JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

REMOTE SENSING SCENE CLASSIFICATION USING DEEP LEARNING

[1] Achanta Divya Krishna sree, [2] Durga Ganga Rao Kola

[1]PG Scholar, Department of Electronics and Communication Engineering, UCEK(A), JNTU Kakinada, Andhra Pradesh, India, 533003.

[2]Assistant Professor, Department of Electronics and Communication Engineering, UCEK(A), JNTU Kakinada, Andhra Pradesh, India, 533003.

Abstract: Remote sensing (RS) images are valuable data sources to measure and observe detailed structures on the earth's surface. As the amount of remote sensing images drastically increases, it becomes more challenging to understand and analyze such a huge data set. Hence, the classification of remote sensing image scenes has drawn more attention in academic studies and industrial uses. However, Advanced computer vision, has made significant progress in classifying millions of sample images. In recent years, Deep learning algorithms have been rapidly developed and widely used, Scene classification uses this field as well and has achieved desirable results. In this work, the application of scene classification using different types of deep learning models on UCMerced land use data set and made performance comparisons among various deep learning models like Convolution neural networks (CNN), GoogLeNet, VGG-16, EfficientNet-B0 and CNN model with recurrent Long short term memory (LSTM) network. It is observed that GoogLeNet and VGG-16 show better accuracy for scene classification upon the UCMerced dataset.

Keywords: **Remote sensing scenes, Deep learning, convolution neural networks (CNN), GoogleNet, VGG16, EfficientNet-B0, long short-term memory (LSTM)**

1. INTRODUCTION:

Remote sensing (RS) is the field of science that obtains information about objects or areas from distance i.e., from aircraft or satellites. Remote sensing images are valuable data sources to measure and observe detailed structures on the earth's surface. Those images contain more objects, and as the amount of RS images is growing drastically that becomes a difficult task for researchers to analyze such huge data. As a fundamental and suitable technology, scene classification plays an important role in the remote sensing field. Images collected from normal sources are different from satellite images. Hence remote sensing scene classification becomes a hot topic in academic research and industrial applications.

Scene classification means extracting the features of an image and based on those features image scenes are classified i.e., labeling scenes to which class the image relates. For extracting features from images different methods are developed, according to the previous works those methods are categorized into 3 types. handcrafted feature extraction, unsupervised feature extraction, and supervised deep learning techniques. In handcrafted feature extraction methods, features are extracted using color histograms, texture descriptors, global image scale invariant transform (GIST), scale-invariant feature transform (SIFT), and histogram of oriented gradients (HOG). Here the features are given manually by humans. It may not show robust results for classification. Second category is unsupervised learning in which the features are extracted using a principal component analysis (PCA), autoencoders, and sparse coding techniques are used for extracting features. Here the images are not provided with labels that are not appropriate for the strong classification but these methods show better results than handcrafted methods. Third category is supervised learning methods in that deep learning techniques are becoming popularly used and widespread models. Strong classification using deep learning techniques are override the previous techniques. Remote sensing images also turned to deep learning models and achieved desired results. Specifically based on deep convolution neural network (CNN) architectures show the improved state-of-the-art in object detection and classification. AlexNet is the 1st breakthrough CNN network in deep learning models. It is a simple CNN architecture with 5 convolution layers. [1]

As the resolution of the satellite images is different from normal images, to classify more types of objects in those images, new networks have to be developed. For this deep learning algorithms are changing their architecture deeper by increasing the number of convolution layers and pooling layers. CNN architecture is scaled either by depth, width, and resolution to become a

deeper network that is used in desired applications containing a huge data set. Chaib et al. [2] proposed to use VGGNet which contains a total of 16 layers with different sizes of convolution filters. It extracts the features from high-resolution images more accurately. Likewise, by changing the sizes of network architecture to the same baseline CNN network layers, various CNN pre-trained models are developed. Such pre-trained networks like VGG16, GoogleNet, EfficientNet-B0, and CNN network with recurrent long short-term memory (LSTM) network are used for remote sensing scene classification in our study and analyze the performance of the individual network on UCMerced land use data set.

