Deep Learning Based Classification Methods for Remote Sensing Images

Deepthi H, Mr. Suresh Kumar M
Student, DSCE Bangalore & India
Assistant Professor, DSCE Bangalore & India

Abstract: In this paper we mainly focus on urban and rural areas, which mostly contains urban built-up areas. Since their function strongly related to the distribution of built-up areas, where reflectivity or scattering characteristics are the same or similar, urban areas have recently been the focus of remote sensing applications. Traditional pixel-based techniques struggle to distinguish between different urban built-up region types. In particular, EuroSAT photos with many changes and scene classes are the focus of this paper's investigation of a deep learning-based categorization approach for remote sensing images. We discuss on convolution neural networks (CNNs) specifically to assist in the development of the corresponding classification algorithms in urban built-up areas. The effectiveness of the suggested methods is assessed based on their loss, precision, and confusion matrix scores as well as their total accuracy. The outcomes showed that SMDTR-CNN had the greatest overall accuracy (95.38%).

Keywords: Remote sensing, Deep Learning, CNN Algorithm.

I. Introduction

Deep learning has been used in numerous areas recently, including computer vision and wireless communications. Deep learning is thought to be an efficient method for extracting multi-layer features that frequently contain abstract and semantic information when compared to conventional pixel-based techniques (e.g., minimum distance supervision classification, iterative self-organization (ISO) cluster unsupervised classification, support vector machine (SVM) classification, random forest classification).

Deep architecture neural networks, such as CNN and CapsNet, can be used by deep learning algorithms to automatically learn features from the inputted raw data and produce effective deep learning features right away. When it comes to scene classification and object detection, these algorithms have produced several impressive outcomes. One of the crucial uses is the identification and classification of urban built-up areas, and its use with EuroSAT photos is significant for real-world applications. In the meanwhile, different land uses and urban functions are related. For instance, although city roadways and people both fall under the same framework, they have different functional purposes. In addition, the functional division of itself has connections to urban planning and emergency response. As a result, the classification of urban built-up regions is crucial for urban planning, evaluating the urban ecological environment, responding to urban emergencies, and other related purpose.

II. Literature Survey

Wenmei Li [1], Proposed a paper based on the hypothesis that land use and land cover classifications using remote sensing images have the same or similar spectra for the same and strange spectra for different ones. The traditional classification approach used pixel-scale spectral information to obtain a class map. Next, they combined the spatial information with the spectral information of the land use classification to improve the accuracy of the classification. With the improvement of the spatial resolution of remote sensing images, semantic level information is attracting attention in land use and land cover classification.

Christian Szegedy [2], Proposed a paper to provide solid evidence that approximating the expected optimal sparse structure through readily available high-density building blocks is a viable way to improve computer vision neural networks. The suggested result that you can see. The main advantage of this method is that the quality is significantly improved with a slight increase in computational requirements compared to a flatter and narrower network. Also note that the detection task was competitive, even though we did not use context or perform bounding box regression. This fact further demonstrates the strength of the Inception architecture.

Alex Krizhevsky [3], Proposed results showing that large, deep convolutional neural networks can achieve record-breaking results on highly demanding datasets using purely supervised learning. It is worth noting that removing the single layer of convolution results in poor network performance. For example, removing one of the middle tiers will reduce the performance of the top one in the network by about 2%. Therefore, depth is very important for getting results.

Gui-Song Xia [4], Proposed a paper on aviation scene classification for review and provided a clear summary of existing approaches. You can see that the progress of the aviation scene classification is severely restricted because the results of the currently popular datasets are already saturated. To solve this problem, we will build a new large dataset, AID. This is the largest and most sophisticated aerial image scene classification. The purpose of the dataset is to provide the research community with benchmark resources to advance cutting-edge algorithms for aerial scene analysis.

Xinhuai Zou [5], Proposed a new grid-based method for classifying tree species from TLS point clouds in complex forest scenes. Our method consists of individual tree extraction and denoising, voxel-based screening to represent tree characteristics, and tree species classification using the DBN model. Experiments have shown that both datasets achieve high accuracy. Screening is a powerful representation of 3D object information. We will continue to consider ways to represent 3D objects more effectively.

