Hybrid CNN-LSTM model for classification of multispectral satellite images

Anumit Garg
Department of Electronics and Communication Engineering
Dr. B.R. Ambedkar National Institute of Technology
Jalandhar, India
anumitgarg@gmail.com

Anil Kumar
Department of Photogrammetry and Remote Sensing
Indian Institute of Remote Sensing (ISRO)
Dehradun, India
anil@iirs.gov.in

***Abstract*—The advent of recent technological breakthroughs in deep learning have introduced some of the state of the art algorithms and techniques for processing and gaining insights from the data. These techniques played an important role in improving existing techniques while replacing many obsolete and computationally inefficient algorithms. Image classification is an important task which finds its application in various fields. In this paper we have used contemporary deep learning techniques to build a model for the classification of multispectral satellite images. In this work two of the most successful principles of deep learning have been combined, namely convolution neural networks and recurrent neural networks for classification task. In this paper hybrid CNN-LSTM architecture for the classification task has been proposed. The model achieved a staggering accuracy of 94.8 percent on 6 class classification task. Moreover this work is a testament of the fact that Neural Networks could be robust classifiers for processing multispectral data, even outperforming traditional learning techniques**

Keywords—multispectral satellite images, remote sensing, multi label classification, recurrent neural networks (RNN), long short term memory (LSTM), convolution neural network (CNN), traditional learning techniques.

I. Introduction

Remote sensing refers to the acquisition and study of information about the properties of specified objects or physical phenomena on the earth without making actual physical contact with the area under supervision. Remote sensing employs use of satellites, aircrafts, balloons or ground based sensors for data acquisition. The key principles behind remote sensing in the characteristic property of the objects to absorb, reflect, transmit and emit electromagnetic radiations from the sun. Every material has a characteristic spectral signature, which is the variation of reflectance or emittance with respect to wavelength. The plot of these variations are called spectral response curve and are of key importance in the identification of particular landmark.

In recent years remote sensing has observed considerable developments due to the easy availability of rich spectral and spatial information. Remote sensing find its application in various domains such as [1] [2] [3] [4]; resource exploration, disaster management, oil spill detection, thematic map creation, weather forecasting, environmental study etc.

A digital image can be defined as an arrangement of individual pixels in a 3 dimensional array comprising of rows, columns and bands as shown in Fig 1. Here each band corresponds to the range of wavelengths (colors) in the electromagnetic spectrum. The value of each element of the array is the intensity of pixel corresponding to the band in which it lies. Images acquired from satellites are rich in information, and plays a strategic role in providing geographical information [5]. These images provide quantitative and qualitative information that considerably reduces the field work and study time [6]. Therefore satellite image classification is a very powerful technique to extract information from satellite data.

Satellite image classification is a process of classifying or grouping pixels to meaningful classes [7]. Image classification is multi-step process that has application in multiple fields**.** There are various methods and techniques for image classification which can broadly be classified into three categories [8] namely; automated, manual and hybrid classification algorithms.

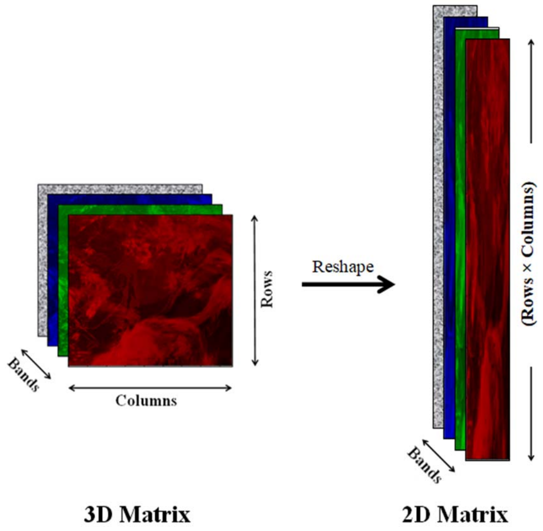

Fig. 1. Representation of a digital image

In this work we propose an automated supervised learning algorithm using deep learning techniques for the classification of satellite images.

