International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-11 Issue-8, July 2022

Classification of Landsat 8 Images using Neural Networks

Varsha Bhosale, Archana Patankar

Abstract*: Several geospatial applications use Classification. Classification is useful in identifying change detection. The Changes amongst various classes are identified at different time periods thus helping to analyze the changes happening in Land Use Land cover (LULC) of the area under consideration over a period. Neural networks have shown its command in majority of fields in solving complex problems, and geospatial field is also benefited by the Neural Networks. Several effective and efficient mechanisms are suggested for supervised satellite image classification. The Neural network's Machine Learning algorithms are gaining popularity for supervised satellite image classification. The objective of this paper is to show how the Convolution Neural Network (CNN) as a machine learning algorithm is implemented for classification of Satellite image. The study area considered is Mumbai Metropolitan region (MMR), the Financial Capital of India. The work was executed considering the Landsat 8 images. The images were obtained and processed in QGIS Open-Source Software; Machine Learning algorithm was developed using Python Scripting. The NN algorithm was effectively implemented, and results showed the competence of Neural Network in generating classification of Landsat 8 satellite Image using CNN.

***Keywords*: Supervised Classification, Land Use Land Cover (LULC), Machine Learning, Convolution Neural Networks, Python Script**

I. INTRODUCTION

Remotely sensed images can help in finding different solutions for the growing number of environmental challenges that the world is facing today. Leveraging the potential of Land Use and land cover will help us in making good use of the natural resources and will also help the Urban planners to plan well for the future. Land cover refers to the physical material covering the surface of the Earth including vegetation, water, soil and artificial surfaces built by human activities. On the other hand, land use refers to the way and how land is used by humans and their habitat (Ramachandra and Kumar, 2004). Machine Learning Algorithms are used for various functionality in Geospatial studies. The various steps need to perform classification of Landsat 8 Image using Neural Networks is depicted in this study

Manuscript received on 11 June 202 | Revised Manuscript received on 19 June 2022 | Manuscript Accepted on 15 July 2022 | Manuscript published on 30 July 2022.

* Correspondence Author

Varsha Bhosale*, Research Scholar, Department of Computer Engineering, Vidyalankar Institute of Technology (VIT), Mumbai (Maharashtra), India.

Dr. Archana Patankar, Professor, Department of Computer Engineering at Thadomal Shahani College of Engineering, Mumbai (Maharashtra), India.

II. STUDY AREA AND DATA SET

A. Study Area

The study region is the Mumbai Metropolitan Region (MMR), which is the capital of Maharashtra state, India's financial center, and a major urban environment in the western half of the country. Mumbai is in western India between the latitudes of 19° 33' 33" N and 72° 52' 39" E. (Figure 1).