Junwei Han [6], A framework proposed to address the problem of object detection in Optical RSI. The innovation that distinguishes the proposed work from the previous work lies in two major aspects. First, this white paper provides a WSL framework that can significantly reduce the human effort of annotating training data while maintaining good performance, instead of using traditional supervised or semisupervised learning methods. Developed. Next, they have developed a deep network for learning high-level functionality in an unsupervised way, providing more powerful descriptors for capturing structural information about objects in the RSI. Therefore, the object detection performance can be further improved. Experiments with three different RSI datasets have demonstrated the effectiveness of the proposed work.

Haifeng Li [7], Proposed papers using crowdsourced data for some notable features such as B. Real-time classification, fast propagation speed, robust information, low cost, large amount of data Is the focus of research in International Geographic Information Science. The crowdsourced database RSI-CB provides new ideas and research directions for the creation and improvement of remote sensing datasets. RSI-CB has 6 categories based on China's land cover classification criteria, each category has several subcategories, and the number of categories and images is significantly improved compared to other remote sensing datasets. Has been done. Classification experiments performed on some traditional deep learning networks have shown that the RSI-CB's classification accuracy is higher than other datasets due to its greater spatial resolution and higher spatial resolution.

Xinting Yang [8], The proposed paper process conducted a thorough and comprehensive survey of current applications of deep learning (DL) for smart aquaculture. Categories are live fish identification, species classification, behavioral analysis, feeding decisions, size or biomass estimates, and water quality predictions. The technical details of the reported method have been extensively analyzed according to the data and algorithms that are key elements of artificial intelligence (AI).

III. Comparison study of "Deep Learning Based Classification Methods for Remote Sensing Images" based on various papers.

Sl. No	Authors	Description	Techniques	Advantages	Drawbacks
1	Wenmei Li	Deep Learning Based Classification Methods for Remote Sensing images in Urban Built Up Areas	Deep Learning, convolution neural network	Remote sensing is cheap.	Sometimes large scale maps cannot be prepared from satellite data which makes data collection incomplete.
2	Christian Szegedy	Going Deeper with Convolutions.	Convolution neural network	Remotely sensed data can be easily processed and analyzed fast.	Remote sensing systems such as radars that emit their own electromagnetic radiation can be

					intrusive.
3	Alex Krizhe vsky.	ImageNet Classification with Deep Convolutional Neural Networks.	Deep convolution neural network.	Remote sensing is constructive method a base map in the absence of detailed land survey methods.	engineering maps cannot be prepared from satellite data
4	Gui-Song Xia	AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification	Deep Learning, Geophysical image processing.	Aid helps rebuild livelihood and housing after a disaster.	It often comes with a high computational cost.
5	Xinhuai Zou.	Tree Classification in complex Forest Point Cloud Based on Deep Learning.	Machine Learning	The parameter initialization of the network is not random.	Line of sight.

