A Project Report

on

Hyperspectral image classification

Submitted in partial fulfillment

of the requirement for the

award of the degree of

Bachelor of Technology in Computer Science and Engineering

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of

Dr. T.Poongodi

Assistant Professor
Department of Computer Science and Engineering

Submitted By

18SCSE1010547- MUDIT JAISWAL

18SCSE1010698- RACHIT SETIA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA DECEMBER - 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **" HYPERSPECTRAL IMAGE CLASSIFICATION"** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**
submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY-2021 to DECEMBER-2021**, under the supervision of **Mr.V. ARUL, Assistant Professor**, **Department of Computer Science and Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

18SCSE1010547- MUDIT JAISWAL
18SCSE1010698 - RACHIT SETIA

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(DR.T.Poongodi, Assistant Professor)

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **18SCSE1010547 - MUDIT JAISWAL, 18SCSE1010698 - RACHIT SETIA** has been held on ____and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

ABSTRACT

Hyperspectral imaging is the collection and processing information from across the electromagnetic spectrum. The primary goal of hyperspectral imaging is to obtain Hyperspectral spectrum for each pixel in the image scene for the purpose of detection of objects or classification. Hyperspectral imaging (HSI) uses continuous and contiguous ranges of wavelengths (e.g. 400 - 1100 nm in steps of 1 nm). Hyperspectral deals with imaging narrow spectral bands over a continuous spectral range, producing the spectra of all pixels in the scene. A sensor with only 20 bands can also be hyperspectral when it covers the range from 500 to 700 nm with 20 bands each 10 nm wide. Hyperspectural remote sensing is used in a wide array of application such as agriculture, surveillance, minerology, environment and civil engineering. Organizations such as NASA and the USGS have catalogues of various minerals and their spectral signatures, and have posted them online to make them readily available for researchers.Classification refers to a predictive modelling problem where a class label is predicted for the given input data. The support vector machine (SVM) is the most popular supervised learning algorithm that has been successfully applied to solve many classification problems. On the other hand, a more recent approach grounds in convolutional networks so popular in multimedia vision: Convolutional Neural Networks (CNN).

Table of Contents

List of Figures

List of Tables

Acronyms

SVM	Support Vector Maching
ML	Machine Learning
DL	Deep Learning
CNN	Convolution Neural Networks
MDC	MINIMUM DISTANCE CLASSIFICATIONS

CHAPTER-1

Introduction

HYPERSPECTRAL remote distinguishing is generally called imaging powerful remote recognizing. It in a general sense involves imaging advancement and absurd technology. The hyperspectral distant sensor get the 2-D numerical spatial information similarly as the 1-D absurd information of the important locale. By and by the hyper spooky data have the development similarly as sort of the image strong shape. During imaging the spatial features of target district, each and every one of the spatial pixel is scattered molding a colossal number of limited gatherings for steady supernatural incorporation. HSIs combine picture information similarly as extraordinary information. The image information shows the external credits of the model like shape, surface and size. Additionally, during this time the extraordinary information shows the qualifications in the genuine developments and manufactured construction of the given model. Thusly, the HSIs shows the broad ascribes of the image. Hyperspectral remote distinguishing development has in like manner been used in regular approach, mineral exploration, accuracy cultivating. By and by days significant learning development has came and by and by all the assessment establishments similarly as scientists used this advancement for the HSI gathering field. Earlier, most of the methodologies are dependent upon appalling information orspatial information for the request. Nevertheless, they are adequately not to remove features with adequate isolation. From some time explores joined both of them ridiculous and spatial information to manage gathering tasks and this is giving incredible results.

SUPERVISION CLASSIFICATION

The controlled characterization strategy is a generally utilized hyperspectral picture order method. The fundamental interaction is first, choosing the discriminant standards based on the well-known example classification and initial data and work out the discriminant work; Generally utilized directed characterization techniques incorporate assist vector machine method, counterfeit neural organization arrangement strategy, choice tree grouping technique, and most important probability order technique.

Support Vector Machines

Support Vector Machine (SVM) is a directed order strategy suggested by Boser et al. In light of factual hypothesis as well as in view of the standard of restricting underlying risk, it tackles a quadratic imperative with disparity requirements. As an AI technique, the help vector machine strategy presumes a colossal segment in photograph and sign handling and acknowledgment. SVM put the important danger minimization guideline to a direct classifier to notice the perfect order surface. It compels that the characterization surface can’t divide both sorts of test focus without mistake yet additionally augment the order hole between the both kinds.

Suppose:

image $X = \{x_1, x_2, \cdots x_n\}$,

$x_i = \{x_{i1}, x_{i2}, \cdots, x_{iD}\}^T$ represents the spectral vector of the pixel i in the image, D represents the total number of bands, and n is the total number of pixels. In addition, we define $y = (y_1, y_2, \cdots, y_n)$ the classification mark image, in the formula $y_i \in (-1, 1)$. The mathematical process of a classic SVM classifier is

$$y_i = \operatorname{sgn}\left(\sum_{j=1}^{l_n} y_i \alpha_i \left(x_j^T \cdot x_i\right) + b\right). \tag{1}$$

Among them, l_n is recorded as the number of a priori marks, and α_i is a soft interval parameter. By setting $b = 0$, the optimal classification plane can pass the origin of the coordinate system, thereby simplifying the calculation. In practical operations, the situations we encounter are not all linearly separable, so we introduce slack variables, the mathematical expression of the support vector machine after introducing the slack variable is

$$\begin{gathered} \max \sum_{i-1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathrm{X}_i^T \mathrm{X}_j\right) \\ \text{s.t. } 0 \le \alpha_i \le C, i = 1, \cdots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0. \end{gathered} \tag{2}$$

Minimum Distance Classification:

Least distance classifier (MDC) is a controlled characterization based on the distance of pixels in the element space like an order premise. It is for the most segment thought to be that in the element space; representing directs having a place toward alike class are bunched in space. The mean vector is managed by these component focuses is used as the focal point of the categorization, and the covariance network is used to show the dispersing of encompassing focuses. Focuses are also evaluated with each categorization. The fundamental suspicion of the closeness measure is if the element differentiate between the both modes are under a limit, both modes are assume to be equivalent.