2. convolution neural networks (CNNs):

Convolution neural networks are a type of network in deep learning. The CNN based classification methods uses a complete learning process, as compared to the traditional approach to classifying images. The image is given as input, the network itself conducts training and prediction and gives the classification output. CNN architecture consists majority of layers are the convolutional layer, pooling layer, normalization layer, activation layers, and fully connected layers. Typically, images are pre-processed before feeding into the network through the input layer. After that, the processed images are gone through the convolution and pooling layers that are ordered alternately. Finally, the outcome from the pooling layer and convolutions layers are given to the fully connected layer that classifies the final outcome and gives output. Figure 1 depicts the CNN structure.

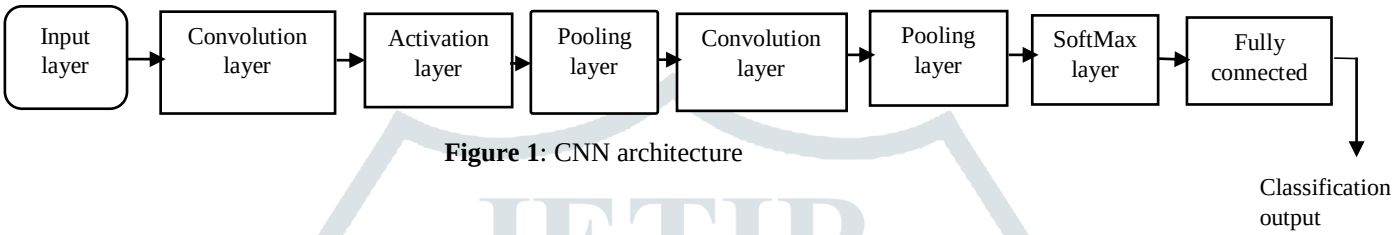

Figure 1: CNN architecture

2.2 Convolution layer:

The convolution layer is the first layer in feature extraction. It consists of a number of filters that performs convolution operation with the input image weights and gives different types of features based on which filters and the number of filters we are using. Each filter in the convolution layer is K*K*C in size, which means that the number of convolution filters should match the number of input channels of an image. Usually, the size of the input feature maps is H(height)*W(width)*C(channels). Convolution process in the layer is shown in figure 2. For example, the input feature map of size 5*5*3, then the size of the convolution filters is 3*3*3. This convolution filter passes over the entire image and gives the convoluted output matrix of size as the filter used. [3]

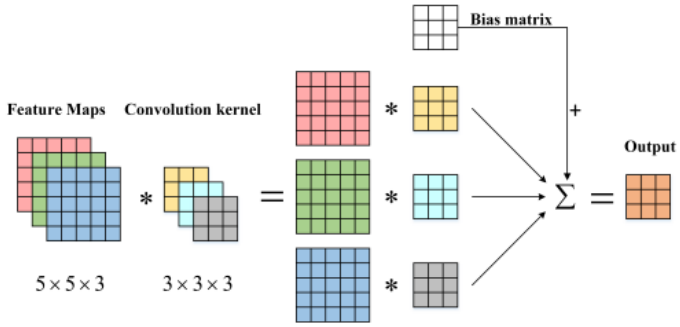

Figure 2: convolution process

2.3 Activation layer

The purpose of the activation function is to establish a basic relation between the input and output, which introduces a linearity and non-linearity system into the neural networks. The performance of the network can be improved with the proper activation function. There are several activation functions like sigmoid, tanh, and ReLU. ReLU is the linear activation function which has the function considered as f(x)=0 when x<0, and f(x) =x when x>=0. Sigmoid and tanh are non-linear functions. The output of the Tanh function saturates at -1 or 1.

2.4 Batch normalization layer

The minibatch data resulting from all the observations of each channel is normalized independently in this layer. To increase the training of the CNN network and reduce the sensitivity that initializes the network. Usually, the normalization layer lies between the convolution and activation layers.