II. Related Works

F Wang et al. [9] proposed a fuzzy supervised classification method. Since its inception it is one of the most widely used algorithms in the field of image classification and remote sensing. In this method the geographical information is represented as fuzzy sets. The algorithm consists of two major steps: the estimation of fuzzy parameters from fuzzy training data, and a fuzzy partition of spectral space. Partial membership of pixels allows component cover classes of mixed pixels to be identified and more accurate statistical parameters to be generated, resulting in higher classification accuracy.

F. Melgani et al. [10] addressed the classification problem of hyperspectral images by support vector machines (SVMs). Their work was aimed at assessing the properties of SVMs in hyperspectral spaces and hypersubspaces of various dimensionalities. They used various performance indicators to access their proposition which includes the classification accuracy, the computational time, the stability to parameter setting, and the complexity of the multiclass architecture. Their results support the fact that SVMs are much more efficient than other conventional non-parametric classifiers, and SVMs can be a suitable alternative of traditional pattern recognition approaches for the classification for remote sensing data.

The work of L. Zhang et al. [11] proposes a patch alignment framework to linearly combine multiple features in the optimal way to obtain a unified low dimensional representation of multiple features for the ease of classification. According to their work in image classification each feature (which includes spectral, texture and shape features) has its particular contribution to the unified representation determined by simultaneously optimizing the weights in the objective function. This scheme considers the specific statistical properties of each feature to achieve a physically meaningful unified low-dimensional representation of multiple features. Experiments on the classification of the hyperspectral digital imagery collection experiment and reflective optics system imaging spectrometer hyperspectral data sets proves the effectiveness of their scheme.

In recent years due to advancement in machine learning and its success in various domains, many researchers have shifted their interests from traditional statistical approaches to deep learning approaches to tackle the problems. Similarly, the work of N. Kussul et al. [12] describes a multilevel deep learning architecture that targets land cover and crop type classification from multi temporal and multi source satellite imagery. An unsupervised neural network forms the basis of their algorithm. It is used for optical image segmentation and missing data restoration due to clouds and shadows. However according to them an ensemble of CNNs outperformed their MLPs allowing a better discrimination between certain summer crop types. This clearly demonstrates the efficiency of neural networks and superiority on CNNs for classification tasks in remote sensing.

Convolution neural networks have proved their mettle time and again for image classification tasks. In their work M. Castelluccio et al. [13] proposes a convolution neural network for the semantic classification of remote sensing scenes. They adopted two state of the art architectures, CaffeNet and GoogLeNet with three different learning models. Besides conventional training from scratch they also adopted transfer learning to fine tune the pre-trained networks for their task. Their results once again testify the effectiveness and wide applicability of their CNN based proposed solution.

Some of the latest works in the field of remote sensing and satellite image processing can be attributes to the much recent advancements in deep learning which has provided much impetus to a more focused research in this domain. The work of [14] proposes two deep RNN approaches to explicitly consider the temporal correlation of Sentinel-1 data; their work also suggests that RNN yields consistent improvements in agricultural classes as compared to classical machine learning approaches. [15] describes a deep learning system for classifying objects and facilities from the IARPA Functional Map of the World (fMoW) dataset into 63 different classes and achieved an overall accuracy of 83%.

III. Study Area and dataset used

In our work we have used the site location in Haridwar district of Uttarakhand state in India. The site is diverse in terms of land cover features and various classes for testing the algorithm can be easily identified from the image. The dataset is available in the form of Formosat-2 satellite image which consists of 4 bands at a spatial resolution of 8 meters.

The specifications which include spatial resolution, spectral resolution, sensor footprint and return interval are stated in Table I.

A sample formosat-2 satellite image has been shown in Fig 2. The color of the image depends on the type of light satellite instrument measures. This is a false color image that incorporates incorporate infrared light and thus may have unexpected colors.