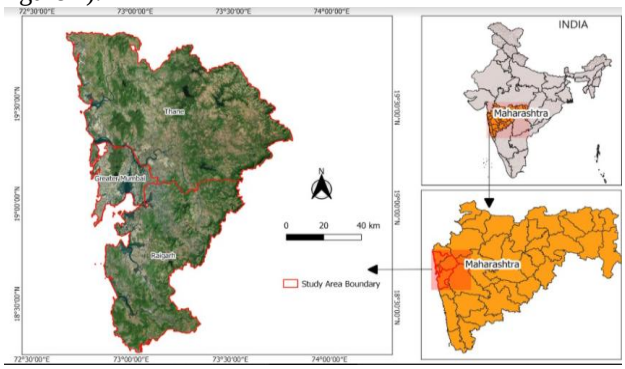

Figure 1: Study Area, Mumbai Metropolitan Region

The Mumbai Metropolitan Region (MMR) is a Maharashtra metropolitan area that includes the state capital Mumbai (formerly known as 'Bombay') and its satellite towns. It is made up of eight municipal corporations (Greater Mumbai, Thane, Kalyan-Dombivali, Navi Mumbai, Ulhasnagar, Bhiwandi-Nizamapur, Vasai-Virar, and Mira-Bhayandar) and nine municipal councils (Ambarnath, Kulgaon-Badalapur, Matheran, Karjat, Panvel, Khopoli, Pen, Uran, and Alibaug), as (https://mmrda.maharashtra.gov.in/about-mmr).It is India's most active and bustling city, as well as the capital of Maharashtra's western state. It encompasses Mumbai city (complete), Mumbai suburbs (complete), Thane (complete), Palghar (partial), and Raigad (partial) (Partial) The Mumbai Metropolitan Region Development Authority (MMRDA), a Maharashtra State Government organization, in charge of town planning, development, transportation, and housing in the region, oversees the entire area. MMRDA is in charge of ensuring that the MMR develops in a balanced manner. [2] The MMRDA was established to solve the issues of metropolitan region infrastructure planning and development. Outside of Brihan Mumbai (Greater Mumbai) and Navi Mumbai, several places have lacked well-planned development. City and Industrial Development Corporation, a Maharashtra government-owned enterprise, advertised Navi Mumbai as one of the world's largest planned cities (CIDCO).

As a result of increased urbanization, the region had issues with haphazard and unlawful growth. Villages around NH3 in Bhiwandi Taluka are examples of haphazard MMR development, including some of India's largest warehousing areas. According to the 2016 Census, the Mumbai Metropolitan Region (MMR) covers 6,355 square kilometers and has a population of 228.85 lakhs.

B. Data Set

The satellite images were obtained from the United States Geological Survey's website. To cover the full MMR region, three tiles have to be downloaded. Landsat 8 Level 2 pictures from 2020 were mosaicked and trimmed to the MMR region area before being used.

The clipped area was processed using seven Bands (Band 1 to 7) of Landsat 8, Level 2 photos from the year 2020.

III. MACHINE LEARNING

Machine learning (ML) is a generic algorithm that learns from data. The data is provided by the user, and the algorithm produces a model that performs a specific task. It aids in the development of intelligent systems, or self-contained systems. The systems must be trained before it can be used. To train the systems, a large number of training examples are required. Regression, Classification, and Prediction are tasks that require a model. Eg: Machine learning is used in various applications like Image Processing, Recommendation systems, Voice recognition, Object Identification, Monitoring of Video feed, Cancer Prediction, email categorization, Sentiment Analysis, etc... In Non- Machine learning methods, the user gives Input data and the Logic (Program) and based on the processing the results are obtained. In case for Machine learning the user gives the Input and the desired Output, at the end a relationship needs to be established i.e., the algorithm learns from the data, and gives a model. Once the learned model is obtained the Input data needs to be given to obtain the desired output.

The various steps in Implementing Machine Learning algorithm are

1. Collecting the Data
2. Pre-processing the Data – Performance of the model depends on the pre-processing of the data
3. Training and Testing Data – Data Splitting is required for Training and Testing, to validate the model to check its accuracy.
4. Evaluating the Performance

While training the model, the quality of the data needs to be very good. Data Pre-processing needs to be done to give good Input data, when good data is given then the results will be good. The Training data and the Testing data need to be similar in order to get good accuracy from the model.

Terminologies related to ML

- **Model –** Specific representation learned from the data by applying some ML algorithms.
- **Feature –** Individual measurable property of data.
- **Feature Vector** – Set of numeric features which are conveniently described by the feature vector.
- **Target or Label** – Value to be predicted by the model.
- **Training –** Algorithm with is built from historical data
- **Testing / Prediction** – Finding the output from the model by giving in Input Feature Vector

In case of Image, we need to perform Image Classification of Multispectral Image of RGB and NIR Band. A pixel will have values related to RGB and NIR Band.

Suppose the Pixel belongs to some LULC class Eg: Vegetation class (or Target/ Label) of the pixel, the measurable quantities in the Bands corresponding to vegetation, then the RGB and NIR will have values or measurable properties. So, RGB and NIR are Features and combination of the band is Feature Vector.