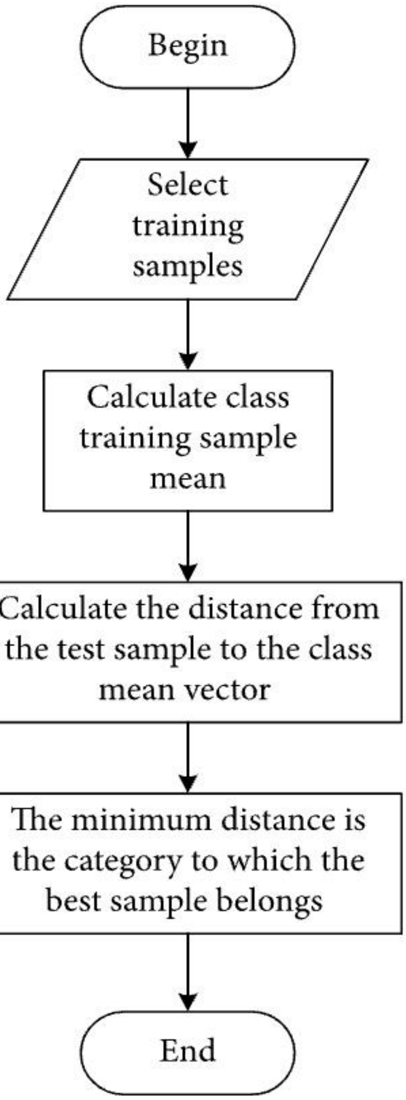


Figure 1: Flowchart representing HSI classification

Neural Network Classification

Neural Networks (Artificial Neural Networks, ANN) is the best man-made consciousness clustering technique as of now. It is represented by mimicking the handling as well as handling of information by human neurons. It has been usually used in savvy control, data handling, and combinatorial advancement. Nonetheless, fake neural organizations moreover have their own drawbacks, like the need for a lot of preparing data, more slow activity speed, and difficulty in obtaining choice surfaces in the component space. Neuron characterization methods are usually used in BP neural organizations , spiral premise neural organizations, and wavelet neural organizations . Among them, the BP neural organization model (feedforward network model) is now the most used neural organization model. It consists of an info layer, a secret layer, and a result layer. At the point when an data mode is given, the info signal is from the information layer to the transmission of the result layer is a forward engendering process.preassumimg there is a blunder between the result signal as well as the perfect sign, the mistake is taken to the regressive spread cycle, and the heaviness of every layer is converted by the extent of the mistake of each layer.

Convolution Neural Network

Characterization strategy based on ghastly elements: Hyperspectral photographs have exceptionally rich unearthly information as well as suprising high phantom goal. Each pixel can extract one-dimensional unearthly vectors. These vectors are created from of phantom data. Order using only one-dimensional unearthly vectors is called as a grouping strategy based on ghostly data. In the order technique based on unearthly data, by and large, the pixel is used to delete otherworldly data or to obtain specific explicit highlights by phantom data through include extraction to manage. Utilizing CNN to group otherworldly elements of hyperspectral photographs is to use one-dimensional CNN (1D-CNN) to detach ghastly elements as well as order them.

HYPERSPECTRAL IMAGE

Hyperspectral imaging (HSI), also known as hyperspectral spectroscopy gives rich information both spatially as well as frightfully. Hyperspectral spectrometers are a team of optical sensors that work otherworldly brilliance from every pixel in a scene. The result is the alleged hyperspectral photograph. The successive spectra on every place of the photograph can carry in excess of 200 neighbour restricted channels wraping apparent as well as related infrared frequencies at a high phantom target. This reality upgrades the capacity of HSI to separate different objects through their special phantom mark. Hyperspectral sensors are generally carry on satellites or airplanes and are used for an large number of remote detecting benefits : line reconnaissance, mine recognition, farming observing, territory investigation for minerals or soil types portrayal, and so on.

Hyperspectral photographs carry a large amount of information, however deciphering them needs for a profundity appreciation of precisely what properties of ground materials we are attempting to gauge, and how they are identified with the sensor estimations. Scene data is obtained in the deliberate brilliance as a piece of nonstop space factor, frequency and time factors. A HSI picture is stated by the obtention of a pile of images addressing the brilliance in the contrast frequencies. Instantly, the HSI detectors provide three-dimensional knowledge chunk with spatial- spatial-unearthly segments.