Table I. Formosat-2 Sensor Details

Specifications	**Formosat- 2**
Spatial Resolution	8 m
Spectral Resolution	B1: (0.45 – 0.52) µm B2: (0.52 – 0.60) µm B3: (0.63 – 0.69) µm B4: (0.76 – 0.90) µm
Sensor Footprint	(24×24) km^2
Return Interval	24 hours

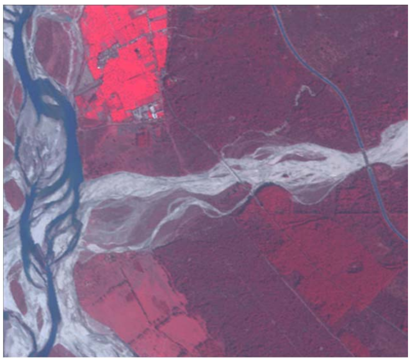

Fig. 2. False color Formosat-2 input image using near infrared, red and green spectral bands mapped to RGB

IV. Methodology

The main objective of the study is to develop a neural network that can robustly classify individual pixels of a multispectral image to one or more of the six feature labels with high accuracy.

In our work we have used contemporary deep learning techniques and developed a Hybrid CNN- LSTM model for the classification of Multispectral Images. The model uses ReLu activation function to include non linearity, softmax activation function for the prediction of probability of occurrence, CNN and LSTM layers due to their flexibility for vision tasks involving sequential inputs and Maxpooling layer for dimensionality reduction. The model architecture is implemented using Keras and Tensorflow libraries in Python.

A. Convolution Neural Network Basics

A CNN finds its application mainly for identifying simple patters within the data which can be further used to identify more complex patterns within higher layers. A 1D CNN is very effective for deriving features from shorter sequences of overall data set and where location of the feature within the segment is not of very high relevance.

All form of CNNs no matter if they are 1D, 2D or 3D follow the same approach. The key difference lies in the dimensions of input data and the filter used for convolution. Fig. 3. shows the implementation of 1D convolution on a dataset. Here x1, x2, x(n) represents the pixel values of reshaped image; w1, w2 and w3 are the weight value of the filter. Upon convolution an output sequence a1, a2, a(n) is generated by convolution operation as shown in equations (1-6) below.

$$a1 = (w1 \times x1) + (w2 \times x2) + (w3 \times x3) \quad (1)$$

$$a2 = (w1 \times x2) + (w2 \times x3) + (w3 \times x4) \quad (2)$$

$$a3 = (w1 \times x3) + (w2 \times x4) + (w3 \times x5) \quad (3)$$

$$a4 = (w1 \times x4) + (w2 \times x5) + (w3 \times x6) \quad (4)$$

$$a5 = (w1 \times x5) + (w2 \times x6) + (w3 \times x7) \quad (5)$$

$$a6 = (w1 \times x6) + (w2 \times x7) + (w3 \times x8) \quad (6)$$

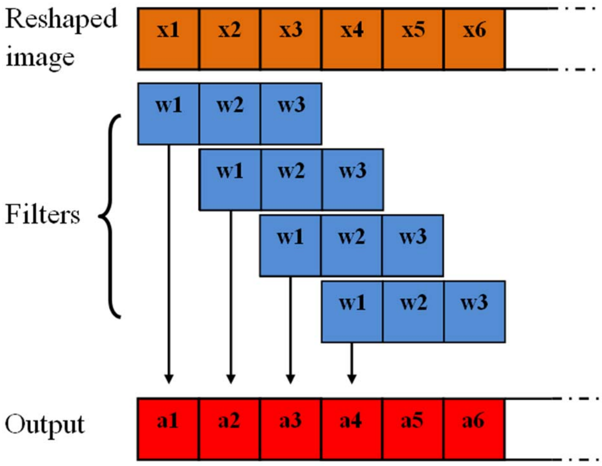

Fig. 3. 1D Convolution

B. Recurrent Neural Network and LSTM basics

RNN effectively enables us to remember short sequences and model sequential data. However they do suffer from one of the major problems of vanishing gradient [16]. In this problem during back propagation the gradient reduces to such a small value that that no parameters are significantly updated, thus restricting RNN to learn effectively from the data.

There are number of measures to overcome this problem. For our work, we have used LSTMs [17] to rectify the problem of vanishing gradient and also enable our model to learn long sequences. The distinguishing feature of LSTMs that makes them more appropriate choice over RNNs is its internal architecture shown in Fig. 4. There are three gates which provide LSTM its functionality.

Here, $c^{<t>}$ represents the cell state, $a^{<t>}$represents output from the block while $c'^{<t>}$ represents candidate for cell state at timestamp (t). Now, to obtain the memory vector for current time stamp the candidate is calculated as given in equation (7).