Machine Learning is used mainly when the parameters are non-linear, where Statistical models do not fit. The ML algorithms improve the performance of the Statistical Models. It is also used for Images with larger size or large no. of Images. The categories of ML classification are Unsupervised and Supervised Learning

IV. NEURAL NETWORK BASED CLASSIFICATION USING PYTHON

The Neural Network algorithm was executed on the MMR dataset. The Python Script [1] were written to perform the following basic steps of Neural Network algorithm

- Create the Neural Network
- Implement the Forward Propagation & Backward Error Propagation Algorithm
- Train the Network Model
- Update the weights
- Train the Network
- Test the Network

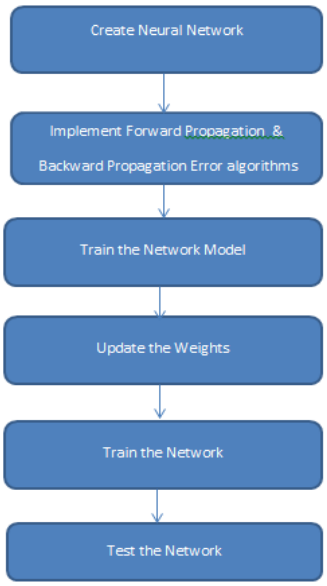

Figure 2: Steps Performed for Artificial Neural Network based Classification of Satellite Images

A. Create a Neural Network

To build Forward and Backward propagation Error models, a network is created. The Network models are then trained to produce NN-based classifications. Structures are used to keep track of NN's details.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

NN has learned something means that it will have specific weight information based on a certain layer and will make predictions based on it.

The hidden layer is connected to the input layer, which has nodes. Through weights, each input layer will be connected to each hidden layer. In addition, each of the nodes in the Hidden layer is connected to an extra Bias node. The connections are defined in the same way from the hidden layer to the output layer, except that the hidden layer now serves as the input.

B. Implement Forward Propagation Algorithm

At the beginning any NN training should give the weight to initialize the network. Following the creation of the weight, we establish forward network propagation, which takes the input, multiplies it by the weight, and generates the matching output. Each node/neuron has its own activation function, i.e., Activation Function w.r.t. the input weight. The Activation function produces the activation output. It is calculated the weighted sum of all the input layers. After that the transfer activation function generates the relevant output based on the input. For the transfer activation function, the Sigmoid (S(X) = 1 / (1+ ex) value is used. The output generated by the neuron in the corresponding hidden layer will be the weighted sum and related transfer activation function output. It is the output generated by a neuron in the appropriate hidden layer once the weighted Sum and the corresponding transfer activation function are obtained. This holds true for all neurons in the input layer as well as the network's hidden layers.

C. Implement Backward Propagation Error algorithm

This step requires the creation of three different parameters: one is the derivative with respect to the activation function, another is the derivation of an activation function with respect to the input weight, and finally the input weight which is dependent on the result in the model. The back propagation formula uses a network and a corresponding predicted error. The overall error will be calculated, and the weight will be updated accordingly. The error must be spread out among all of the connections. The first error is generated by the output layer, and it goes backwards towards the input. Before utilizing the back propagation procedure, the transfer derivative function must be established. The sigmoid function's derivative (S'(X) = X * (1 - e) is used here, as the function is used in forward propagation. Backpropagation Error must be dispersed among all connections.

Updates the weights – Once the Backpropagation error has been calculated, the weights must be changed. The relevant learning rate value and the accompanying derivative are used to update the values, and the corresponding layer or row must be altered.

Train the Network – The network is used for training, by setting the training data, learning rate, and number of epocs used for training, as well as the number of outputs. We perform the training for each epoc and then compute the cumulative error. For each error for each epoch on each training the error is accumulated. For each of the training sample input the output generated by forward propagation network is calculated. The expected output that is supplied for a specific row or layer is identified. After that, the layer output values are given. We compute the overall sum, which is the predicted value minus the output created; as the summation square error or RMSE error is utilized, the square of (expected – actual value) is taken; this value will be generated for each of the neurons. The sum is computed for the total expected values, which is the total summation of errors, and the backward propagation is carried out based on this error sum, which is the total backward propagation error generated, and is then used for updating. The network from the previous step is used as an input, and the expected values are used to calculate the back propagation error. Back propagation error is a two-input network with expected output values that is used in training. The back propagation error is created so that the value is stored directly in the network and then updating of weight is done on each epoch at a later stage, so updating of weight for the network that will get created or will be created is done. Once the weights are updated for the network that will be created or generated for each of the layers based on the learning rate, we can look at the epoch function to see what the learning rate was at a particular epoch, the precision of the learning rate is set with three precision values, and the cumulative error is displayed.