2.5 pooling layer

The pooling layer comes after the convolution layers. Pooling layers work to gradually lower the representation's dimensionality, which in turn lowers the model's computational complexity and parameter count. There are different types of pooling they are max pooling, and average pooling, other techniques, like L2 Pooling, Mixed Pooling, Stochastic Pooling, Spatial Pyramid Pooling (SPP), and Multi-scale Orderless Pooling, etc... Among them max and Average pooling are shown in Figure 3. [3]

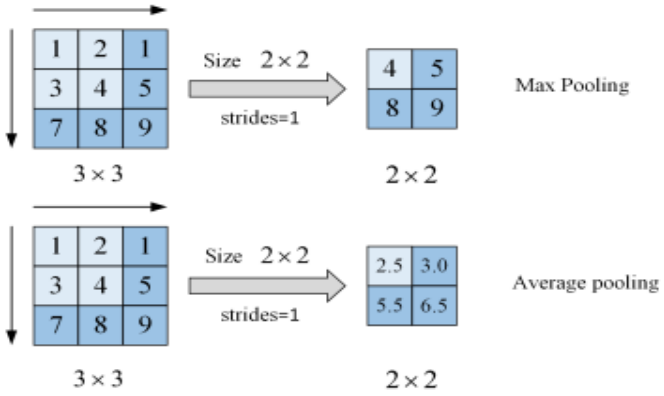

Figure 3: Average and maximum pooling [3]

2.6 Fully connected (FC) layer

The fully connected layer is the final layer which is used for classifying images. It is possible to utilize one or more fully connected layers, in which every neuron is linked to every other neuron in the layer above it. The number of classes for the considered dataset should be defined in this layer and the probability of classification score will be provided in this layer.

3. Models

In this section different CNN models that are used for remote sensing scene classification are reviewed.

3.1 VGG 16

VGG-16 model demonstrated an important improvement over the state-of-the-art setups by analyzing the networks and boosting the depth using an architecture with very small (3*3) convolution filters. The depth was increased to 16 weight layers, yielding around 138 trainable parameters and classifying 1000 images into 1000 categories. The number 16 in the network represents the weighted layers. It has 13 convolution layers, 5 maximum pooling layers, and 3 fully connected layers total of 21 layers build VGG16 architecture. But sixteen of them are weight layers, also known as learnable parameters layers. 224*224*3 is the size of input layer. Throughout the whole architecture, the convolutional and max pooling layers are ordered uniformly. In Convolutional layer-1 there are 64 filters, in Convolutional layer-2 128 filters, 256 filters in Convolutional layer-3, and in Conv-4 and Conv-5 there are 512 filters. This stack of convolutional layers are followed by 3 FC layers. 4096 channels are for the first two fully connected layers and the third connected layer is with 1000 channels. Figure 4 gives the architecture of the VGG16 model. [4]