IV. Methodology

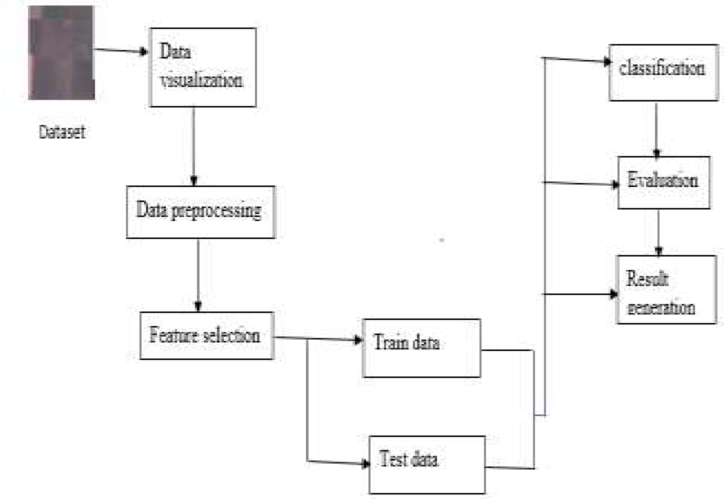


Fig. 1: System Architecture of our system

In Figure 1 illustrates the system architecture of our proposed system. A convolutional neural network is a deep learning algorithm that takes an input image, assigns importance (learnable weights and biases) to different aspects / objects in the image, and distinguishes them from each other. ConvNet requires much less pre-processing compared to other classification algorithms. The primitive method requires well-trained and manual development of filters, but ConvNet has the ability to learn these filters / properties. An image is nothing more than a matrix of pixel values, isn't it? So why not flatten the image (for example, a 3x3 image matrix into a 9x1 vector) and feed it to a multi-layer perceptron for classification? ConvNet can successfully capture spatial and temporal dependencies in an image through the application of relevant filters. The implementation architecture is better suited to image datasets due to reduced number of parameters involved and weight reuse. In other words, the network can be trained to better understand image sophistication.

Image size = 65 (height) x 65 (width) x 3 (number of channels, e.g., RGB). The element involved in performing the convolution operation in the first part of the convolution layer called the kernel/filter, K. We choose K to be a 3x3x1 matrix. The kernel shifts 9 times due to Stride Length = 1 (non-Strided), each time doing a matrix multiplication between K and the P part of the image, the kernel is hovering over. The filter moves to the right with a certain stride value until it parses the full width. Moving on, it will jump to the top (left) of the frame with the same stride value and repeat the process until the entire frame is traversed.

In the case of multichannel (e.g., RGB) images, the kernel has the same depth as the depth of the input image. Matrix multiplication is performed between stack Kn and In ([K1, I1]; [K2, I2]; [K3, I3]) and all results are summed with offsets to give us a channel complex function output at the crushing depth.

The purpose of the convolution operation is to extract high-level features, such as edges, from the input image. ConvNets need not be limited to a compound class. Usually, the first ConvLayer is responsible for capturing low-level features like edges, colors, gradient directions, etc. With additional layers, the architecture also accommodates high-level functions, giving us a network with a sane understanding of images in the dataset. When we increase the 5x5x1 image to a 6x6x1 image and then apply a 3x3x1 multiplier on it, we see that the transformation matrix turns out to be 5x5x1 in size.

Otherwise, if we do the same operation without padding, we'll see a matrix the size of the core itself (3x3x1) - valid padding. Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of results to the operation - one where the transformed function is reduced in size relative to the input, and another where the size is increased or kept the same. This is done by applying valid padding in the first case or the same padding in the second case.

There are two types of clustering: maximum clustering and mean clustering. Max Pooling returns the maximum value of the portion of the image covered by the kernel. On the other hand, Average Pooling returns the average of all values of the portion of the image covered by the kernel. The convolutional and pooling layers combine to form the i-th layer of the convolutional neural network. Depending on the complexity of the image, you can increase the number of such layers to capture even lower levels of detail, but at the expense of computing power. After going through the above process, the model is able to understand its function. The final output is then smoothed and passed to a regular neural network for classification.

Now that we've converted our input image into a shape suitable for our multilevel Perceptron, we'll flatten the image into a column vector. The flattened output is fed to a feedback neural network, and back-propagation is applied to each iteration of the training. Over a series of epochs, the model can distinguish between prominent features and some low-level features in an image and classify them using the Soft max classification technique.

A trained CNN has hidden layers whose neurons correspond to possible abstract representations on the input features. When faced with an invisible input, the CNN doesn't know what abstract representation it has learned will fit that particular input.

The Dropout layer is a mask that removes the contribution of certain neurons to the next layer and leaves all others intact. We can apply the Dropout class to the input vector, in which case it negates some of its characteristics; but it can also be applied to a hidden layer, in which case it will destroy certain hidden neurons. Dropout classes are important in training a CNN because they prevent over fitting of the training data. If they are not present, the first batch of training samples will disproportionately affect learning. This, in turn, prevents learning of features that appear only in later samples or batches. Using convolution, rectification, and pooling into three sub modules (layer C, pooling layer, fully connected layer) to produce the final comparison matrix. The output of the pooled layer is flattened and graded using the Soft Max activation function, which is used to classify the scene images.

