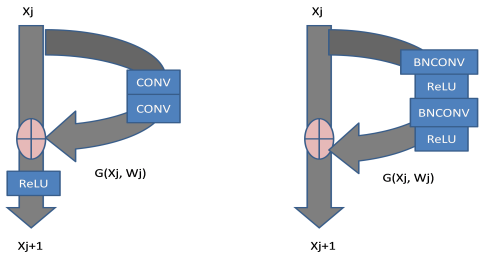

Figure 3: Basic Architectural residual blocks. Residual framework without batch normalization (left) and with batch normalization (right)

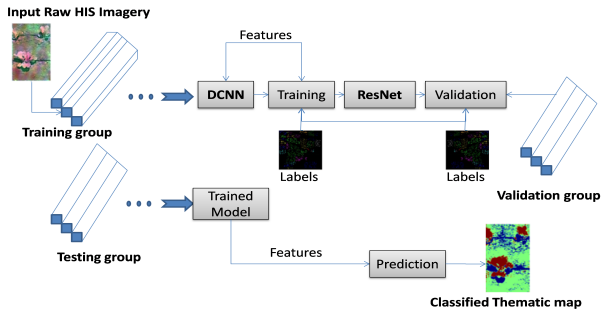

Figure 4: Process Workflow

test data. Monitoring the whole training set such that it measures classification with validation, training, loss, and accuracy as a function of epochs and iterations are carried out. Finally, the testing group is used for the evaluation of unknown samples. The trained model thus calculates on the basis of quantitative analysis, prediction, and creates a thematic map of the classified image (Petrakos et al., 2001).

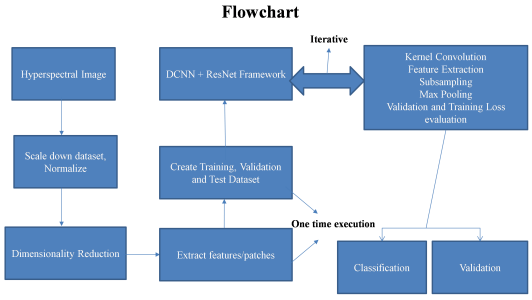

Figure 5: Flow-Diagram

Here, the 3D convolutional layered response with the induction of batch normalization and CONV-3D convolutional layer with ReLU function setting the weights and other relevant operations guides us to produce desired classification quantitative analysis and visualization thematic maps Guo et al. (2018). The proposed network gives a general description of the cycle of workflow as mentioned below:

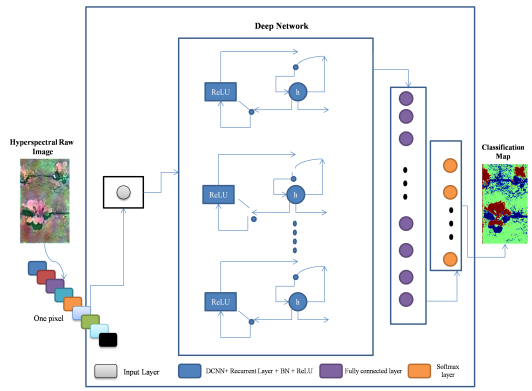

Figure 6: Deep learning with ResNet Recurrent neural network for hyperspectral image classification: Proposed deep network

$$X_i = ReLU\,(log \sum_{i=1}^{N} X_k * H_i k + b_i k) \tag{5}$$

5. RESULT AND DISCUSSION- PART I : PRINCIPAL COMPONENT ANALYSIS AND WAVELET NOISE REDUCTION TECHNIQUE

Principal Component Analysis (PCA) plays a significant role in hyperspectral image/spectral classification. Here, similar co-related features which have the same distribution spread for them eigenvalues and eigenvectors are found to have similar stats and un-correlated features which are determined to have maximum variance and spread for this we use covariance matrices (Prasad et al., 2008).