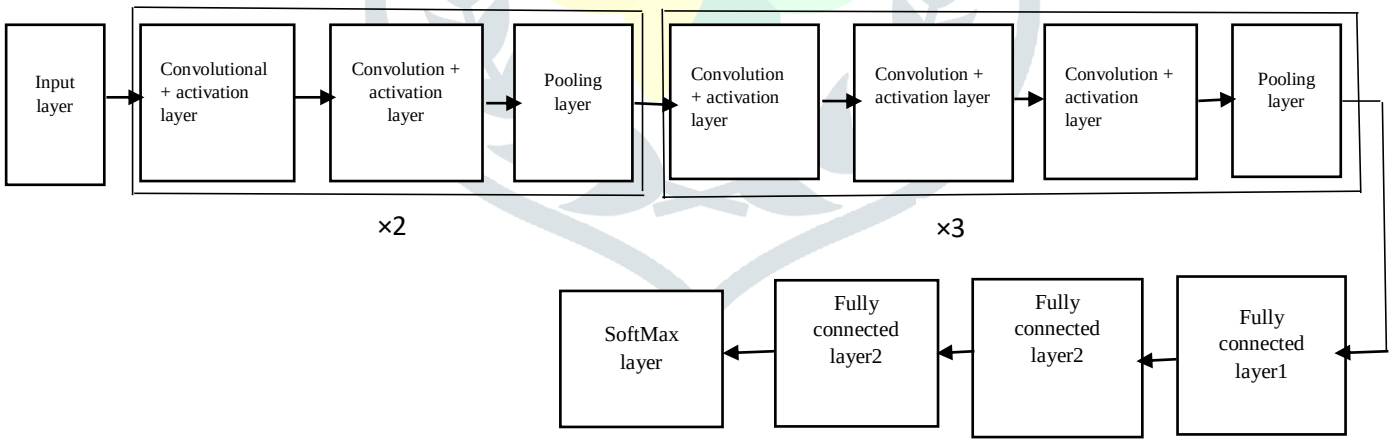

Figure 4: VGG-16 Architecture

3.2 GoogleNet

GoogLeNet is another breakthrough network in the convolutional neural networks that GoogleNet has 22 layers deep. Most of the issues that huge networks face is overfitting due to many deep layers. GoogleNet architecture with inception module solves this overfitting problem. The Inception module is a neural network design that makes use of feature identification at various sizes through convolutions with different filters while dimensional reduction is used to reduce the computational cost of training a large network. Total 9 inception modules are present in total 22 layers of the Google Net architecture (27 layers when pooling layers are included).

A technique known as global average pooling is employed in the GoogLeNet architecture. This layer averages a 7*7 feature map to a 1*1 size. Additionally, this reduces the number of trainable parameters to 0 and raises the accuracy. An inception network is a type of deep neural network with a repeating architectural structure called inception module. Architecture of the GoogLeNet is shown in figure 5. The inception module consists of various sizes of convolution filters like 1*1,3*3,5*5 and pooling layer. In the middle of the Inception architecture, there are a few intermediate classifier branches that are only used during training. These

branches include a SoftMax classification layer, two fully connected layers with 1024 outputs each, a 5*5 average pooling layer with a stride of 3, an 11 convolutional layer with 128 filters, and two layers with 1000 outputs each. [5]

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Figure 5: Architecture of the GoogleNet

3.3 EfficientNet-B0

ConvNets are scaled up furtherly to increase the accuracy of the network. Scaling ConvNets means scaling our model network either by depth, width, and resolution. EfficientNet-B0 performs a compound scaling which means all the scaling methods (depth, width, and resolution) are performed in the same network. In the compound scaling method, we use the compound coefficient ϕ for scaling the depth, resolution and width of a network.

$$\text{Depth} = \alpha^{\phi} \quad (1)$$

$$\text{Width} = \beta^{\phi} \quad (2)$$

$$\text{Resolution} = \gamma^{\phi} \quad (3)$$

Where α, β, γ are constants that can be obtained by a grid search. Φ can be the value that is chosen by the user based on the requirement [6]. This makes the EfficientNet differ from other ConvNets. A baseline network is very important while scaling, here we evaluate the existing ConvNets and developed a new mobile-size baseline for increasing the effectiveness of the network. This network is called EfficientNet and architecture is shown is table 1.

Table 1: Architecture of EfficientNet-B0 [7]

Input-resolution	Operator	t	c	n	s
224*224*3	Conv3*3	-	32	1	2
112*112*32	MBConv, k3*3	1	16	1	1
112*112*16	MBConv, k3*3	6	24	2	2
56*56*24	MBConv, k5*5	6	40	2	2
28*28*40	MBConv, k3*3	6	80	3	2
14*14*80	MBConv, k5*5	6	112	3	1
14*14*112	MBConv, k5*5	6	192	4	2
7*7*192	MBConv, k3*3	6	320	1	1
7*7*320	Conv 1*1 & pooling & FC	-	1280	1	1

In the Table **n** represent how many times the corresponding operator of each line repeats, **c** represents the number of output channels, **t** represents the expansion factor and **s** is the sliding step. These all parameters in the table combinedly give the overall detailed architecture of the EfficientNet-B0. [7]

3.4 Long Short Term Memory (LSTM)

LSTM network is a type of Recurrent neural networks (RNN) which has the feedback connections. These connections make the neural network to memories information for over many time steps. Because the LSTM unit specifically employs a vector memory cell for storing long term memory, it is able to retain the information for a long period of time more effectively and perform better than typical RNN. It is also a solution for the vanishing gradient problem in backpropagation for updating weights in the networks [8]. In this regard, the LSTM unit's status update can be described as (4). [9]

$$\text{LSTM: } h_{t-1}, m_{t-1}, x_t \rightarrow h_t, m_t \quad (4)$$

Where m_t and m_{t-1} represents the state of the cell at time t and t-1. Four neural networks, often known as cells, and different memory building elements make up the chain structure of the LSTM. Cells and gates both play a role in memory processing and retention of information. Moreover, the LSTM unit has the ability to choose which type of memories should be sent along and which one should be forgotten at each time step. Three gates are there in LSTM network. They are the output gate, the forget gate, and the input gate. These gates are used for memory manipulations and LSTM network structure is given in figure 6.