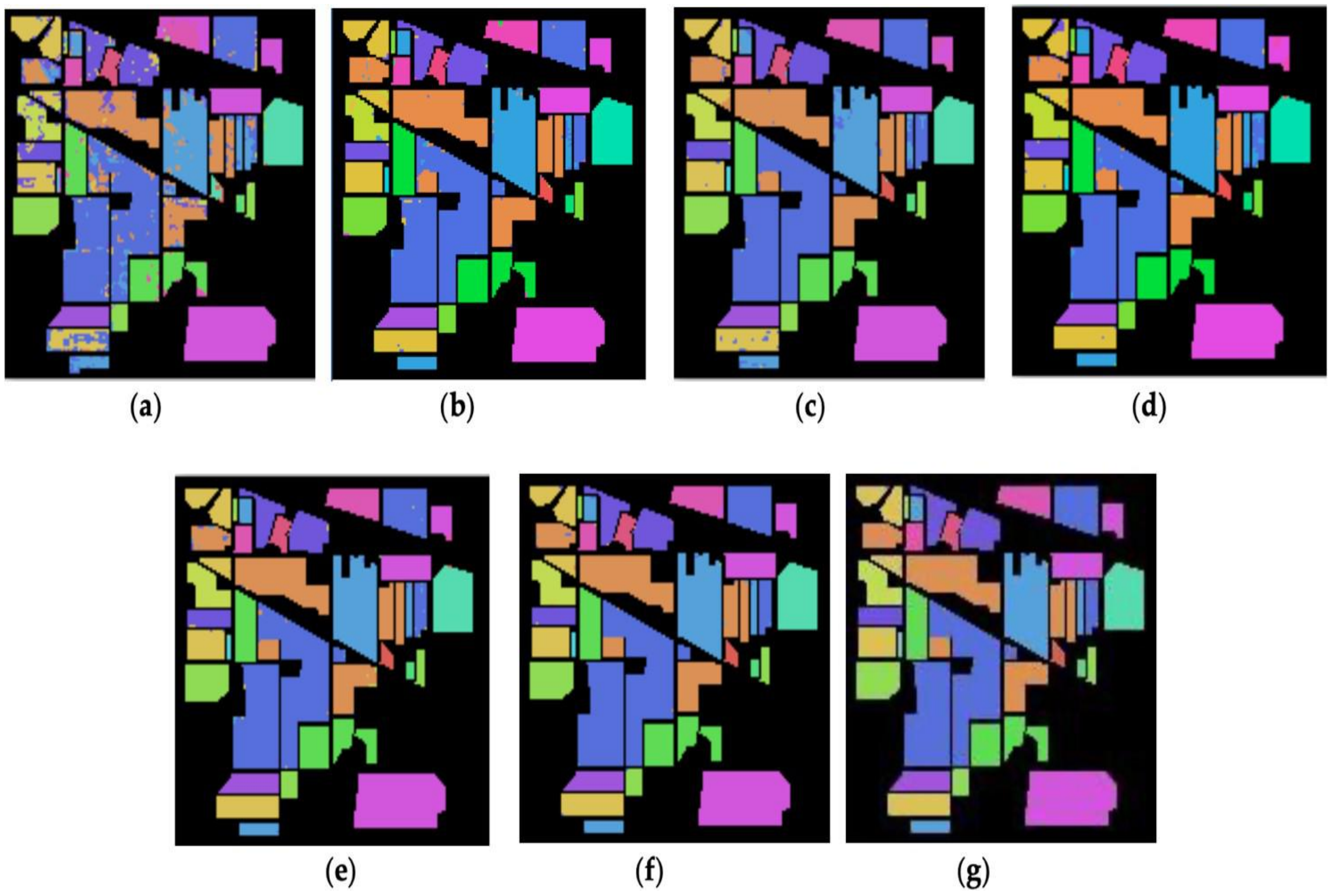


Figure 2: HSI visualization

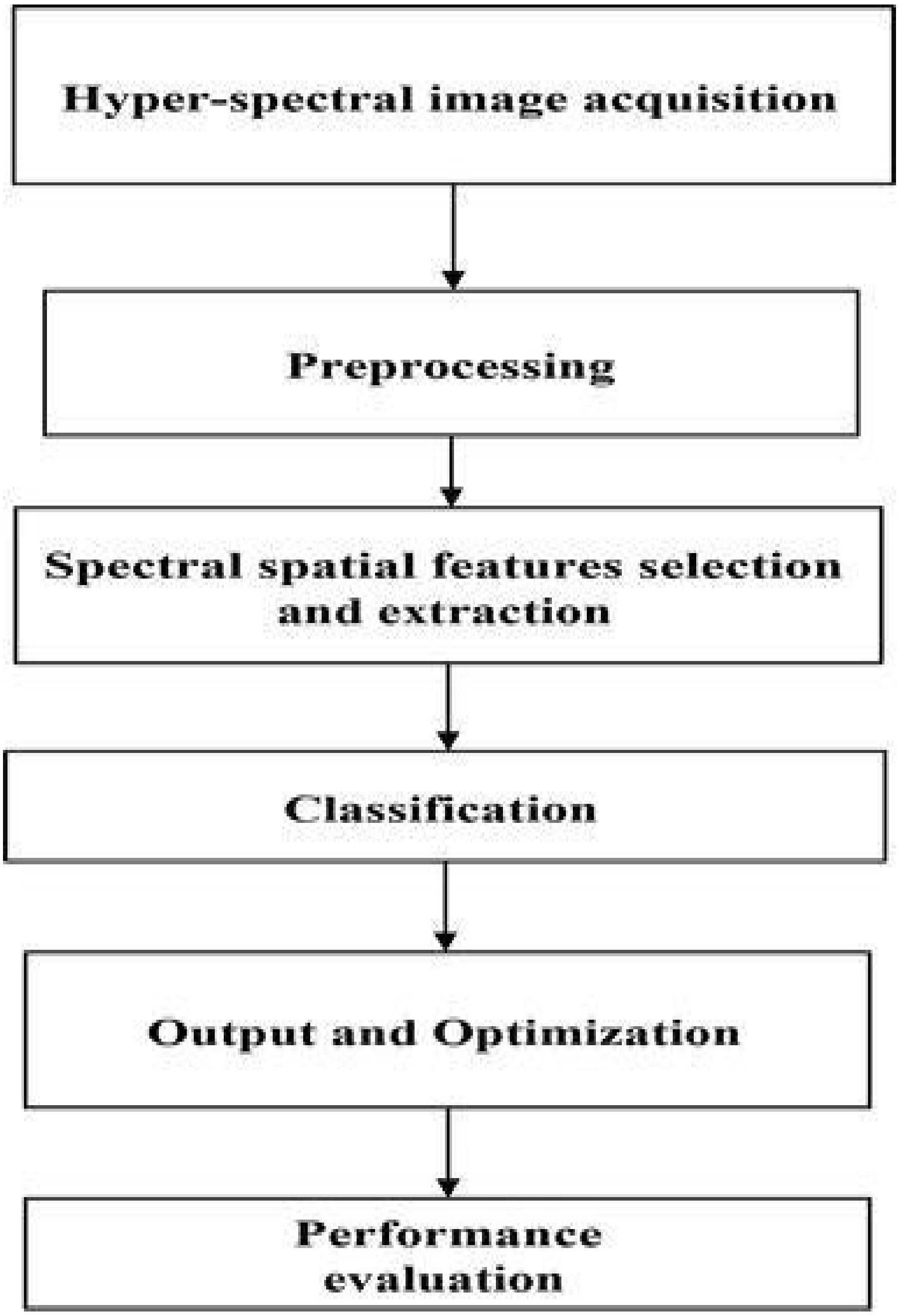


Figure 3: Flowchart representing the processing

Chapter 2

Issues/Challenges with the Existing Work

It is evident from the literature that using just 2-D-CNN or 3-D-CNN had a few shortcomings such as missing channel relationship information or very complex model, respectively. It also prevented these methods from achieving better accuracy on HSIs. The main reason is due to the fact that HSIs are volumetric data and have a spectral dimension as well. The 2 -D-CNN alone is not able to extract good discriminating feature maps from the spectral dimensions. Similarly, a deep 3-D-CNN is more computationally complex and this alone seems to perform worse for classes having similar textures over many spectral bands.

Objectives of the Proposed Work

The objective of the work is to classify and validate the results for Hyperspectral image of Indian pines dataset prove the effectiveness of the method. Hence, the 3 -D-CNN and 2 -D- CNN layers are assembled for the proposed model in such a way that they utilize both the spectral as well as spatial feature maps to their full extent to achieve maximum p ossible accuracy.

Chapter 3

LITERATURE REVIEW

Image classification using hyperspectral data (as opposed to RGB/multispectral data) is very effectual in critical applications. However, due to the existence of large number of spectral bands, hyperspectral image classification often turns out to be a computationally demanding task. Although, various learning algorithms (e.g., K-nearest neighbours, Gaussian maximum- likelihood classification, and random forests) have been successfully put in to solve the classification problem, support vector machine (SVM) is the most popular non-parametric classifier among all. problem, support vector machine (SVM) is the most admired non- parametric classifier among all. The two main reasons behind SVM's popularity are its availability in many software libraries, and its relative independence on the feature of the input data.

SVM has been widely applied in respective studies on hyperspectral image classification, and it is deemed to be the most dominant and efficient learning algorithm available in the literature.