$$cov(A) * W = \lambda * W \tag{6}$$

$$\det(\lambda - A * I) = 0 \tag{7}$$

Here, cov (A) is the covariance matrix. W is trans coupling or transformation matrix, λ is eigenvalues of the covariance matrix. How does one determine if some eigenvectors are orthogonal or not? It is determined with the covariance matrix being non-imaginary and non-symmetric or vice versa. The eigenvector corresponds to the first eigenvector having maximum variance and maximum spread. How much dimensional space is required is determined by eigenvalues, eigenvectors, variance, and spread (Prasad et al., 2008). Single wavelet denoising transform is mainly used in areas of image compression, feature segmentation, feature extraction, image fusion, segmentation Prasad et al. (2008). Wavelet transform decomposes the signal into smaller levels of decomposition using scaling $\tau(t)$ and wavelet transform techniques $\xi(t)$:

$$\tau_{x,y}(t) = 2^{x/2} \sum_{i=1} h(y)\tau(2^x t - y) \tag{8}$$

$$\xi_{x,y}(t) = 2^{x/2} \sum_{i=1} g(y)\xi(2^x t - y) \tag{9}$$

Here h(y) and g(y) are low pass and high pass filter coefficients and x is the scaling factor. Inverse wavelet transform is applied to generate the decomposed or compressed signal back to the original signal.

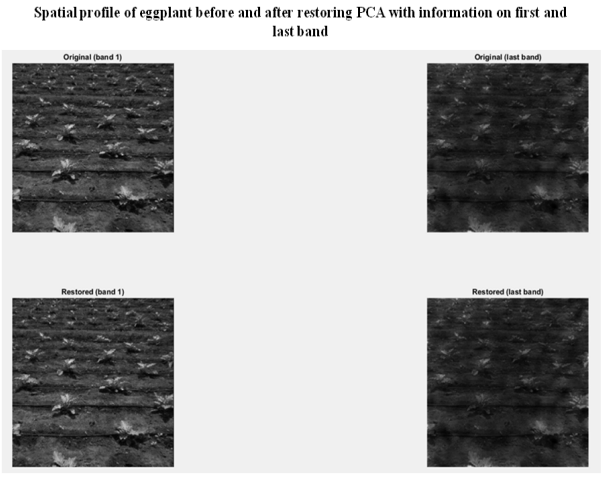

Figure 7: Spatial profile of eggplant before and after applying PCA + wavelet denoising technique

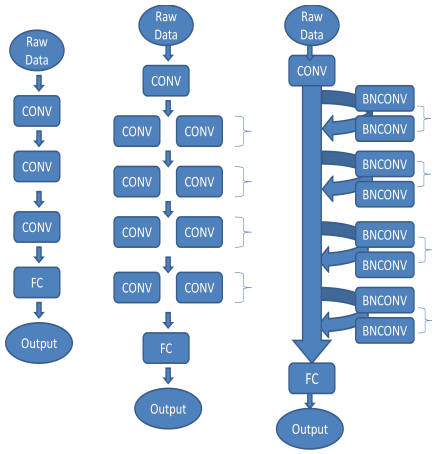

Figure 8: Example of CNN and ResNet models. CNN-4(left) CNN-10(middle)and ResNet-10(right)

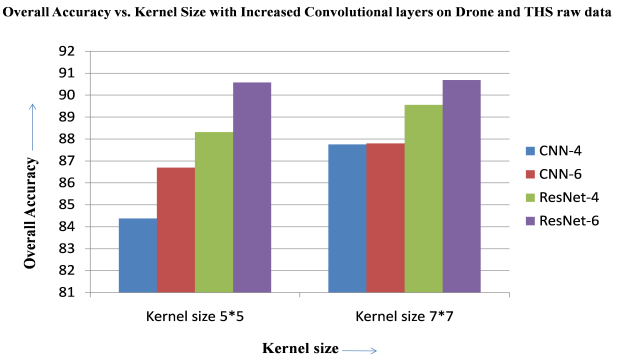

Figure 9: Overall Accuracy vs Kernel Size with varying convolutional layers and residual blocks

6. RESULT AND DISCUSSION- PART II: CONVOLUTIONAL NEURAL NETWORK (CNN) MODEL AND ARCHITECTURE

A baseline CNN does not constitute any more than one hidden layer. Unlike any deep learning model or framework with ResNet, we use CNN has no set of pooling or sub-sampling layers or fully connected layers and it has only just one convolutional layer (Yue et al., 2015). The use of loss function for training the standard L2 regularized cross-entropy error function and cross-validation on a training pixel sample set (x,y) of P pixels is represented as :

$$LOSS_{CNN}(W) = -\frac{1}{P}\Sigma_{i=1}^{P}\left\langle y_i log y_i + (1-y_i) log(1-y_i)\right\rangle + (\lambda_1 . \|W\|^2)$$

Here, in equation (5) first term means cross-entropy error loss minimization function and second terminology is L2 regularization. Also $y_i = \psi_2(w_2.\psi(w_1.x_i))$ is network's output,ψ_1 and ψ_2 are activation functions , x_i is i-th pixel , W = [w1,w2] are the weights from input to single hidden layer (w1) and from hidden layer to output (w2) and y_i is target label (Yue et al., 2015).