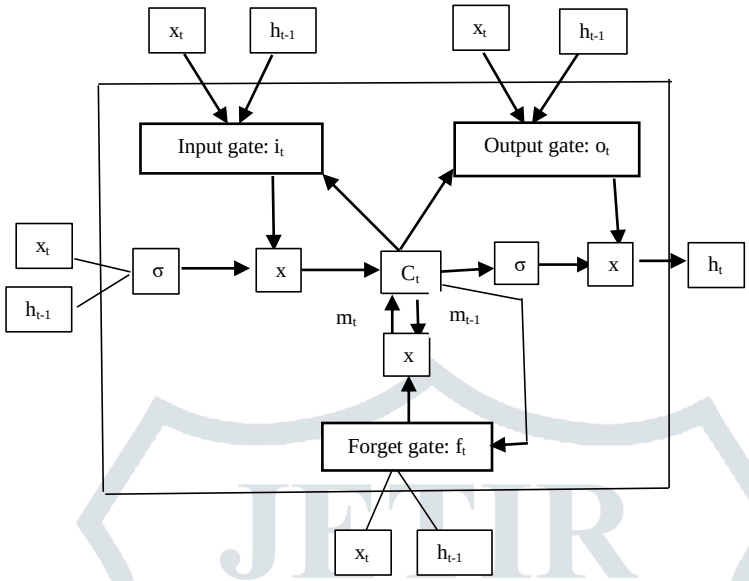

Figure 6: lstm structure [9]

A single floating-point value, m_t, is stored in memory by each LSTM unit. The value may be decreased or eliminated through a multiplicative interaction with the forget gate f_t or by combining the current input x_t with the activation of the input gate. The hyperbolic tangent nonlinearity transforms the stored memory ct and the output gate o_t controls the release of h_t.

4. UCMerced Land use dataset:

Yang and Newsam from the University of California Merced provide the UCM land use dataset, which contains the ground truth extracted from an available high resolution overhead image. It is reachable using the National map of U.S Geological Survey, and is processed by people from aerial photographs. There are 21 categories in the land use scene, including airplane, baseball court, beaches, buildings, chaparral, dense residence, agricultural land, forest, golf court, medium density residence, overpass, harbour, expressway, rivers, parking lot, runway, tennis court, intersection, storage tanks, mobile home park, and sparse residence. 100 images in each category with 256×256 in size. UCM is the most difficult remote sensing data set because of the exceptionally high levels of interclass similarity between classes.[7]

Figure 7: Image classes in UCM land-use data set

4.1 Performance metrics

Overall accuracy and confusion matrix are most widely used Performance metrics, which are applied to our data set.

1. Confusion matrix: confusion matrix is a representation of the classifier models predicted value and real value in a table-like structure, where all rows correspond to predicted value and all columns to real value. All values in the table correspond to the number of inputs that were given to the model for classification. It is a widespread procedure to graphically demonstrate how effectively supervised learning algorithms perform, particularly for classification problems. With the help of truth labels and associated predicted labels, the N*N confusion matrix for a set of data with N classes may be calculated and normalized. Every row and column in a matrix are thought of as categories fundamental truths. As a result, it might be considered a categorization result at the category level. [10]

2. Overall accuracy: An estimate of the model's performance across all classes is accuracy. It is determined using the reciprocal of the number of accurate predictions to all predictions. Accuracy is calculated as

$$\text{Accuracy} = (\text{True}_{\text{negative}} + \text{True}_{\text{positive}}) / (\text{False}_{\text{positive}} + \text{False}_{\text{negative}} + \text{True}_{\text{positive}} + \text{True}_{\text{negative}}) \quad (5)$$