There are two types of clustering: maximum clustering and mean clustering. Max Pooling returns the maximum value of the portion of the image covered by the kernel. On the other hand, Average Pooling returns the average of all values of the portion of the image covered by the kernel. The convolutional and pooling layers combine to form the i-th layer of the convolutional neural network. Depending on the complexity of the image, you can increase the number of such layers to capture even lower levels of detail, but at the expense of computing power. After going through the above process, the model is able to understand its function. The final output is then smoothed and passed to a regular neural network for classification.

Now that we've converted our input image into a shape suitable for our multilevel Perceptron, we'll flatten the image into a column vector. The flattened output is fed to a feedback neural network, and back-propagation is applied to each iteration of the training. Over a series of epochs, the model can distinguish between prominent features and some low-level features in an image and classify them using the Softmax classification technique.

A trained CNN has hidden layers whose neurons correspond to possible abstract representations on the input features. When faced with an invisible input, the CNN doesn't know what abstract representation it has learned will fit that particular input.

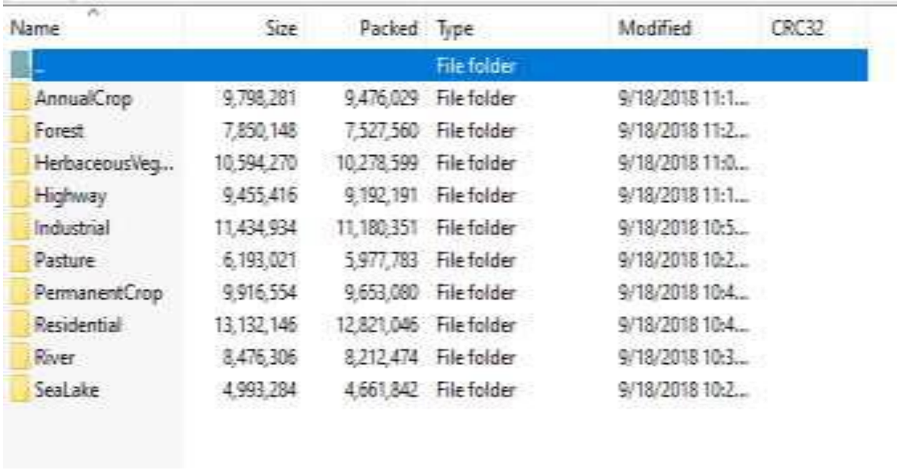
V. Implementation

There are 6 modules in this phase

- Data Selection and Loading
- Data Preprocessing

- Feature Selection
- Classification
- Prediction
- Result Generation

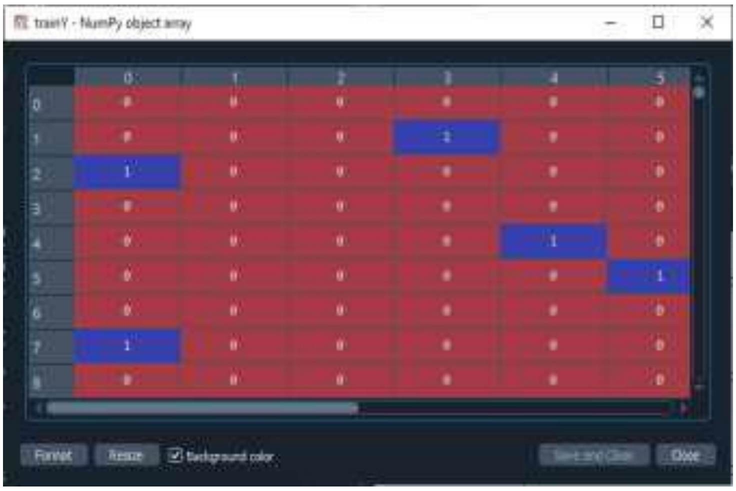
Data selection and loading: The EuroSAT datasets are collected and are loaded for preprocessing step. The data is the independent variable which are considered as the overall dataset, and the labels are the dependent variable, which contains 10 different labels such as annualcrop, forest, pasture, highway, herbaceous vegetation, sealake, industrial, pasture etc. The overall images contained are 27000 images.