The hyper-parameters of CNN include:
. Learning Rate (η) (Default value used: 0.003)
. Momentum (Default value used: 0.9)
. Number of convolutional layers (Default value: 3)
. Size of convolutional kernels N (Default value used: 3,5,7)
. Stride for convolution s ([1 1 1])
. L2 regularization constant λ
. Number of Epochs/Iterations (Default: epoch count is 12)

Table 1: Architecture for CNNCONV and ResNet based Deep Learning Models

Architecture	Convolutional Layers	Residual Blocks
CNNConv-4	3	None
CNNConv-6	5	None
ResNet-4	1	1
ResNet-6	1	2

The details of the architecture is mentioned in the above table and the details of different layers of CNN and ResNet with varying kernel sizes is mentioned below.

7. RESULT AND DISCUSSION- PART III: DEEP RESIDUAL NETWORK MODEL AND ARCHITECTURE

To effectively extract high-level invariant features, the Deep Residual Networks model is applied. It can progressively learn deep spatial-spectral features of hyperspectral data layer by layer and extract high-level features. Using softmax regression which is based on creation of high feature maps and visualizing classified thematic map. Creating deep networks is not as simple as adding layers Zhu et al. (2017). One problem is the vanishing gradient problem, which affects the hyper-parameter as it effects the rate at which converging occurs. Another problem is the rate of fallout, if the depth of a network increases, the accuracy doesn't always rise but instead, it starts getting saturated and falls outs rapidly Zhu et al. (2017). Residual Network can be equated as mentioned below:

$$B(x) = A(x) + x \tag{10}$$

Here in Eqn. (8), $A(x)$ is termed as Residual function with input x. Also $B(x)$ is a non-linear function.To determine how $A(x)$ is related to $B(x)$ we can determine a co-relation as:

$$A(x) = W_3 * W_2 \omega(W_1 x) + x \tag{11}$$

Here, W_1, W_2,W_3 represents weights for different convolutional layers and hidden layers. Also, ω represents an activation function which is used to trigger an impulse thus, activating neurons to update weights and biases. We have used ReLU (Rectified Linear Unit) as an activation function to trigger the neurons and update the weights and biases Zhu et al. (2017). Finally, how well the residual block is equated is mentioned as:

$$X_j = \omega(\omega W_3(\omega W_2(\omega(W_1 x)))) + x \tag{12}$$

The relation between output of the jth unit along with batch normalization, for W_j denoting weight parameters for the residual structure is as shown below:

$$X_{j+1} = max(0, X_j + G(X_j, W_j)) \tag{13}$$

The details of invariant and extracted high-end features is as mentioned below:
Table 2: Details for different layers of CNN based layers and ResNet layers with varying kernel sizes

Kernel Layers	CNN-4	CNN-6	ResNet-4	ResNet-6
Kernel size (5*5)	84.38	86.7	88.32	90.58
Kernel size (7*7)	87.76	87.80	89.56	90.69

1. Extraction of feature maps under initial phase: Firstly, from the hyperspectral cube having 7*7*139 dimensions we are creating the feature maps by reshaping raw pre-processed denoised images and setting to 1-D spectral features. This is achieved by flattening the raw image using pixel-based kernel operation (convolution), sub-sampling and doing the pooling operations Zhu et al. (2017). The original raw reshaped is sub-categorized into Ω spectral responses with deviation ζ the formula is as below:

$$s_j = v(k), 1 \leq j \leq \Omega, \zeta(j)+1 \leq k \leq \zeta(j)+W \qquad (14)$$

With the bias W, s_j is the jth spectral response. Before doing the pre-processing, v is the initial spectral shape length. To get the feature map, we get the two parts of basic spectral response as independent variables. Thus feature map is given by:

$$F_k = \sqrt{(s_i.s_j^T)}, 1 \leq i, j \leq \Omega, 1 \leq k \leq \Omega \qquad (15)$$

where F_k is kth spectral feature map extracted and Ω is the number of feature maps determined as part of the extracted features. Thus, we get Ω spectral features from 1D flattened image. We reshape the flattened 1D shape image into 2D residual maps which are passed as an input to the deep learning model inclusive of the ResNet architecture Zhu et al. (2017), Yue et al. (2015).