D. Test the Network: The above training sets is tested on two input and two output network and two nodes at the hidden layer. The epocs considered are 20.

V. CONVOLUTION NEURAL NETWORK (CNN)

The basic ANN has some limits or drawbacks, such as losing the spatial orientation of an image as the number of parameters increase. As the number of parameters increase, so does the number of samples, and therefore the computation capacity. As a result, this network has been altered. The process of extracting features, as well as the task of classification, are both carried out in the same computational architectures in the case of CNN. The basic goal of CNN is to extract features first, then combine them to perform categorization. In CNN, samples are presented, and distinct characteristics are identified, and classifications of the features are derived based on the sample. The CNN is commonly used to handle a variety of complex problems, such as image recognition and classification, by using a series of feed-forward layers. The CNN is similar to a traditional neural network in that it is made up of neurons with learnable weights and biases. Because the neurons receive a set of inputs and perform some non-linear processing, it can be thought of as a feed-forward artificial neural network [4]. A convolutional layer, a pooling layer, and full connection layers are all part of the conventional CNN structure [5]. The CNN approach consists of a series of processing layers, each of which learns how to represent data from low to high levels. Such characteristics give data that can be combined later to detect higher-level features, allowing the CNN to behave as an automatic feature extractor. Many recent computer vision algorithms use CNN to extract local characteristics that are limited to specific sub-regions of the image.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Convolutional Layer (Conv), Activation Function (in this example it is Rectified linear unit (ReLU) layer, pooling layer (Pool), and fully connected layer (FC) are the four primary types of layers in a basic CNN (Figure 1). Figure 3 depicts a basic CNN structure

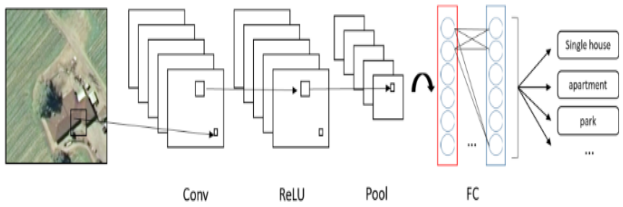

Figure 3 Convolution Neural Network (CNN)

An image is organized as a matrix of pixel values. Given a random input image, the CNN recognizes the class it belongs to and the values of probability that the input belongs to each class. There are four basic types of layers making up a CNN as presented on Figure 4 and their roles are as follows:

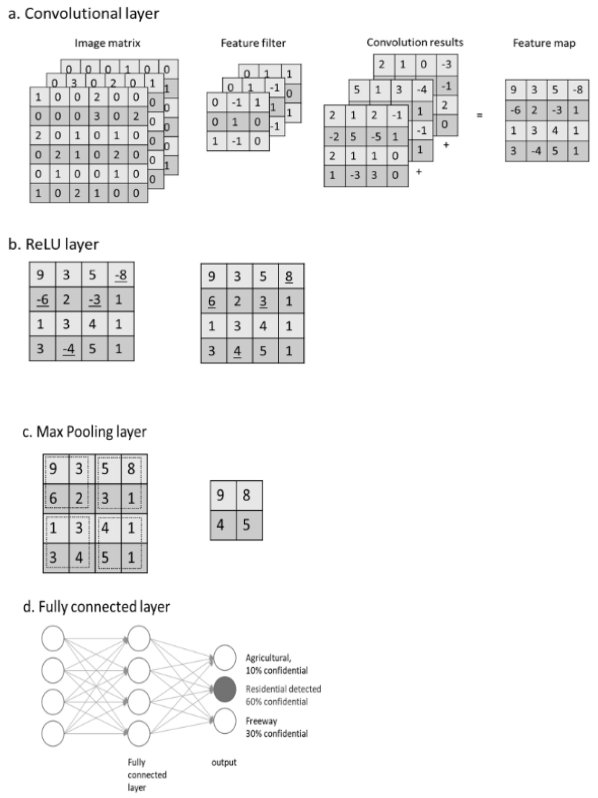

Figure 4 Layers in Basic Convolution NN

4a. Convolutional Layer (Conv)
4b. Activation Function (Rectified Linear Unit Layer (ReLU))
4c. Max pooling Layer
4d. Fully Connected Layer

Convolutional layer performs the linear operation by doing element-wise multiplications and summations on every sub-input, using a group of weight matrixes called feature filters (Figure 4a). Feature filters learn their weights from training data; then each filter can detect the existence of a specific feature.

The outputs are called feature maps. A feature map records not only the operation values but also the relative spatial location of these values. On the feature map, a higher output value indicates the potential existence of the corresponding feature at its relative location. Rectified linear units (ReLU) layer performs a nonlinear function (such as tanh and sigmoid) on the input (Figure 4b). Introducing nonlinearity to the system can improve computational efficiency without losing much accuracy. When performing a threshold operation: f(x) = max (0, x), where f is a nonlinear function applied to all values x in the matrixes, the negative values of input matrixes are rectified as 0 keeping the size of input volume unchanged. Pooling layer is a down-sampling layer (Figure 4c), aiming to decrease the size of the feature maps and to reduce the computational cost. A max pooling layer keeps only the maximal number of every sub-array (2 × 2) as the element of output array. Output omits unimportant features, while keeps the relative location of features The fully connected layer multiplies the input by a weight matrix and then adds a bias vector to the output, an N-dimensional vector (Figure 4d). The output gives the probability of the input image belonging to each class, where N is the number of the category in a classification task.

VI. IMPLEMENTATION OF CNN USING PYTHON

The processing of Satellite Images for Classification is carried using Python Scripts [1] with the help of GDAL library of python. The feature which is consistent across data is considered. The large image is divided in smaller sizes according to the input requirement. The following steps are executed to perform Classification of the Landsat 8 Image into two Classes Builtup and Non-Built up

A. Taking an image: The input Satellite image is taken and small Tiles / chips are created they are generally of the size 64 x 64 or 128 x 128 or 256 x 256. The output generated from this tiling process is moved to input folder. The Labels (Classified) input Satellite image is also considered. The classification is done for dominant features of Built Up and other types. Subsampling of the dominant feature is considered, and the CVS file is made.

B. Creating multiple smaller chips from the Image and then taking the dataset and organizing it into a feature and corresponding input NumPy array: The details of Bands present in the Chip are obtained and the data for rows and columns. The size of the chip considered is 64x64 and the no. of bands are 7, all the files are considered. The file with No. of Bands and the chip size is read, then it is stored in a NumPy array. For each file a NumPy array is created with no. of bands and no. of rows, this data is there in the features and the corresponding output levels, so in the level there is only one band giving the information about the images. In this two features and corresponding no. of bands we start storing the information present in the different chip which has been listed in the images. It is basically an image list, it will have a chip no. so from this chip information will get stored in an index no. and corresponding rows columns and no. of bands will also be stored, these all will be part of the features.

Similarly in case of labels, labels are actually being represented or will have a prefix like subset 2005 and corresponding chip no. so these two are stored and based on this two stored information the feature and level NumPy sets will get created once these details are available then we can look into the no. of features available to us and can also visualize it just to make sure that corresponding Image is being created.