Name	Size	Packed	Type	Modified	CRC32
..			File folder		
AnnualCrop	9,798,281	9,476,029	File folder	9/18/2018 11:1...	
Forest	7,850,148	7,527,560	File folder	9/18/2018 11:2...	
HerbaceousVeg...	10,594,270	10,278,599	File folder	9/18/2018 11:0...	
Highway	9,455,416	9,192,191	File folder	9/18/2018 11:1...	
Industrial	11,434,934	11,180,351	File folder	9/18/2018 10:5...	
Pasture	6,193,021	5,977,783	File folder	9/18/2018 10:2...	
PermanentCrop	9,916,554	9,653,080	File folder	9/18/2018 10:4...	
Residential	13,132,146	12,821,046	File folder	9/18/2018 10:4...	
River	8,476,306	8,212,474	File folder	9/18/2018 10:3...	
SeaLake	4,993,284	4,661,842	File folder	9/18/2018 10:2...	

Fig 2: datasets used in our model

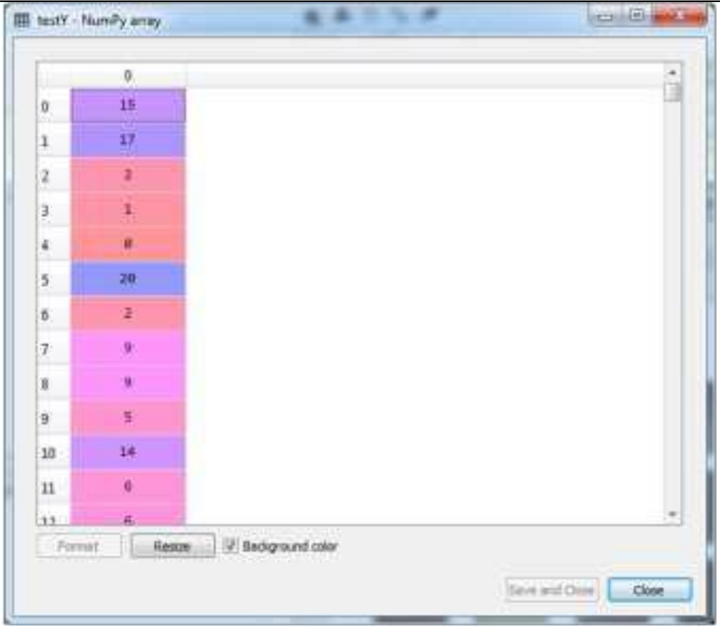
Data pre-processing: Here the images are converted to rgb from grayscale. The resolution of the image with minimum height and minimum width. Here hence we are classifying the low-resolution image that is 64*64 according to the configuration of system. We initialize the data for the empty list and append image into image path empty list. in which we select the first 10 list in the image path. Before which we shuffle the image to show different labels. Then we read the image from image path and resize the image and append images into data and convert the data into array because the algorithms allows only array format data. Here we have 0 and 1 images where images are present in 0th index and labels in 1th index. Hence here we properly arrange our dependent and independent variables.



	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	1	0	0
2	1	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1
6	0	0	0	0	0	0
7	1	0	0	0	0	0
8	0	0	0	0	0	0

Fig 3: Appending the images into array

Feature selection: we spilt the data into train and test datasets. So the input will be train x, test x and the train y, test y will be for testing labels. We categorize our dependent variable that is the train data.