1) Forget gate: it is responsible for the removal of information from the cell state that is no longer needed for the LSTM to understand and learn from the data. As shown in equation (8), the gate takes two input, input ($x^{<t>}$) at time stamp t and the activation from previous state ($a^{<t-1>}$). These set of input states are multiplied to the weight matrix (w_f) and a bias term (b_f) is added to it, then the resultant is passed through sigmoid function. The output of sigmoid function decides which value to keep and which value to discard; if the output is ‘0’ the forget gate no longer remembers the past value while for output equals ‘1’, the past value is retained.

2) Update gate: it is primarily responsible for the addition of information to the cell state. As shown in equation (9), the gate takes two input, input ($x^{<t>}$) at time stamp t and the activation from previous state ($a^{<t-1>}$). These set of input states are multiplied to the weight matrix (w_u) and a bias term (b_u) is added to it, then the resultant is passed through sigmoid function.

Based on the output of sigmoid function, the gate ensures that only that information is added which is of significant importance in the learning process.

3) Output gate: it is responsible for filtering the information that is not required to be output by the LSTM cell. As shown in equation (10), the gate takes two input, input ($x^{<t>}$) at time stamp t and the activation from previous state ($a^{<t-1>}$). These set of input states are multiplied to the weight matrix (w_o) and a bias term (b_o) is added to it, then the resultant is passed through sigmoid function. The output gate’s job is to select only useful information from current cell and propagate it as an output to the next cell.

Once the value of all the gates are calculated final cell state and activation for the next cell are predicted using these values as shown in equation (11) and (12).

$$c'^{<t>} = \tanh(w_c [a^{<t-1>}, x^{<t>}] + b_c) \quad (7)$$

$$\Gamma_f = \sigma(w_f [a^{<t-1>}, x^{<t>}] + b_f) \quad (8)$$

$$\Gamma_u = \sigma(w_u [a^{<t-1>}, x^{<t>}] + b_u) \quad (9)$$

$$\Gamma_o = \sigma(w_o [a^{<t-1>}, x^{<t>}] + b_o) \quad (10)$$

$$c^{<t>} = \Gamma_u * c'^{<t>} + \Gamma_f * c^{<t-1>} \quad (11)$$

$$a^{<t>} = \Gamma_o * \tanh(c^{<t>}) \quad (12)$$

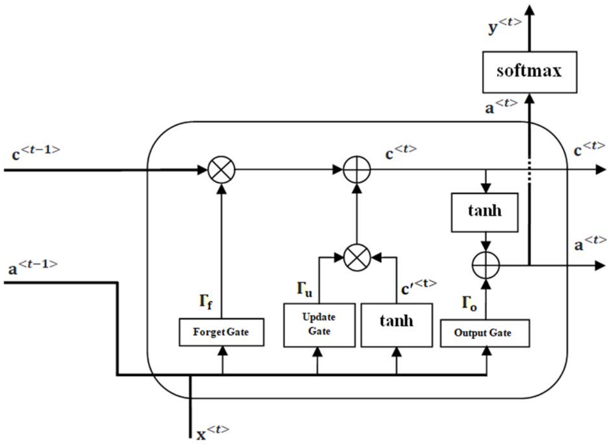

Fig. 4. LSTM cell

V. Proposed Algorithm

The above mentioned CNN and LSTM architectures have contributed to some of the most successful deep learning models for various problems. However each of them has an inherent drawback which fails to exploit the wide multitude of information encoded in formosat – 2 satellite images.

CNNs have the ability to extract useful features from input sequence of pixels; however they lack the ability to deduce correlations among different classes of the image. For example, a CNN can robustly classify a patch of land as “water body” but knows very little about the probability of occurrence of other classes such as “forests”, “deserts” etc. near the classified patch of land. On the other hand LSTMs are very well suited for gaining insights from the sequences and finding patters in the data, however unlike CNNs, they are not able to extract enough features from input data necessary for predictions.