2. Up-sampling, Down-sampling and softmax regression classifier: At the output layer predicting a discrete variable whether a grid of pixel intensities is represented by labels as 0 or a 1 or a 2 or a 3 or a 4 label. This is a classification and a label problem. Logistic regression which is a simple classification algorithm was used for learning to make such decisions Zhu et al. (2017).

$$P(y=1|x) = h_\Theta(x) = \frac{1}{1+exp(-\Theta^T(x))} = \sigma(\Theta^T(x)) \qquad (16)$$

$$P(y=0|x) = 1 - P(y=1|x) = 1 - h_{\Theta(x)} \qquad (17)$$

For the logistic regression SoftMax has been used to predict the value of y_i for the ith example x_i using a linear function: $y = h_\Theta(x) = \Theta^T * \mathbf{x}$ This is clearly not a great solution for predicting labeled-values $y_i \in 0,1,2,3,4$ and so on (Zhu et al., 2017).

The input to a convolutional layer is a 7 * 7 * 139 image where 7 is the height and width of the image and 139 is the number of spectral band channels. The convolutional layer will have k filters (or kernels) of size 5*5*24 where 5 represents the smaller dimension of the image and 24 is the smaller subset of band spectral channels than total bands of 139 obtained with different kernel window filters. The size of the filters gives rise to the locally connected structure which is each convoluted with the image to produce k feature maps of size 7−5+1. Each map is then sub-sampled typically with mean or max-pooling over p*p where p $\in$ (2,5). After activation with some ReLU layers, we can add the pooling layers. Thus, it helps to down-sample the feature maps. A kernel window of minimum size 2*2 and stride of length 1, helps to keep applying at the input volume and output the size of 4*4 or 8*8 size (Li et al., 2017).

3. Identifying total number of feature maps, hidden layers: As more number of layers add the chances of test accuracy may reduce even if training accuracy is very high (the problem of overfitting). And another problem that can arise is high bias and low variance in the training data (the problem of underfitting). The right set of hidden layers and activation function is used to minimize the weight factors and remove the vanishing gradient challenges (Myasnikov 2016).

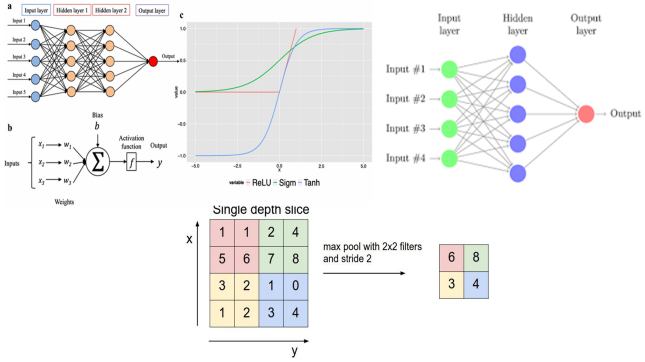

Figure 10: a. Neural Network Layer, b. Inputs, weights, bias, activation function and output, c. Plot difference between ReLU, sigmoid and Tanh, d. Hidden layers, e. Depth slicing with 2*2 max pooling kernel window and stride 2

$$y(x) = h_\Theta(x) = \Psi * (W^\top \mathbf{x} + bias) \qquad (18)$$

Here, Ψ represents the activation function which can either be: ReLU, sigmoid or tanh.

4. Batch size, epochs and iterations: Batches of total samples are fed to a hidden layer for training the model. Whenever batch size is defined in the equation as a hyperparameter it represents the sample size that are used for training, validation, testing. Smaller the batch size better is the response of the hyperparameter tuning (Ioffe et al., 2015).

5. Learning Rate (η): Learning rate is the order in which weights are trained and fed into forward or backward propagation. These are applied mostly to weights and biases at time of backward propagation Ioffe et al. (2015). The learning rate used for this study was in the order of 0.01 for stochastic gradient descent (SGD) optimizer and 0.0003 for RMS prop optimizer. The weight vectors as a function of learning rate and bias can be represented as:

$$W_i = W_i - \eta * \frac{\partial W_i}{\partial t} + bias(b) \qquad (19)$$

6. Gradient descent, Log-Likelihood function: Gradient descent optimizes the weight such that it has to reduce the cost function iteratively moving in the direction of the local minima and the descent and is defined as the negative gradient of weights as function of time. This cost or loss function needs to be minimized to reduce validation errors (Myasnikov 2016).

$$F = \frac{1}{P}\sum_{i=1}^{P} F_k \qquad (20)$$

Here F_k points to loss likelihood or cross-entropy function.

$$F_k = -\log(p(y_k|x_k)) \qquad (21)$$

In other words, how would we go about calculating the partial derivative of cost/loss function with respect to θ of the

cost function (the logs are natural logarithms) is given as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^i \log(h_\theta(x^i)) + (1-y^i)\log(1-h_\theta(x^i)) \tag{22}$$

Our aim using gradient descent is to normalize and reduce the effect of cost/loss function.