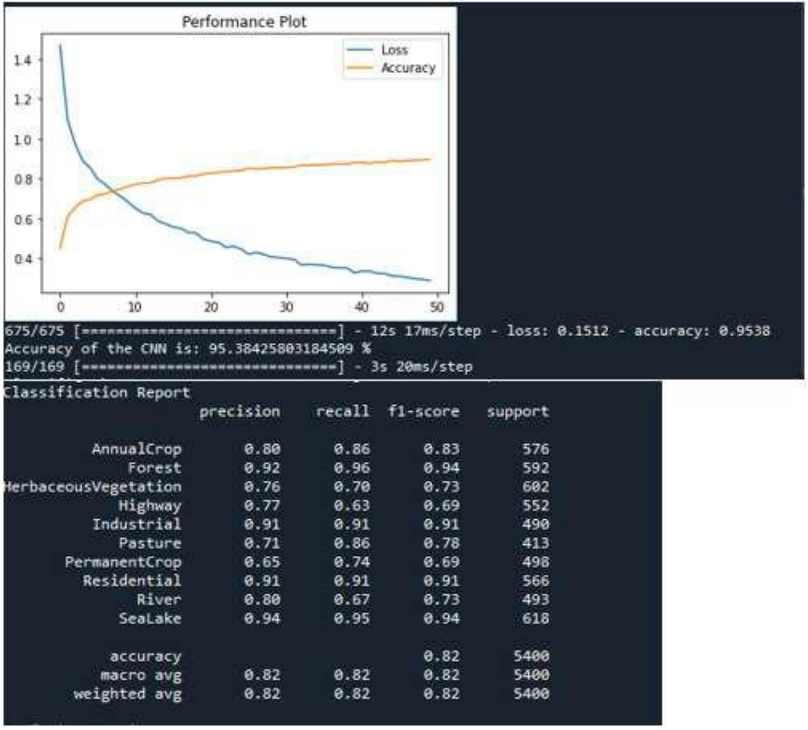


	0
0	15
1	17
2	2
3	1
4	8
5	20
6	2
7	9
8	9
9	5
10	14
11	6
12	6

Fig 4: splitting into test dataset

VI. Results and Discussion

Classification: Here the maxpolling reduces the dimension of the data. We extract the total trainable datasets that is 416. There are total of 32 filters in the algorithm. The extra or unwanted features will be eliminated and hence it is called dropout layer and there is no extra layer in our model so the value will be 0. Flatten layer is used to ensemble all the trainable datasets. In dense layer we apply Relu activation which is used to process the data with 128 features. In final dense layer we provide the proper count of the inputs and outputs. Activation softmax is used for multiclass classification coz we have 10 different categories in our model. Softmax extracts the final feature classification from neural network algorithm. Categorical cross entropy is also known as multiclass classification where we find the accuracy. Whenever we reduce the loss and increase the accuracy the prediction will be valid. Hence in every layer we reduce the loss based on which we calculate the epochs which are the training levels. In out model we've used 50 levels of epochs



```
675/675 [==============================] - 12s 17ms/step - loss: 0.1512 - accuracy: 0.9538
Accuracy of the CNN is: 95.38425803184509 %
169/169 [==============================] - 3s 20ms/step
Classification Report
                       precision    recall  f1-score   support

          AnnualCrop       0.80      0.86      0.83       576
              Forest       0.92      0.96      0.94       592
HerbaceousVegetation       0.76      0.70      0.73       602
             Highway       0.77      0.63      0.69       552
          Industrial       0.91      0.91      0.91       490
             Pasture       0.71      0.86      0.78       413
       PermanentCrop       0.65      0.74      0.69       498
         Residential       0.91      0.91      0.91       566
               River       0.80      0.67      0.73       493
             SeaLake       0.94      0.95      0.94       618

            accuracy                           0.82      5400
           macro avg       0.82      0.82      0.82      5400
        weighted avg       0.82      0.82      0.82      5400
```

Fig 6: Plotting performance graph based on accuracy and loss

Prediction: In final dense layer we extract the feature from the CNN layer and predict the images taking 32 images at once for optimization of time. And then we visualize the performance graph and accuracy increases for the training datasets. We use test y datasets for predicting the test data and compare the algorithm predicted result and the actual result. After prediction we find the classification report based on precision. Precision is number of predicted results. F1 score is used to measure the test accuracy based on all labels to increase this we should increase the epochs. Confusion matrix is used to collect number of predictions.