An algorithm that combines efficacies of both CNN and LSTM has been proposed in literature [18] [19] [20] [21] for gesture recognition, voice recognition, rainfall prediction, machine health condition prediction, text analysis, and other fields as well. Therefore in our experiment we have adopted a hybrid approach which incorporates CNN for spatial feature extraction and LSTM layers for gaining insights about adjacency and interdependency of various classes for the classification task. A naïve approach might be to construct 6 different models trained for predicting area corresponding to a single class but the correlative relationship among different labels proves the robustness of a single multi- label classification network over many binary classifiers.

For our model, we converted our 3D image into a 2D array consisting of bands and pixels arranged in a single row as shown in Fig. 1, so that our model can learn to classify pixels effectively. For each experiment we have pixel values of dimension (4×1), which is fed to our model in batches of 32 pixels at a time. Our first 1D convolution layer takes this 2D array as input and performs the convolution operation using 64 filters with a kernel size of 3, the layer uses ReLu activation function. The next layer is an LSTM layer consisting of 64 cells. The output of convolution layer acts as an input for LSTM layer, further the output of LSTM is fed to a maxpooling layer for dimensionality reduction and increasing the computational efficiency of our algorithm.

The output from maxpooling layer follows a similar pattern of layers constituting a convolution layer consisting of 128 filters and kernel size being 4, the output from this layer is further fed to the LSTM layer consisting of 128 cells. The output from this layer is again fed to a maxpooling layer followed by a similar pattern of convolution layer and LSTM layer consisting of 256 filters and cells respectively. At last to predict the correct label to a pixel it is fed to a dense layer consisting of six neurons. The last layer uses softmax activation function to predict the probability of occurrences of each class. The model is detailed in Algorithm 1 and Fig. 5.

We used 100 epochs and trained our model using categorical crossentropy loss function and RMSProp [22] as the optimizer. To reach this final model configuration we experimented with various combinations of activation functions, layers size, filter size and other hyper-parameters and were also limited by hardware constraints to try much bigger networks. Our model performs fairly well on the dataset and proves the robustness of hybrid models for machine learning tasks.

```
Algorithm 1 : Pseudo code for Hybrid CNN- LSTM Model
Requirement : Training Dataset: features x_train, labels y_train
              Testing Dataset: features x_test, labels y_test
Procedure:
    NeuralModel(x_train, y_train)

    batchSize =  32; inputDim = (4, 1); n_classes =6; epochs = 100

    model = Sequential()

    model.add(Conv1D(64, 3, activation='relu', padding='same', input_shape=inputDim))
    model.add(LSTM(64, return_sequences=True))
    model.add(MaxPooling1D(2))
    model.add(Conv1D(128, 4, activation="relu", padding='same'))
    model.add(LSTM(128, return_sequences=True))
    model.add(MaxPooling1D(2))
    model.add(Conv1D(256, 3, activation="relu", padding='same'))
    model.add(LSTM(256))
    model.add(Dense(n_classes, activation='softmax'))

    model.compile(loss='categorical_crossentropy',  optimizer='rmsprop',
                  metrics=['accuracy'])
    model.fit(x_train, y_train, batch_size=batchSize, epochs=epoch)
    score = model.evaluate(x_test, y_test, batch_size=32)
    accuracy = score[1]
    return accuracy

end procedure
```

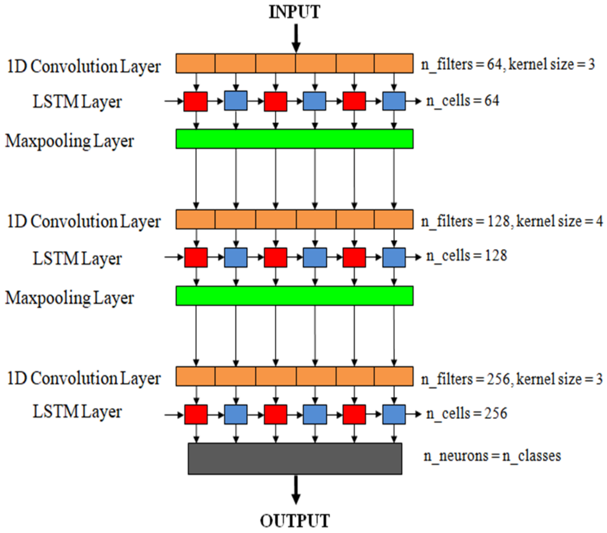

Fig.5. Hybrid CNN-LSTM Model

VI. Results and Discussions

We trained the model on our training data till the training loss reduces and attains an almost stagnant value as shown in Fig. 7. Approximately 500 labeled pixel values for all four bands were split into training and validation sets to train and evaluate the model. As each class has approximately equal number data points acquired from wide range of terrain, thus the dataset is fairly balanced in its composition.