8. RESULT AND DISCUSSION- PART IV: OPTIMISER, ERROR PROPAGATION REDUCTION

With the created ground truth labels, classification, prediction and thematic classification maps are created with 6 different optimiser. Finally, the best suited model is used for prediction and visualization. Ground labeled image is added in the semi-supervised learning Kuo et al. (2011).The best suited optimiser and reduction in error propagation is mentioned below:

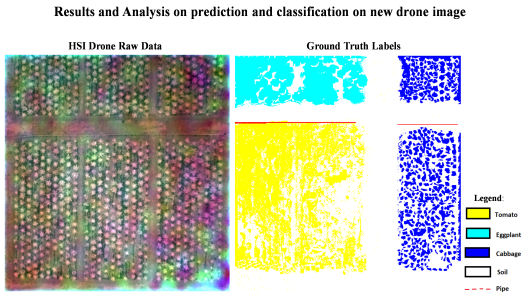

Figure 11: HSI Raw imagery and ground truth labeled image

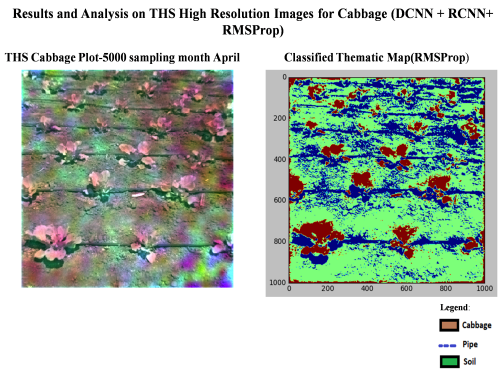

Figure 12: Results and analysis on THS high resolution imagery for cabbage using DCNN + RCNN + RMSProp optimiser

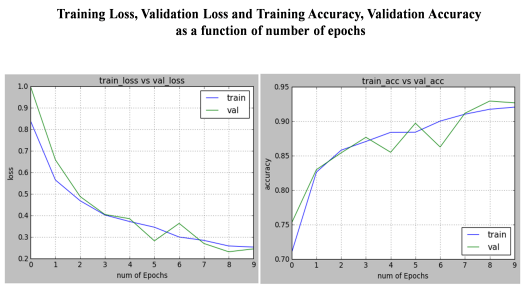

Figure 13: Training loss, validation loss, training accuracy, validation accuracy as function of number of iterations and epochs

Table 3: Confusion Matrix using RMSProp

	Soil	Cabbage	Pipe	Accuracy
Soil	2,55,456	8,822	119	96.61%
Cabbage	19,601	5,84,335	3,496	96.19%
Pipe	104	6,774	1,21,516	94.6%
Reliability	92.83%	97.40%	97.11%	O.A

Overall Accuracy = 96.13%, Users Accuracy = 95.78%
Producers Accuracy = 94.81%
Omission Error(underestimation) = 1- users acc = 0.042
Commission Error(overestimation) = 1- producers acc = 0.051

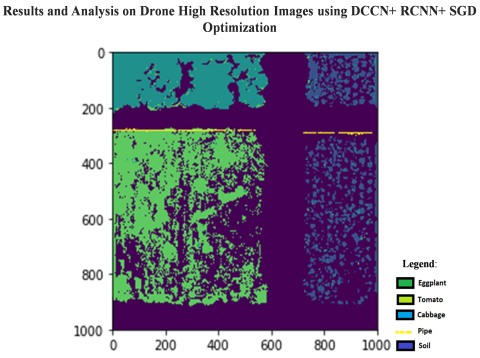

Figure 14: Results and analysis on drone high resolution imagery using DCNN + RCNN + SGD optimiser

Figure 15: Training loss, validation loss as function of number of iterations and epochs

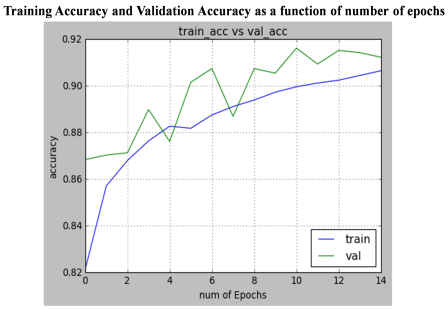

Figure 16: Training accuracy, validation accuracy as function of number of iterations and epochs

9. LIMITATIONS AND CONCLUSION

Limitations: The limitations of this research can be listed in a few major points.