5. Experimental Results

Experiments are done on UCM land use data sets using many different CNN networks like simple CNN, GoogleNet, EfficientNet, VGGNet, and CNN model with recurrent LSTM network. Our data set contains 256×256 size images which are not accepted by CNN pre-trained networks. Hence before giving our dataset images to the network, they have to be pre-processed accordingly to our networks. So, initially, input images are scaled to size 224*224. In all the models 80% of the dataset is given as training set and 20% as testing and validation sets. Next, the options given for training are the batch size is set to 64, 0.001 learning rate, optimization method in updating parameters is sgdm. 30 epochs are covered in training and this training process is conducted on Matlab R2021b.

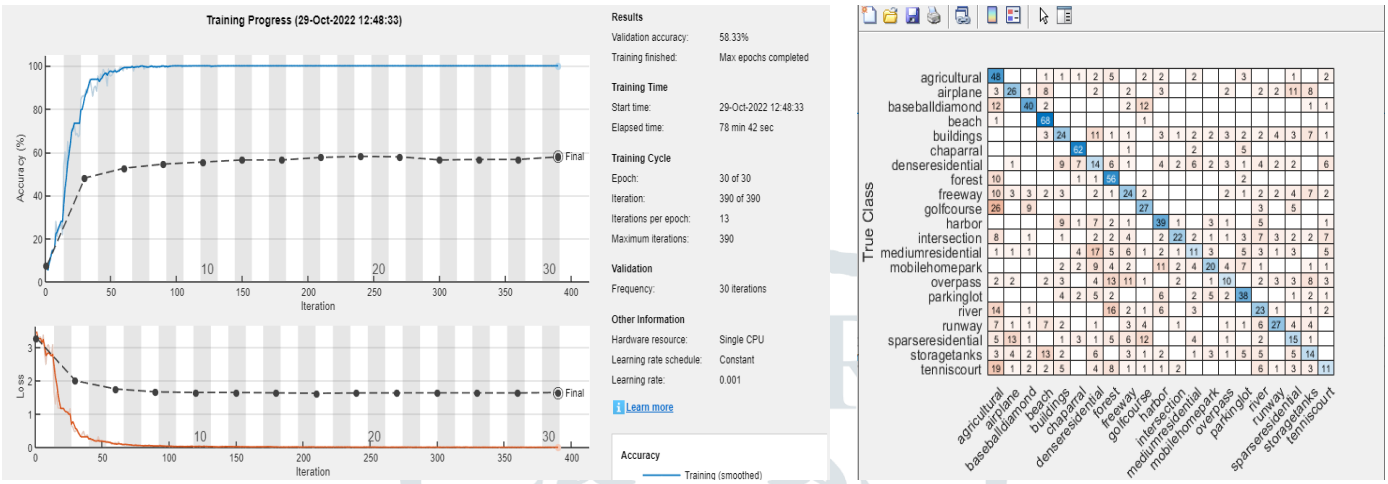

Figure 8: Training and confusion matrix for UCM data set using CNN network (overall accuracy =58.33%)