To evaluate our results we use accuracy as the metric, equation (13) defines accuracy as the ratio of correct predictions to the total number of predictions made by the model.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}} \tag{13}$$

Our Hybrid CNN- LSTM model gives us a staggering overall accuracy of 94.8% on the test data. The normalized confusion matrix obtained on the dataset is shown in Fig. 6. The classification accuracies of individual classes are also shown in Table II. The accuracy measure suggests the fact that our model outperformed some of the recent works [15] [23] who achieved an overall accuracy of 83% and 90.36% respectively in their classification tasks. Further the model is used to classify the pixels of the image to their corresponding classes. The result of classification is shown in Fig 8.

Table II. Class-wise Accuracy Measure

Class	Accuracy (in percentage)
Class-1 (Clay)	98%
Class-2 (Eucalyptus)	72%
Class-3 (Forest)	98%
Class-4 (Grassland)	100%
Class-5 (Water)	97%
Class-6 (Wheat patch)	100%

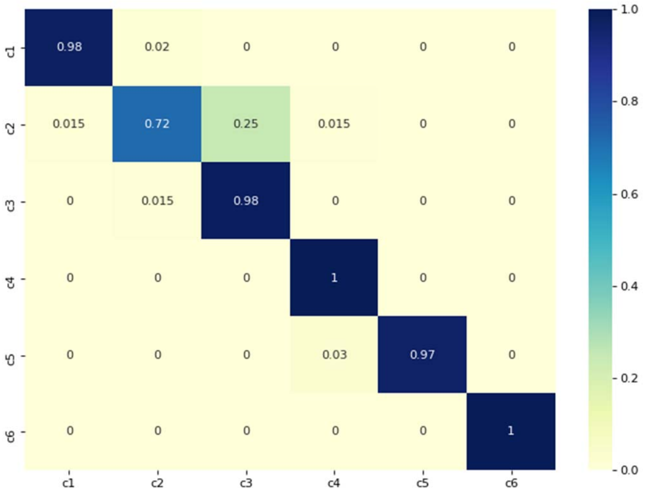

Fig.6. Normalised Confusion Matrix for six classes

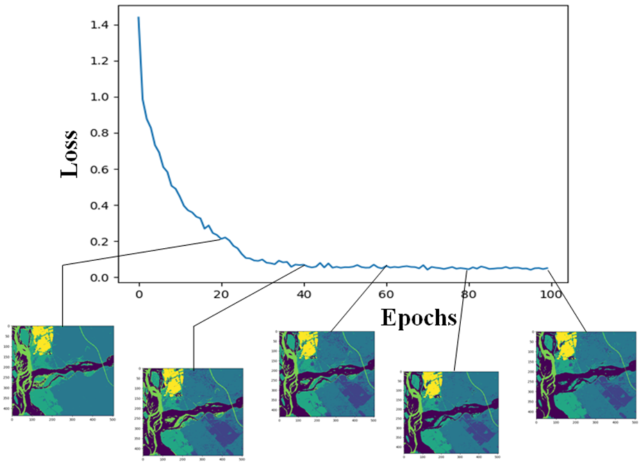

Fig. 7. Training Loss

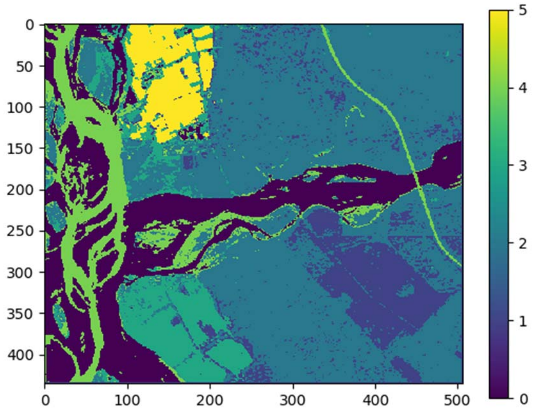

Fig. 8. Classified Image

VII. Conclusion

In this paper we have built upon prior research in the field of image processing using deep learning techniques. We have studies some of the most successful classifiers built using convolution neural networks and implemented a model using a hybrid approach. Our study provides the state of the art classification accuracy on the multi-label classification problem under study. The results of our algorithm were validated using formosat-2 data of multispectral satellite images, we achieved an accuracy of 94.8% in classification of the pixels in six classes. Our classification model uses contemporary deep learning techniques such as CNNs, LSTMs, Maxpooling, ReLu and softmax activation functions to achieve the desired results. This work is testament of the fact that CNN and LSTM forms a robust set of classifiers for remote sensing applications.

References

[1] Seelan, Santhosh K., et al. "Remote sensing applications for precision agriculture: A learning community approach." Remote Sensing of Environment 88.1-2 (2003): 157-169.

[2] Jensen, John R., and Kalmesh Lulla. "Introductory digital image processing: a remote sensing perspective." (1987): 65-65.

[3] Brekke, Camilla, and Anne HS Solberg. "Oil spill detection by satellite remote sensing." Remote sensing of environment95.1 (2005): 1-13.

[4] Navalgund, Ranganath R., V. Jayaraman, and P. S. Roy. "Remote sensing applications: An overview." Current Science (00113891) 93.12 (2007).

[5] Shahbaz, Muhammad, et al. "Classification by Object Recognition in Satellite images by using Data mining." Proceedings of the World Congress on Engineering. Vol. 1. 2012.