1. Challenges occurred in Transfer learning responses were much larger than anticipated, in which ground truth labels prepared in a particular set were unusable in cases where the test sample sets had changed.

2. Certain optimizers performed better whereas some had extremely poor performance.

3. ResNet model with deepened convolutional layers showed a diminishing performance in accuracy.

4. Complex architecture hampers the overall performance by increasing execution time.

Conclusion :

In the end, this research study works on the use of advanced deep learning models and techniques for spatial-spectral crop classification of UAV and THS hyperspectral images. Most of the research work during earlier phases was confined to the use of complex model architectures for standard but old datasets. A lot of research studies and questions were raised on these standalone datasets by advanced deep learning model architectures which in time became obscure in nature. This paper breaks the barrier and addresses fresh problems and challenges pertaining to high-resolution imagery. It also adds an additional possibility of seeking multiple ways and means of determining ground truth labeled data.

REFERENCES

Acquarelli, J., Marchiori, E., Buydens, L. M., Tran, T., van Laarhoven, T. (2017). Convolutional neural networks and data augmentation for spectral-spatial classification of hyperspectral images. Networks, 16, 21-40.

Bruzzone, L., & Serpico, S. B. (2000). A technique for feature selection in multiclass problems. International Journal of Remote Sensing, 21(3), 549-563

Chang, C. I. (2003). Hyperspectral imaging: techniques for spectral detection and classification (Vol. 1). Springer Science and Business Media.

Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y. (2014). Deep learning-based classification of hyperspectral data. IEEE Journal of Selected topics in applied earth observations and remote sensing, 7(6), 2094-2107.

Ding, H., Xu, L., Wu, Y., Shi, W. (2020). Classification of hyperspectral images by deep learning of spectral-spatial features. Arabian Journal of Geosciences, 13(12), 1-14.

Duchi, J., Hazan, E., Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7).

Grahn, H., Geladi, P. (Eds.). (2007). Techniques and applications of hyperspectral image analysis. John Wiley & Sons.

Guo, A. J., Zhu, F. (2018). Spectral-spatial feature extraction and classification by ANN supervised with center loss in hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing, 57(3), 1755-1767.

Ioffe, S., Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR.

Jiang, J., Sun, H., Liu, X., Ma, J. (2020). Learning spatial-spectral prior for super-resolution of hyperspectral imagery. IEEE Transactions on Computational Imaging, 6, 1082-1096.

Kuo, B. C., Chen, I. L., Li, C. H., & Hung, C. C. (2011, July). Combining ensemble technique of support vector machines with the optimal kernel method for hyperspectral image classification. In 2011 IEEE International Geoscience and Remote Sensing Symposium (pp. 3903-3906). IEEE.

Li, Y., Zhang, H., Shen, Q. (2017). Spectralspatial classification of hyperspectral imagery with 3D convolutional neural network. Remote Sensing, 9(1), 67.

Myasnikov, E. (2016, July). Evaluation of stochastic gradient descent methods for nonlinear mapping of hyperspectral data. In International Conference on Image Analysis and Recognition (pp. 276-283). Springer, Cham.

Ozdemir, A., Polat, K. (2020). Deep learning applications for hyperspectral imaging: a systematic review. Journal of the Institute of Electronics and Computer, 2(1), 39-56.

Palsson, B., Sigurdsson, J., Sveinsson, J. R., Ulfarsson, M. O. (2018). Hyperspectral unmixing using a neural network autoencoder. IEEE Access, 6, 25646-25656.

Petrakos, M., Benediktsson, J. A., & Kanellopoulos, I. (2001). The effect of classifier agreement on the accuracy of the combined classifier in decision level fusion. IEEE Transactions on Geoscience and Remote Sensing, 39(11), 2539-2546.

Prasad, S., Bruce, L. M. (2008). Limitations of principal components analysis for hyperspectral target recognition. IEEE Geoscience and Remote Sensing Letters, 5(4), 625-629.

Yue, J., Zhao, W., Mao, S., Liu, H. (2015). Spectralspatial classification of hyperspectral images using deep convolutional neural networks. Remote Sensing Letters, 6(6), 468-477.

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geoscience and Remote Sensing Magazine, 5(4), 8-36.