C. Reading the Chip data and Storing into Normalized format input and output: In this step within a feature we need to Identify which level are we going to train, here a simple CNN is created which classifies the image in two categories one is the Built up class and the other is non built up, the label of the built up class is considered, so only those labels are to be extracted and put into a level feature rest all of them are discarded. Similarly in case of all the places in the input feature are also subsetted into Built up features, non-built-up features will be there where built up feature is not present. These two are combined and put up as separate classes and then we look for test or samples where the built up and non-built up are can be categorized, so after doing this there is a classification of the image where it gets categorized into two parts one is where the builtup is present and another one where built up is not present, so CNN network shall do the classification for two labels one where builtup is present(1) and another where builtup is not present (0). To avoid the large sampling error, we take a small portion of the input features where feature is present as well as the non builtup area is also selected in the same ratio. We are creating the places where samples are having a builtup and non- builtup areas of equal record. After doing that we do a kind of combination of this two information so it's basically ensuring that in the feature and input feature label there are equal no. of process and non-present classes and these two are actually present in the same nos. Once this is done then normalization is done in the range 0 to 1. If normalization is not done, then weight will fluctuate heavily. This creates the Label feature along with the input features which is normalized

D. Divide the samples into Training and Testing groups: All the Samples are taken 60% are used for training and 40% are used for testing based on random selection. The function will select all the Input features and does random selection so that features are obtained based on the randomly generated indexes so that the data set is sub-setted into two groups as training and testing groups

E. Modelling using the Data set: For modelling the Keras Library is used to perform Machine Learning. The library will help to create large, complicated networks which can be used for creating a output datasets. To create an image a model framework is created and then a Convolution Filter is created, total 32 different convolution filters are created of filter size is 1 x 1(i.e., 1 x 1 window is used). With 32 different filters based on the input size (i.e. 64 X 64)in this case along with the seven bands we will take the input and corresponding first convolution network is created, then the second convolution is created, so the second convolution filter is created and added on top of that then again in between these two stages there is a special layer called drop out, In the dropout layer, if certain data sets are to be avoided then dropout is a way of correcting that, during the training process the randomly selected link of the weightage will be put down to zero values i.e. switching off the corresponding output from one layer to another layer so that is what the dropout function does, the user needs to specify how many of the network connections should get drops in this case it can be 25 % dropout we can accept so there are two convolutions then there has to be a drop of or the features that has been learnet under the convolution features the corresponding weight will be whenever we are going to use those weights out of those convolution and weight we are going to drop the 25%.

The 32 filters first get created, after the first 32 filter there is an output generated so the second stage 48 next filter is getting created. After these two simple convolution layer we connect a fully connected network so that basically ensures that the total 64 x 64 input features generated by the input are getting fully connected dense network means there is full connectivity between all these 44 by 48 by 45 nodes and corresponding 64 fully connected networks are generated in this case so once we are having this fully connected 64 dense network then again we are having a dropout of 50% and finally we generate a softmax giving the output in two range which is basically value between 0 and 1 and that is being getting controlled by with the help of activation function ReLU and corresponding softmax function so softmax function will ensure that our output is actually in two different categories that's what after the unbundling and then creating a two process output the corresponding model is complete. So once we are having the model created this model will actually look like that will be like, it will be having a structure where there will be a input 64 x 64, we are having 7 bands so 7 x 7 x 64 input size will be provided, then 32 Convolution networks and then 48 in the next layer so two layers after the input , third layer and finally fourth so it's basically a four layers with two outputs will get generated so in between you are having an input and output layer and the three in between are the hidden layers. Once this model is ready then the training is done with the help of model fit, whenever we are creating a model for training purpose then we need to define a kind of differences in this case we are just dealing with two categorical datasets so categorical cross entropy is the function which will be used for creating the losses which will be used for backpropagation and for that we are going to use the optimizer RMS is the error so RMS proportion error will be used for optimization and basic accuracy of measurement of computing of the losses we are going to use the metric accuracy, so once this model is created we have to fit it so the total data set will get executed only on the training sections and once the training is complete then we can use for evaluation of its own data set based on the same training set (i.e. part of the input)i.e. same input and same corresponding output.

A Confusion matrix library of sklearn(scikit learn) is used and we are extracting different scores or matrix for generating the evaluation matrix, so once we have the evaluated matrix available to us then we can compare it with the result and the result will get generated in the computing of the confusion matrix which can be used for evaluation of the models. This will ensure that the predictions and testing samples or the model is ready to execute.