Braun et al. implemented and compared different forms of SVM (based on the loss function) using a Gaussian kernel to tabulate hyperspectral data.

Shao and Lunetta applied SVM for characterizing land cover using moderate resolution imaging spectroradiometer data, and used the Gaussian kernel as the primary SVM kernel. Support Vector Machines (SVM) aims at procuring the optimal Hyperspectral remote sensing is used in a wide array of applications such as agriculture, surveillance, minerology, environment and civil engineering.

Organizations such as NASA and the USGS have catalogues of numerous minerals and their spectral signatures, and have posted them online to make them readily available for researchers.

ocused at the issues that the handcraft network formed cannot be altered well to different types of datasets; they also preferred the pre- programmed CNN models called 1-D auto-CNN and 3-D auto-CNN for HSI classifying. Initially, a search algorithm established on the gradient descent is used to adeptly search performance on the confirmation set. Then, the top CNN planned is chose for the HSI classification model. Furthermore, the author plotted a new regularization method called “cut put,” that aimlessly detaches a definite area from the native image.

Chen et al. predicted a classification model which was worked on the unification of deep learning model and randomly subspace established ensemble learning, and also worked on TL strategy to increase up the learning stage.

Cui et al. hand overed a multiscale spatial–spectral CNN grid to amalgamate multiple accepting field merging features and multiscale spatial features at several levels. The merged feature is enlarged by employing the lightweight chunks of various reception fields, that consist of numerous varieties of dilated convolutions.

Li et al. presented a seed that utilizes spatial– spectral feature studying grids to reflect the interchange in spatial details and to acquire robust compatible features. Instead of relating unconventional spatial features and spectral features, this skeleton merged CNN and SAE to directly withdraw linking of spatial-spectral features from HSIs.

In , a spatial transformation network (STN) was surveyed to obtain the greatest input for CNN- based HSI classification. The initiation of the STN network is handed down to interpret, revolve, and scale the authentic image, and survey advanced inputs for following CNNs. Furthermore, after the ease of overfitting, the regularization mechanism of DropBlock is initiated to get best classification accuracy.

Hamida et al. proposed a 3-D CNN established model, which uses several sizes of 3-D convolution at the same time to operate the spatial and spectral constituent, to train the model with less framework.

He et al. offered a handcrafted feature removal technique established on the multiscale covariance maps, that has a good robust feature and classification presentation.

Li et al. put forwarded a robust 3-D-CapsNet model, that commenced the maximum correntropy criterion to label the noise and outlier problem.

Shi and Pun made a multiscale super pixel established with RNN with SAEs for classifying.

Sun et al. worked for an attention mechanism to establish an end-to-end spectral–spatial attention net- work (SSAN). Along this network, they aimed to capture the particular spectral– spatial attributes from the awareness from the areas of HSI cube.

Chapter 4

DATASET USED

Datasets which are taken with the use of hyperspectral sensor have been made available to the scientific community. They usually come with the annotations to study the classification performances.

Indian Pines: This dataset is taken by the use of AVIRIS sensor.Most of the areas of this image present fields with variety of crops apart from this all the area denotes forest and dense vegetarian.Altogether16 classes are there for example-corn, grass, soybean, woods and many more. The rarest among all are falafel and oats. Before processing water absorption bands are generally removed. It is one of the most important reference datasets of the community despite of its limited size. Though rare classes are generally not taken while checking classification algorithms.

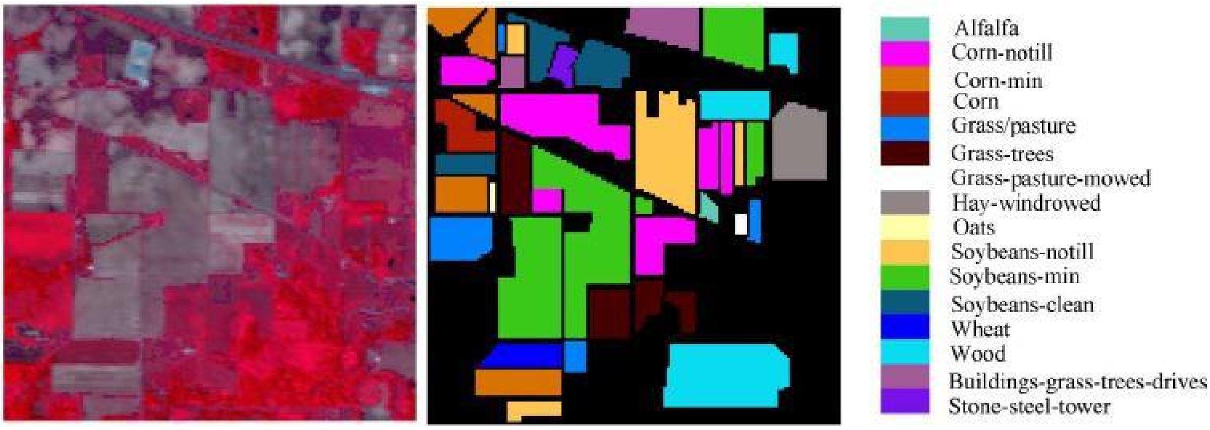
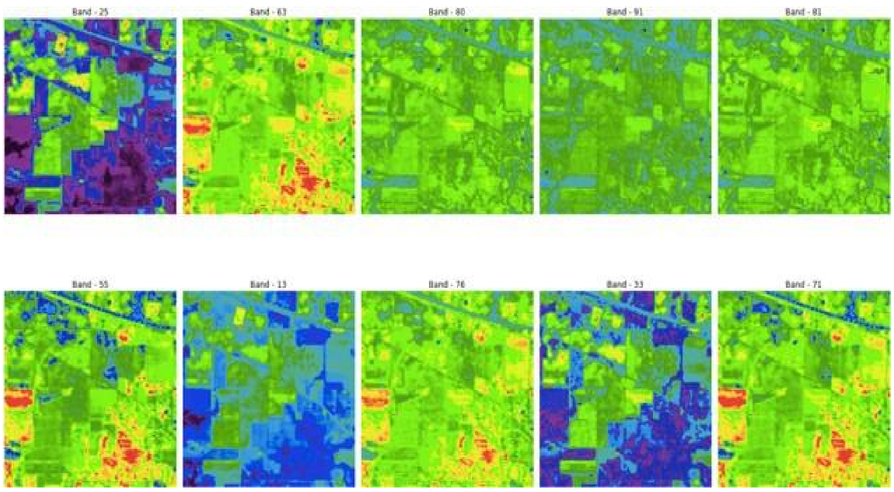


Fig 4. The Indian Pines dataset and the classified classes



i

Figure 5: Data visualization of HSI bands

Dataset	Pixels	Bands	Range	GSD	Labels	Classes	Mode
Pavia (U & C)	991,040	103	0.43-0.85 μm	1.3 m	50,232	9	Aerial
Indian Pines	21,025	224	0.4-2.5 μm	20 m	10,249	16	Aerial
Salinas	111,104	227	0.4-2.5 μm	3.7 m	54,129	16	Aerial
KSC	314,368	176	0.4-2.5 μm	18 m	5,211	13	Aerial
Botswana	377,856	145	0.4-2.5 μm	30 m	3,248	14	Satellite
DFC 2018	5,014,744	48	0.38-1.05 μm	1 m	547,807	20	Aerial

Table 1: Main public labelled datasets in hyperspectral imaging.