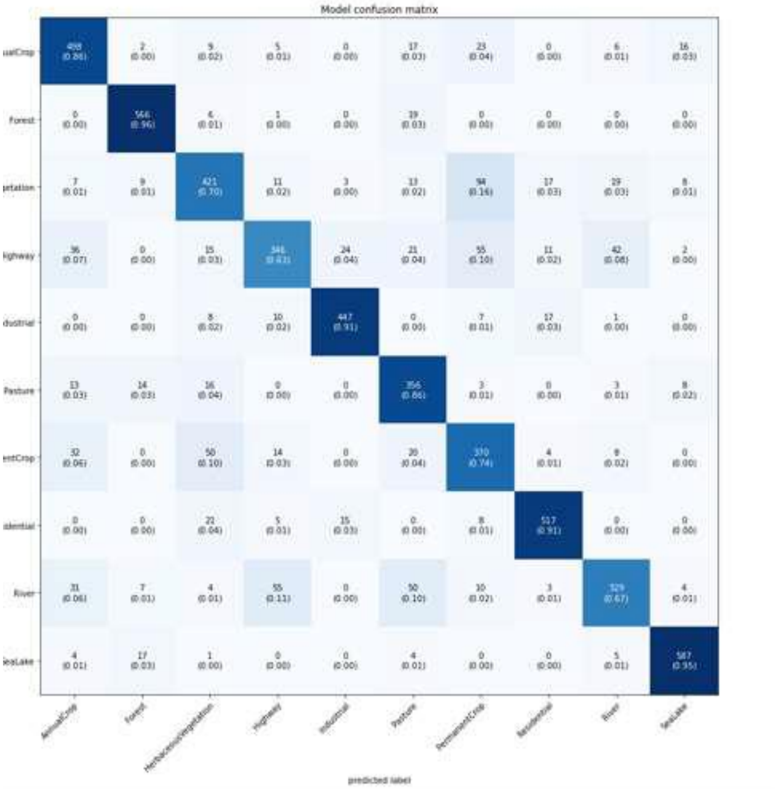


Fig 7: Confusion matrix for all the labels.

Conclusion

In this study, the deep learning classification is used to analyze the remote sensing scene from the images. With the help of EuroSAT datasets we select and view the datasets preprocess and spilt the datasets into train and test data, and then classify the images based on convolution neural networks for predicting the result of the image and the accuracy obtained for the lower solution images. The accuracy obtained with our study is 95.376%

References

[1]. Wenmei, Li.; Haiyan, Liu.; Yu, Wang.; Guan, Gui., “Deep Learning based Classification Methods for Remote Sensing Images in Urban Built-up Areas," 2019 Fourth IEEE International Conference on, vol., no., pp.7, 12 Jan. 2019.

[2]. C. Szegedy et al., "Going deeper with convolutions,"2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9, doi:10.1109/CVPR.2015.7298594.

[3]. Alex Krihevsky, Ilya Sutskever, Geoffrey E, Hinton “ImageNet Classification with Deep Convolution Neural Network” communications of the ACM Volume 60, Issue 6, pp 84-90, June 2017

[4]. Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguagng Shi,”AID: A benchmark dataset for performance evaluation of aerial scene classification” IEEE International 2016.

[5]. X. Zou, M. Cheng, C. Wang, Y. Xia and J. Li, "Tree Classification in Complex Forest Point Clouds Based on Deep Learning," in IEEE Geo Science and Remote Sensing Letters, vol. 14, no. 12, pp. 2360-2364, Dec. 2017, doi: 10.1109/LGRS.2017.2764938.

[6]. Hifeng Li,” RSI-CB: A Large-Scale Remote Sensing Image Classification Benchmark via Crowd source Data” National Library of Medicine, doi: 10.3390/s20061594 March 2020.

[7]. X. Tang, X. Zhang, F. Liu, and L. Jiao,"Unsupervised deep feature learning for remote sensing mimage retrieval," Remote Sens., vol. 10, no. 8, p. 1243, 2018.

[8]. Q. Zhu, Y. Zhong, L. Zhang, and D. Li, "Scene classification based on the fully sparse semantic topic model," IEEE Trans. Geosci. Remote Sens., vol. 55, no. 10, pp. 5525–5538, Oct. 2017.

[9]. X. Tang and L. Jiao, "Fusion similarity-based reranking for SAR image retrieval," IEEE Geosci. Remote Sens. Lett. vol. 14, no. 2, pp. 242–246, Feb. 2017.

[10]. J. Marçais and J.-R. de Dreuzy, "Prospective interest of deep learning for hydrological inference," Ground Water, vol. 55, pp. 688–692, 2017.