[6] Vaiphasa, Chaichoke, et al. "A Normalized Difference Vegetation index (NDVI) Time-series of idle agriculture lands: A preliminary study." Engineering journal 15.1 (2011): 9-16.

[7] Anders Karlsson, 2003. “Classification of high resolution satellite images”, August 2003, available at http://infoscience.epfl.ch/record/63248/files/TPD_Karlss on.pdf

[8] Abburu, Sunitha, and Suresh Babu Golla. "Satellite image classification methods and techniques: A review." International journal of computer applications 119.8 (2015).

[9] Wang, Fangju. "Fuzzy supervised classification of remote sensing images." IEEE Transactions on geoscience and remote sensing 28.2 (1990): 194-201.

[10] Melgani, Farid, and Lorenzo Bruzzone. "Classification of hyperspectral remote sensing images with support vector machines." IEEE Transactions on geoscience and remote sensing 42.8 (2004): 1778-1790.

[11] Zhang, Lefei, et al. "On combining multiple features for hyperspectral remote sensing image classification." IEEE Transactions on Geoscience and Remote Sensing 50.3 (2011): 879-893.

[12] Kussul, Nataliia, et al. "Deep learning classification of land cover and crop types using remote sensing data." IEEE Geoscience and Remote Sensing Letters 14.5 (2017): 778-782.

[13] Castelluccio, Marco, et al. "Land use classification in remote sensing images by convolutional neural networks." arXiv preprint arXiv:1508.00092 (2015).

[14] Ndikumana, Emile, et al. "Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France." Remote Sensing 10.8 (2018): 1217.

[15] Pritt, Mark, and Gary Chern. "Satellite image classification with deep learning." 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR). IEEE, 2017.

[16] Hochreiter, Sepp. "The vanishing gradient problem during learning recurrent neural nets and problem solutions." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6.02 (1998): 107-116.

[17] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[18] Tsironi, Eleni, et al. "An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition." *Neurocomputing* 268 (2017): 76-86.

[19] Zhao, R.; Yan, R.; Wang, J.; Mao, K. Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks. Sensors 2017, 17, 273

[20] Xingjian, S. H. I., et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." *Advances in neural information processing systems*. 2015.

[21] Sainath, Tara N., et al. "Convolutional, long short-term memory, fully connected deep neural networks." *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.

[22] Tieleman, Tijmen, and Geoffrey Hinton. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude." COURSERA: Neural networks for machine learning 4.2 (2012): 26-31.

[23] Cheng, Gong, Junwei Han, and Xiaoqiang Lu. "Remote sensing image scene classification: Benchmark and state of the art." Proceedings of the IEEE 105.10 (2017): 1865-1883.