METHODS OF DEEP LEARNING USED FOR HYPERSPECTRAL DATA

Current works use deep learning approaches for categorizing hyperspectral images. The evaluation which follows is assembled to spot the main families of procedure.

1. *Pre-processing:*

   Pre-handling techniques utilized for profound learning are similar to the ones used for standard AI. Despite the fact that, it is esteemed taking note of that larger part works don't use band determination or immersed range can dispose of however actually depend on the strength of neural organizations. For outright, another legitimacy, solo procedure, exist in an organization with two stacked auto-encoders: the underlying one is used for denoising while another does the first unmixing by applying a sparsity limitation.

2. *Spatial-spectral approaches:*

   1D or 2D Convolutional Neural Networks: A later strategy grounded in convolutional networks so renowned in blended media vision: Convolutional Neural Networks (CNN). In PC vision, most CNN are wanted to use an underlying section which is convolutional and plays out the part extraction and depiction learning, and a second part which is totally related and plays out the portrayal. Regardless, the quantity of channels is comparative with the amount of data channels, for instance for a first convolutional layer with a section 5x5 and n yield channels, there will be 5 x n x3 for a RGB picture (3 channels) be that as it may, 5x5xnx100 for ordinary hyperspectral pictures (with 100 gatherings). Consequently, a notable method for managing making an interpretation of significant convolutional associations to hyperspectral imaging comprises in reducing the ghost estimation to close the opening among hyperspectral and RGB pictures.

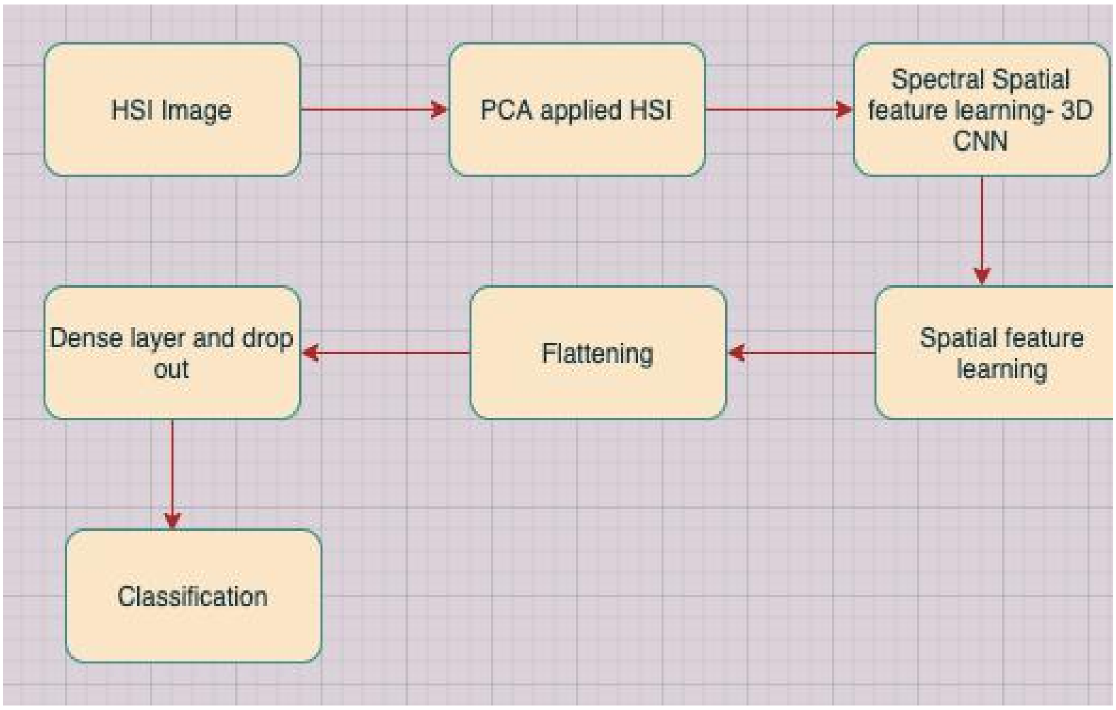


Figure 6: Methodology used in proposed model

The methodology shown in the proposed model above in *figure 4* gives the understanding of hyperspectral data where it is first assessed for PCA ,that is, pre-processing. Then CNN is used on the same dataset (spatial-spectral approach for supervised learning). This method is a combination of 3-D and 2-D model for HSI classification where it basically combines the information of spatial-spectral and spectral in the form of 3-D and 2-D convolutions, respectively.

Later the accuracies are calculated and comparative analysis is produced in this model. Below *Table 1* shows that CNN used for HSI classification is highly accurate compared with SVM and other methods as the overall accuracy obtained is 81.39%.

Table 2: Experimental results of various models from Indian Pines dataset

Methods	Indian Pines **Accuracy**
CNN	**81.39**
SVM	**72.38**
KNN	**66.21**

3. 2D+1D CNN:

Supervised learning:

Without a doubt, the meaning of miraculous information gives rise to new approaches trading with the hyperspectral 3D shape. The inspection is to determine the hypercube using a beginning to end remarkable learning measure. To answer the matter of the weird estimation turn down, a couple of works attempted to design CNN dealing spatial convolutions with terrible ones to reliably reduce the size of the component maps. Clearly, this presents a CNN which examines both the spatial and absurd neighbourhoods of a given pixel, for example that acknowledges a 3D fix as an information. The elementary layers decrease the powerful estimation using a 1*n segment, then, the spatial ones with a k*1 piece, and so forth. Over the prolonged years, two totally linked layers play out the last game plan step. This permits them to build feature maps where both hide besides, spatial pictures are learned on the other hand. In a similar project, suggested an elective ideology that performs spatial-spectral convolutions in the necessary layer to carry out extraordinary estimation turn down, correspondingly to what precisely specific could be an ordinary from PCA, yet controlling and including spatial data. Most important layers structure a traditional 2D CNN that is in return.