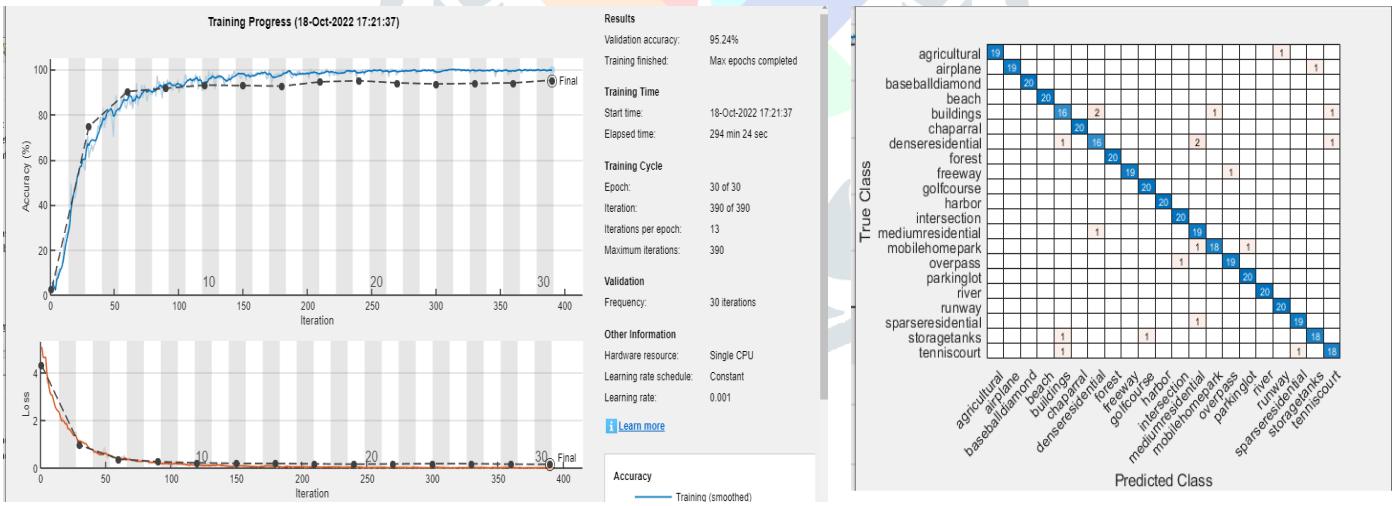

Figure 9: Training and confusion matrix for UCM data set using GoogleNet (overall accuracy = 95.24%)

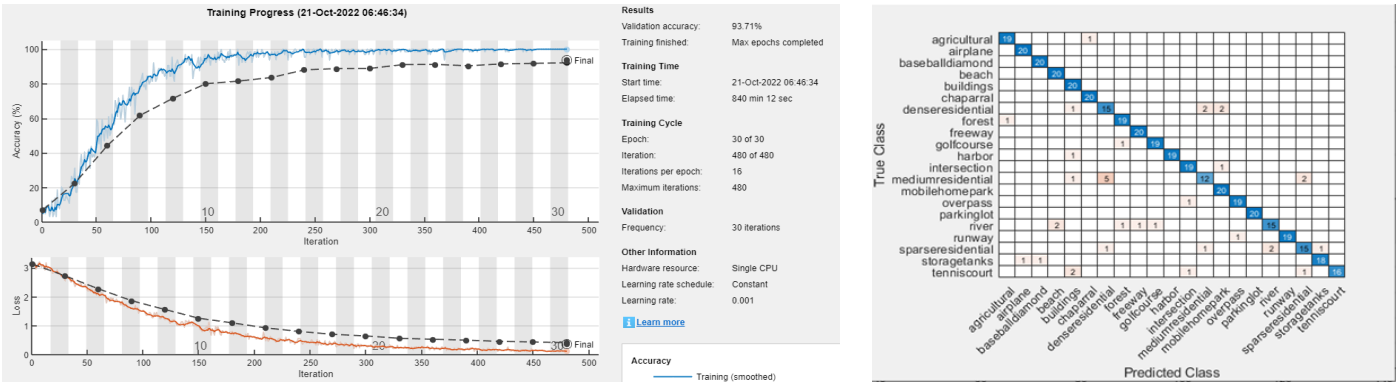

Figure 10: Training and confusion matrix for UCM data set using EfficientNet-B0 network (overall accuracy = 93.71%)