For training we use the training data set and for Testing we use the Testing data sets, so 40% will be used for classifications, after that we evaluate the results and finally save the output in a corresponding location.
This gives the output in form of model file which can be used for predictions. Training needs to be done just once, so once we have the trained model available and evaluated and we conclude that this is a maximum result expected from this network are obtained then these train models can be used for predicting the new data sets.
To create new data sets whatever input satellite image is there it needs to be broken down into smaller chip using the tile functions as we are having earlier where each of the tiled objects again needs to be passed on or each of the satellite images have to be converted in the form of features as we did for the input and finally what may happen is that we can use each of those chips to generate the predicted image and corresponding predicted image output can be stored for the specific tile. To generate the final results all the generated outputs are to be integrated as we have split the input in the beginning, we have to reverse the split done in the beginning. All the 64x64 chip individual images will get stitched together and the output Satellite image will be obtained.

VII. RESULT AND DISCUSSION

Machine learning algorithms are considered as the most modern and major method for mapping LULC. Neural networks NN is one of the main ML algorithms that gained robust performance in supervised satellite image classification area. The Neural Network and Convolution Neural Networks algorithms were executed using Python Script on Landsat 8 image which was mosaicked for Mumbai Metropolitan region (MMR) using QGIS 3.22. The performance of the NN algorithm is dependent on the training data set.

VIII. CONCLUSION

There are many papers which give the details of Machine Learning algorithms, but this paper gives a step-by-step process for performing supervised classification of two classes, that is implemented in python for NN and CNN. This can further be extended to Multiple output classes.

ACKNOWLEDGMENT

The authors are grateful to Vidyalankar Institute of Technology, Wadala(E), Mumbai, and the University of Mumbai for their assistance with this project.

REFERENCES

1. ashutoshkumarjha/IIRSOUTREACHPROGRAMME-2022-AIEO: Labs data and program for the Artificial Intelligence (AI) for Earth Observation (EO) and Geodata Handling and Processing (github.com)
2. https://en.wikipedia.org/wiki/Mumbai_Metropolitan_Reoon_Development_Authority)
3. Ramachandra TV, Kumar U (2004). Geographic Resources DecisionSupport System for land use, land cover dynamics analysis,Proceedings of the FOSS/GRASS Users Conference, Bangkok,Thailand.Reis S
4. Yu, X., Wu, X., Luo, C., Ren, P.: Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework. GISci. Remote Sensing (2017) [CrossRef]
5. Cheng, G., Li, Z., Yao X., Guo, L., Wei, V.: Remote sensing image scene classification using bag of convolutional features. IEEE Geosci. Remote Sensing Lett. 14(10), (2017) [CrossRef]

AUTHORS PROFILE

Varsha Bhosale, is currently a Research Scholar in Department of Computer Engineering, Vidyalankar Institute of Technology (VIT), Mumbai. She is currently pursuing her Ph.D. in Geoinformatics from Mumbai University and has completed M.E. (Information Technology) from University of Mumbai in 2005, with total Teaching Experience of 23 Years, which includes managerial Experience of 17 Years. She has published 20 Technical Papers in various Conferences and Journals and authored 2 books. She has guided 15 students in Master of Engineering.

Dr. Archana Patankar, is a Professor in Department of Computer Engineering at Thadomal Shahani College of Engineering, Mumbai. She is also Dean Research and is awarded Distinguished Professor Award by CSI-Technext India. She has completed her PhD from NMIMS, M.E. from VJTI and B.E. from Walchand College of Engineering Sangli. She has total of 22.5 Years of Teaching Experience along with 15 Years of ME Teaching. She is guiding PhD students since 7 years has 90 publications in various International Journals, Conferences, IEEE Explore Springer, Scopus Indexed, some are invited Papers, Editors choice and Best papers She has guided 30 groups, 25 at Master's level, 2 PhD students, 1 PhD student has subitted her thesis and currently is guiding 4 reasearch scholars.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)