Unsupervised learning:

On the production side, presented a 2D CNN with an extra learning point of view that can get to know a need of low-estimation depiction of the hyperspectral pixels and their zone. Finally, another proposes a Fully Convolutional Network (FCN) which controls N gatherings. The necessary layer picks out a multiscale spatial-ridiculous unit using a module roused by the Origin plan. The model appeals in the rough data a couple of convolutions with a growing chunk size in the spatial approximation, for instance 11N, 33N and 33N where N is the amount of group. This decreases the supernatural estimation and be the side plays out a multi-scale isolating of the content. These coming with regards to beginning maps are then dealt with to a movement of nonlinear and 1D convolutions that projects the part of maps in the last course of movement space. It is massive that gratitude to its Fully Convolutional plan, this alliance makes figures for all pixels in the data fix, and in cooperation to the central one. This concludes that it is more capable at induction time while gliding over the whole picture.

4. *3D CNN:*

In the event that spatial-powerful procedures at this point reach particularly satisfying consequence, especially relevant to the spatial consistency, they need a certain level of science in their positioning which isn't totally worthy with the "information to-yield" saying that relates significant learning. A promising procedure handles directly the hyperspectral 3D shape with 3D CNN which works all together on the three estimations using 3D convolutions. This rationally less perplexing technique to some degree to a greater extent fosters the request presentations with reverence to 2D+1D models. Numerous plans have been put forward to manage 3D convolutional neural associations for hyperspectral data, generally to look into prominent techniques from remarkable learning for PC vision, for example, multi-scale feature extraction and semi-administration. Rather than fetching 2D part maps, these 3D CNN produce 3D component strong shapes 7 that are satisfactory for plan affirmation in a volume and have all the earmarks of being somewhat speculatively more significant for hyperspectral picture request. Some researchers particularly showed that 3D CNN for request of hyperspectral pictures obtained more high displays than their 2D accomplices. Point of fact, regarding repulsive or 2D+1D CNN, 3D CNN merges those two model affirmation frameworks into one channel, requiring not so much limits and layers. They can make out how to see more convoluted 3D instances of reflectance: collocated shocking engraves, numerous differentiations of maintenance among groups, etc. Notwithstanding, all labels or titles are not same in the hyperspectral data block, so these models are not ready as straightforward volumetric data. Concerning 2D+1D CNN, it suggests extra thought when arranging the association like using anisotropic channels.

Chapter 5

IMPLEMENTATION

```python
# -*- coding: utf-8 -*-
"""
Created on Sun Aug  8 12:34:02 2021

@author: Sushmita Gautam
"""

import keras
from keras.layers import Conv2D, Conv3D, Flatten, Dense, Reshape, BatchNormalization
from keras.layers import Dropout, Input
from keras.models import Model
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint
from keras.utils import np_utils

from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, cohen_kappa_score

from operator import truediv

import numpy as np
import matplotlib.pyplot as plt
import scipy.io as sio
import spectral

## GLOBAL VARIABLES
test_ratio = 0.7
windowSize = 25

def loadData():
    data = sio.loadmat('C:/Users/Sushmita Gautam/Downloads/indianpines/Indian_pines.mat')['indian_pines']
    labels = sio.loadmat('C:/Users/Sushmita Gautam/Downloads/indianpines/Indian_pines_gt.mat')['indian_pines_gt']
    return data, labels

def splitTrainTestSet(X, y, testRatio, randomState=345):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=testRatio, random_state=randomState,
                                                        stratify=y)
    return X_train, X_test, y_train, y_test
```

```python

def applyPCA(X, numComponents=75):
    newX = np.reshape(X, (-1, X.shape[2]))
    pca = PCA(n_components=numComponents, whiten=True)
    newX = pca.fit_transform(newX)
    newX = np.reshape(newX, (X.shape[0],X.shape[1], numComponents))
    return newX, pca


def padWithZeros(X, margin=2):
    newX = np.zeros((X.shape[0] + 2 * margin, X.shape[1] + 2* margin, X.shape[2]))
    x_offset = margin
    y_offset = margin
    newX[x_offset:X.shape[0] + x_offset, y_offset:X.shape[1] + y_offset, :] = X
    return newX

def createImageCubes(X, y, windowSize=5, removeZeroLabels = True):
    margin = int((windowSize - 1) / 2)
    zeroPaddedX = padWithZeros(X, margin=margin)
    # split patches
    patchesData = np.zeros((X.shape[0] * X.shape[1], windowSize, windowSize, X.shape[2]))
    patchesLabels = np.zeros((X.shape[0] * X.shape[1]))
    patchIndex = 0
    for r in range(margin, zeroPaddedX.shape[0] - margin):
        for c in range(margin, zeroPaddedX.shape[1] - margin):
            patch = zeroPaddedX[r - margin:r + margin + 1, c - margin:c + margin + 1]
            patchesData[patchIndex, :, :, :] = patch
            patchesLabels[patchIndex] = y[r-margin, c-margin]
            patchIndex = patchIndex + 1
    if removeZeroLabels:
        patchesData = patchesData[patchesLabels>0,:,:,:]
        patchesLabels = patchesLabels[patchesLabels>0]
        patchesLabels -= 1
    return patchesData, patchesLabels


X, y = loadData()

X.shape, y.shape
```

```


X, y = loadData()

X.shape, y.shape

K = X.shape[2]

K = 30
X,pca = applyPCA(X,numComponents=K)
X.shape

X, y = createImageCubes(X, y, windowSize=windowSize)
X.shape, y.shape

Xtrain, Xtest, ytrain, ytest = splitTrainTestSet(X, y, test_ratio)
Xtrain.shape, Xtest.shape, ytrain.shape, ytest.shape

#model and training

Xtrain = Xtrain.reshape(-1, windowSize, windowSize, K, 1)
Xtrain.shape


ytrain = np_utils.to_categorical(ytrain)
ytrain.shape

S = windowSize
L = K
output_units = 16

## input layer
input_layer = Input((S, S, L, 1))

## convolutional layers
conv_layer1 = Conv3D(filters=8, kernel_size=(3, 3, 7), activation='relu')(input_layer)
conv_layer2 = Conv3D(filters=16, kernel_size=(3, 3, 5), activation='relu')(conv_layer1)
conv_layer3 = Conv3D(filters=32, kernel_size=(3, 3, 3), activation='relu')(conv_layer2)
print(conv_layer3.shape)
conv3d_shape = conv_layer3.shape
conv_layer3 = Reshape((conv3d_shape[1], conv3d_shape[2], conv3d_shape[3]*conv3d_shape[4]))(conv_layer3)
```

```
print(conv_layer3.shape)
conv3d_shape = conv_layer3.shape
conv_layer3 = Reshape((conv3d_shape[1], conv3d_shape[2], conv3d_shape[3]*conv3d_shape[4]))(conv_layer3)
conv_layer4 = Conv2D(filters=64, kernel_size=(3,3), activation='relu')(conv_layer3)

flatten_layer = Flatten()(conv_layer4)

## fully connected layers
dense_layer1 = Dense(units=256, activation='relu')(flatten_layer)
dense_layer1 = Dropout(0.4)(dense_layer1)
dense_layer2 = Dense(units=128, activation='relu')(dense_layer1)
dense_layer2 = Dropout(0.4)(dense_layer2)
output_layer = Dense(units=output_units, activation='softmax')(dense_layer2)

# define the model with input layer and output layer
model = Model(inputs=input_layer, outputs=output_layer)

model.summary()

# compiling the model
adam = Adam(lr=0.001, decay=1e-06)
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])


# checkpoint
filepath = "best-model.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='acc', verbose=1, save_best_only=True, mode='max')
callbacks_list = [checkpoint]

history = model.fit(x=Xtrain, y=ytrain, batch_size=256, epochs=100, callbacks=callbacks_list)

#validation

# load best weights
model.load_weights("best-model.hdf5")
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])

Xtest = Xtest.reshape(-1, windowSize, windowSize, K, 1)
Xtest.shape
ytest = np_utils.to_categorical(ytest)
ytest.shape
```