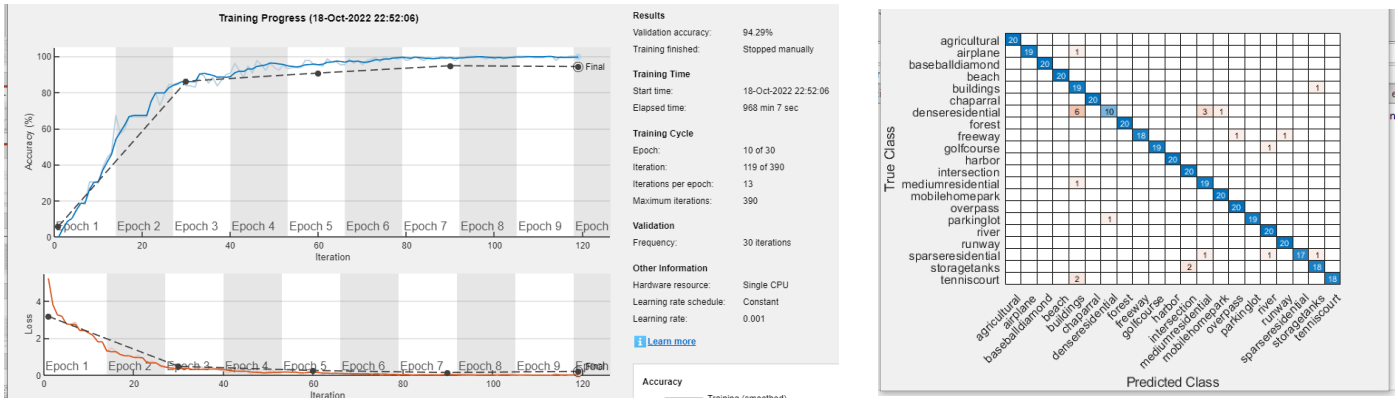

Figure: Training plot and confusion matrix for UCM data set using VGG-16 (overall accuracy = 94.29%)

Addition to the above experiments another two networks are trained on UCM data set with same training parameters. They are GoogleNet with recurrent LSTM network and EfficientNet-B0 with recurrent LSTM network and shows 93.10% and 91.19% accuracy respectively.

Table 2: Comparison Table

Networks used	Accuracy for 80% of training data
Convolution neural network (CNN)	58.33%
EfficientNet-B0	93.71%
VGG-16	**94.29%**
GoogleNet	**95.24%**
EfficientNet-B0 with recurrent LSTM Network	91.19%
GoogLeNet with recurrent LSTM Network	93.10%

Accuracies obtained for various CNN networks on UCMerced dataset are shown in table 2. From the observations GoogleNet and VGG-16 shown the best results that is highlighted in bold.

6. Conclusion

In the work, as deep learning is a wide spread technology, Deep learning architectures are studied and applied to an application remote sensing image scene classification. Deep learning models like Convolution neural networks (CNN), GoogLeNet, VGG16, EfficientNet-B0, and CNN model with recurrent Long short term memory (LSTM) network are applied to our UCMerced land use data set. After the tests done on various models with our training parameters, the results have shown that GoogleNet and VGG16 gave the best performance with 95.24% and 94.29% accuracy respectively.

REFERENCES

1. Rafael Pires de Lima, Kurt Marfurt.: Convolutional Neural Network for Remote-Sensing Scene Classification: Transfer Learning Analysis. MDPI 2020, 12, 86; doi:10.3390/rs12010086.
2. Chaib, S., et al.: Deep feature fusion for VHR remote sensing scene classification. IEEE Trans. Geosci. Remote Sens. 55(8), 4775–4784 (2017).
3. Review of Image Classification Algorithms Based on Convolutional Neural Networks Leiyu Chen 1, Shaobo Li 1,2, *, Qiang Bai 1, Jing Yang 1,2, Sanlong Jiang 1 and Yanming Miao 3.
4. Everything you need to know about VGG1 Author: Rohini G. https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918.
5. Going Deeper with Convolutions ,Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. https://doi.org/10.48550/arXiv.1409.4842.
6. Scene Classification with Recurrent Attention of VHR Remote Sensing Images Qi Wang, Senior Member, IEEE, Shaoteng Liu, Jocelyn Chanussot, Fellow, IEEE, and Xuelong Li, Fellow, IEEE.
7. Le Liang, Guoli Wang.: Efficient recurrent attention network for remote sensing scene classification. IET Image Processing 15(8), 1712-1721(2021)
8. Deep Learning | Introduction to Long Short Term Memory. https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/
9. Tan, M., Le, Q.V,: "Efficientnet: Rethinking model scaling for convolutional neural networks". ICML 2019, 9–15 June, Long Beach, California, USA, pp. 6105–6114. Vol. 97 of Proceedings of ML Research, PMLR (2019).
10. Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall By Ahmed Gad, KDnuggets Contributor on February 19, 2021 in Accuracy, Confusion Matrix, Deep Learning, Metrics, Precision, Recall.