CNN_SUSHMITA_HYPERSPECTRAL.py*

```
ytest = np_utils.to_categorical(ytest)
ytest.shape


Y_pred_test = model.predict(Xtest)
y_pred_test = np.argmax(Y_pred_test, axis=1)

classification = classification_report(np.argmax(ytest, axis=1), y_pred_test)
print(classification)


def AA_andEachClassAccuracy(confusion_matrix):
    counter = confusion_matrix.shape[0]
    list_diag = np.diag(confusion_matrix)
    list_raw_sum = np.sum(confusion_matrix, axis=1)
    each_acc = np.nan_to_num(truediv(list_diag, list_raw_sum))
    average_acc = np.mean(each_acc)
    return each_acc, average_acc

def reports (X_test,y_test,name):
    #start = time.time()
    Y_pred = model.predict(X_test)
    y_pred = np.argmax(Y_pred, axis=1)
    #end = time.time()
    #print(end - start)
    if name == 'IP':
        target_names = ['Alfalfa', 'Corn-notill', 'Corn-mintill', 'Corn'
                        ,'Grass-pasture', 'Grass-trees', 'Grass-pasture-mowed',
                        'Hay-windrowed', 'Oats', 'Soybean-notill', 'Soybean-mintill',
                        'Soybean-clean', 'Wheat', 'Woods', 'Buildings-Grass-Trees-Drives',
                        'Stone-Steel-Towers']
    elif name == 'SA':
        target_names = ['Brocoli_green_weeds_1','Brocoli_green_weeds_2','Fallow','Fallow_rough_plow','Fallow_smooth',
                        'Stubble','Celery','Grapes_untrained','Soil_vinyard_develop','Corn_senesced_green_weeds',
                        'Lettuce_romaine_4wk','Lettuce_romaine_5wk','Lettuce_romaine_6wk','Lettuce_romaine_7wk',
                        'Vinyard_untrained','Vinyard_vertical_trellis']
    elif name == 'PU':
        target_names = ['Asphalt','Meadows','Gravel','Trees', 'Painted metal sheets','Bare Soil','Bitumen',
                        'Self-Blocking Bricks','Shadows']

    classification = classification_report(np.argmax(y_test, axis=1), y_pred, target_names=target_names)
```

```

    classification = classification_report(np.argmax(y_test, axis=1), y_pred, target_names=target_names)
    oa = accuracy_score(np.argmax(y_test, axis=1), y_pred)
    confusion = confusion_matrix(np.argmax(y_test, axis=1), y_pred)
    each_acc, aa = AA_andEachClassAccuracy(confusion)
    kappa = cohen_kappa_score(np.argmax(y_test, axis=1), y_pred)
    score = model.evaluate(X_test, y_test, batch_size=32)
    Test_Loss =  score[0]*100
    Test_accuracy = score[1]*100

    return classification, confusion, Test_Loss, Test_accuracy, oa*100, each_acc*100, aa*100, kappa*100


classification, confusion, Test_loss, Test_accuracy, oa, each_acc, aa, kappa = reports(Xtest,ytest)
classification = str(classification)
confusion = str(confusion)
file_name = "classification_report.txt"

with open(file_name, 'w') as x_file:
    x_file.write('{} Test loss (%)'.format(Test_loss))
    x_file.write('\n')
    x_file.write('{} Test accuracy (%)'.format(Test_accuracy))
    x_file.write('\n')
    x_file.write('\n')
    x_file.write('{} Kappa accuracy (%)'.format(kappa))
    x_file.write('\n')
    x_file.write('{} Overall accuracy (%)'.format(oa))
    x_file.write('\n')
    x_file.write('{} Average accuracy (%)'.format(aa))
    x_file.write('\n')
    x_file.write('\n')
    x_file.write('{}'.format(classification))
    x_file.write('\n')
    x_file.write('{}'.format(confusion))

# load the original image
X, y = loadData()

height = y.shape[0]
width = y.shape[1]
PATCH_SIZE = windowSize
```

```
height = y.shape[0]
width = y.shape[1]
PATCH_SIZE = windowSize
numComponents = K

def Patch(data,height_index,width_index):
    height_slice = slice(height_index, height_index+PATCH_SIZE)
    width_slice = slice(width_index, width_index+PATCH_SIZE)
    patch = data[height_slice, width_slice, :]

    return patch

X,pca = applyPCA(X, numComponents=numComponents)

X = padWithZeros(X, PATCH_SIZE//2)

# calculate the predicted image
outputs = np.zeros((height,width))
for i in range(height):
    for j in range(width):
        target = int(y[i,j])
        if target == 0 :
            continue
        else :
            image_patch=Patch(X,i,j)
            X_test_image = image_patch.reshape(1,image_patch.shape[0],image_patch.shape[1], image_patch.shape[2], 1).astype('float32')
            prediction = (model.predict(X_test_image))
            prediction = np.argmax(prediction, axis=1)
            outputs[i][j] = prediction+1



ground_truth = spectral.imshow(classes = y,figsize =(7,7))

predict_image = spectral.imshow(classes = outputs.astype(int),figsize =(7,7))

spectral.save_rgb("predictions.jpg", outputs.astype(int), colors=spectral.spy_colors)
```

Chapter 6

RESULTS AND CONCLUSION

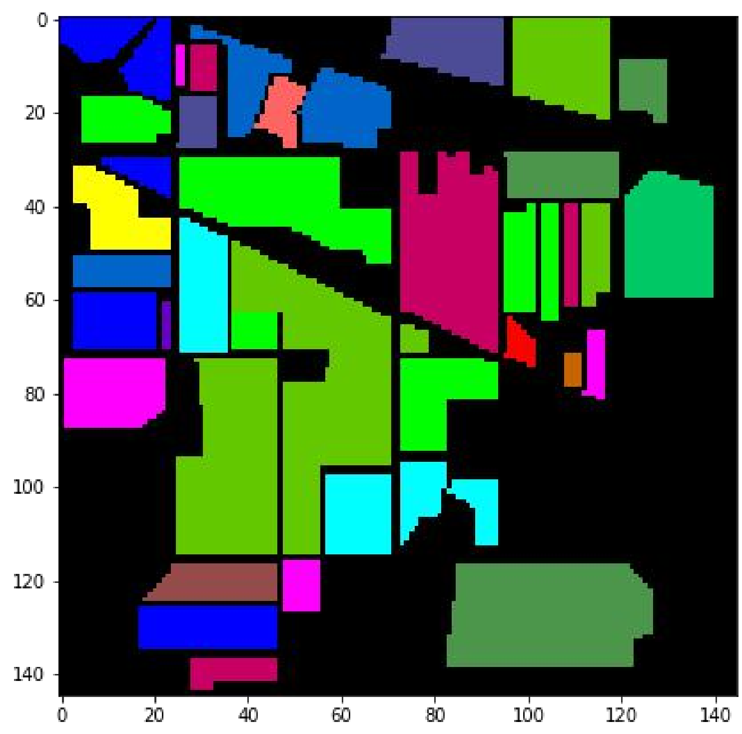
```
model.summary()

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 25, 25, 30, 1)     0
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 23, 23, 24, 8)     512
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 21, 21, 20, 16)    5776
_________________________________________________________________
conv3d_3 (Conv3D)            (None, 19, 19, 18, 32)    13856
_________________________________________________________________
reshape_1 (Reshape)          (None, 19, 19, 576)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 17, 17, 64)        331840
_________________________________________________________________
flatten_1 (Flatten)          (None, 18496)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               4735232
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 16)                2064
=================================================================
Total params: 5,122,176
Trainable params: 5,122,176
Non-trainable params: 0
```

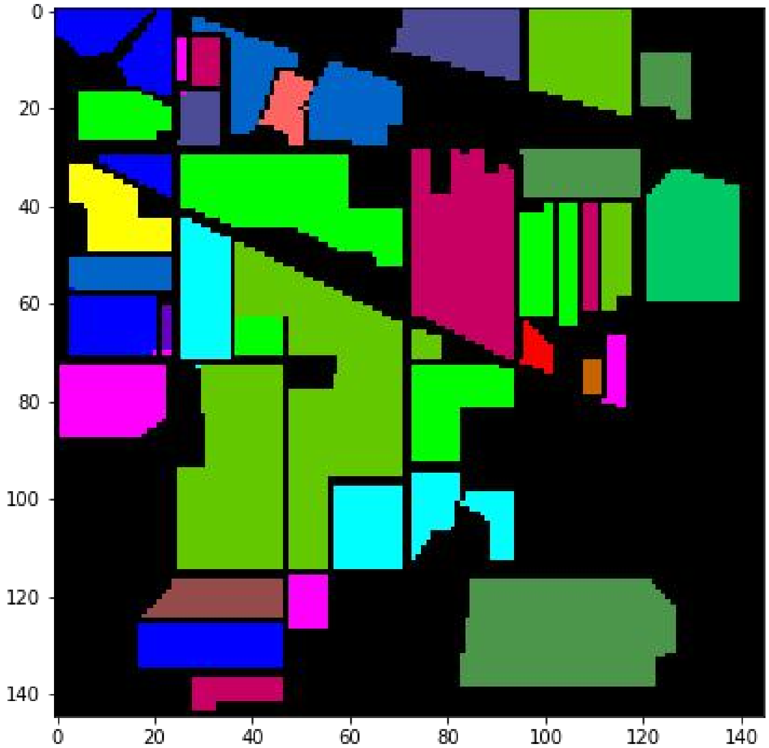
```
print(classification)
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        32
           1       1.00      1.00      1.00      1000
           2       1.00      1.00      1.00       581
           3       1.00      1.00      1.00       166
           4       0.99      1.00      0.99       338
           5       1.00      1.00      1.00       511
           6       1.00      1.00      1.00        20
           7       1.00      1.00      1.00       335
           8       1.00      0.86      0.92        14
           9       1.00      1.00      1.00       680
          10       1.00      1.00      1.00      1719
          11       1.00      1.00      1.00       415
          12       1.00      1.00      1.00       143
          13       1.00      1.00      1.00       886
          14       1.00      1.00      1.00       270
          15       0.98      1.00      0.99        65

   micro avg       1.00      1.00      1.00      7175
   macro avg       1.00      0.99      0.99      7175
weighted avg       1.00      1.00      1.00      7175
```



Ground Truth Image.



Predicted Image.

Conclusion

This method is a combination of 3-D and 2-D model for HSI classification. This model basically combines the information of spatial-spectral and spectral in the form of 3-D and 2-D convolutions, respectively. This method for HSI classification is highly accurate compared with SVM and other methods as the overall accuracy obtained is 99.7%.

Reference

[1]Y. Yu, Z. Gong, C. Wang, and P. Zhong, “An unsupervised convolutional feature f usion network for deep representation of remote sensing images,” IEEE Geosci. Remote Sens. Lett., vol. 15, no. 1, pp. 23–27, Jan. 2018.

[2]W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, “Data augmentation for hyperspectral image classification with deep CNN,” IEEE Geosci. Remote Sens. Lett., vol. 16, no. 4, pp. 593–597, Apr. 2019.

[3]W. Song, S. Li, L. Fang, and T. Lu, “Hyperspectral image classification with deep feature fusion network,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 6, pp. 3173 –3184, Jun. 2018.

[4] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, “Exploring hierarchical convolutional features for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[5] J. M. Haut, S. Bernabé, M. E. Paoletti, R. Fernandez-Beltran, A. Plaza, and J. Plaza, “Low–high-power consumption architectures for deeplearning models applied to hyperspectral image classification,” IEEE Geosci. Remote Sens. Lett., vol. 16, no. 5, pp. 776–780, May 2019.

[6] 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 Trans. Geosci. Remote Sens., vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[7] Z. Zhong, J. Li, Z. Luo, and M. Chapman, “Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework,” IEEE Trans.

[8]L. Mou, P. Ghamisi, and X. X. Zhu, “Unsupervised spectral–spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 1, pp. 391–406, Jan. 2018