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

Analysis of Deep Learning Architectures in Deforestation Detection

[1]Anusha R Shenoy, [2]Mahathi N, [3]Ranjitha L, [4]Impu, [5]Swetha P M

[1]Student, [2]Student, [3]Student, [4]Student, [5]Assistant Professor
[1]Computer Science and Engineering,
[1]JSS Science and Technology University, Mysore, India

Abstract: The Amazon ecological system has been studied several times over the years, and possibilities for such research have only increased since the prospect of utilising satellite imagery for the purpose opened up. One of the major concerns here is the alarming rate at which the forests in the rainforest is being chopped off, even to convert the land into other commercially profiting establishments such as farms and plantations. With the advent of deep neural networks and transfer learning, breaking down images taken even from great altitudes to be able to decipher the change in landforms from one piece of land to another became a credible solution to spot deforestation through an entirely automated process. A reliable and acceptably accurate Convolutional Neural Network (CNN) should not only enable the administration to nab perpetrators, warn the general public of the impending nature crisis that the world is headed towards and distinguish what the newly deforested regions are being used for - coffee plantations, roads, human settlements, or date palm plantations as the commonly observed patterns suggest. A successful solution of this nature can then be replicated for natural landscapes of all kinds that are being subjected to exploitation across the world.

***Index Terms* – Deep Learning Architecture, Amazon Rainforest, Deforestation Detection, Convolutional Neural Network**

I. INTRODUCTION

The Amazon rainforest, rightly titled 'the lungs of the earth', is the world’s largest tropical rainforest and carbon sink, famed for its rich diversity of flora and fauna and its role in fighting climate change. However, it has been established by a number of modelling studies that today, the Amazon may have two “tipping points”, namely, a temperature increase of 4°C or deforestation exceeding 40% of the forest area.[1] The Amazon sequesters an excess of two billion tonnes or 5% of the annual emissions of carbon out of the air. Hence it becomes precarious to take immediate action to protect this precious resource, in turn ensuring a better prospect of life on Earth for humanity itself.

Deforestation has been one of the major causes of the climate crisis that the Earth has been battling for years now. A number of technological innovations and ecological preservation mechanisms have been suggested and taken up by people and organizations across the world. Favourably, it has been observed by recent studies that satellite imagery being captured for other scientific research can be effectively utilised to detect deforested regions in dense forests[2] such as the Amazon where it is humanly impossible to manually inspect each unit of area. Such images have to be processed using image processing and deep learning techniques - perhaps enhanced by transfer learning - to automate the inspection process. Harnessing the power of artificial intelligence, it can be expected to have a better view of the human enterprises being carried out within the Amazon, thereby enabling an improvement in regulations accordingly through images captured by satellites and common computer vision applications. [3]

A convolutional neural network uses a three-dimensional neural network to process the red, blue and green channels of the image at the same time.[4] This provides the major advantage of considerably reducing the number of neurons required in comparison to traditional feed-forward neural networks. The network receives images as an input and uses them to train a classifier. Although it is infeasible to simply employ someone to continuously analyse satellite imagery, the deployment of a well-trained deep neural network can automate this entire process. Such a modification would also serve to replace the currently applied methodologies that involve a lot of manual operations. CNN architectures have been found by several studies to be extremely effective in learning similarity functions between image pairs[5] that implicitly suffered some transformations and other kinds of effects such as rotation, translation and illumination. The common observation has been that transfer learning provided much better results than models that had been specially hand-crafted to suit this application.

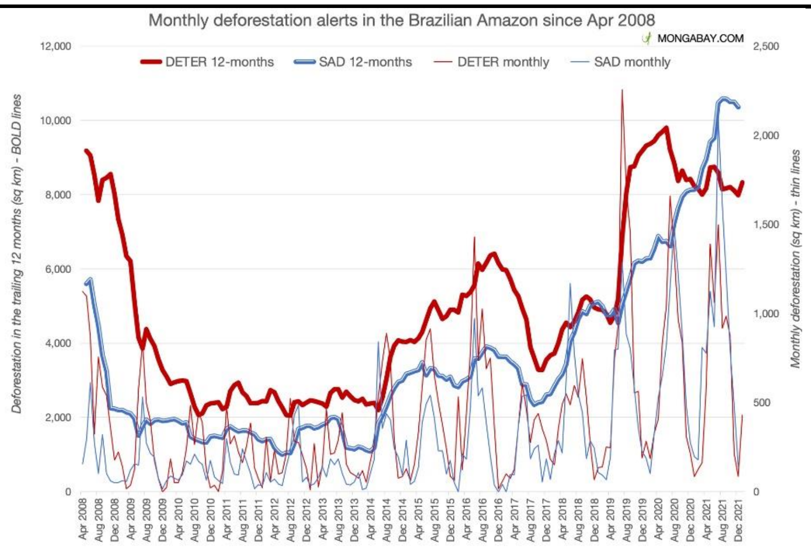

Fig. 1 Deforestation in the Amazon rainforest, Brazil region from 2008 to 2021 (source: Mongabay News)

A number of researchers have proposed the development of a tool to detect deforestation in the rainforest on a day-to-day basis using satellite images from the MODIS/TERRA sensor[6] by applying Artificial Neural Networks and U-net architectures. They mostly conducted spectrum-temporal natured studies on satellite images. Maps of deforestation hotspots were generated in the Venezuelan forests using these very images, attaining an accuracy of 92.5% to analyse vegetation covers over a span of 5 years.[7] Prominent accomplishments in the area were also marked by Multitemporal Deforestation Detection[8] using reflectance values on the original image and target image and combining NDVI and dNBR in the algorithm for improved accuracy on bigger datasets, and processing of high-resolution remote sensing data from the Kompsat-3[9] satellite achieving an accuracy of 98.4%. NDVI values were also applied to track deforestation in the Yavatmal district of Maharashtra state, computing the parameter separately for each pixel.[10] Fused optical and radar images were trained on a U-net architecture model as well but faced challenges from the multidimensionality and complexity of data.[11]

Getting unbiased, global insights on a frequent basis might be crucial to encourage the fight against climate change. Hence, the project will be built with the intention to automate the monitoring of human activity and operations in the Amazon rainforest by training multiple deep convolutional neural networks and analysing them for performance. A simple and straightforward Convolutional Neural Network, models with the exact same structures as the VGG16[12] and VGG19 models have been built, and a pre-trained ResNet50 model[13] has been chosen to perform in-depth performance analysis. The utilisation of a novel FastAI[14] approach is also attempted here, in comparison with the conventional TensorFlow classes and functions, to make the best out of the limited computational resources offered by Google's Colaboratory.[15] The best performing among the experimented models will be made accessible for easy use and exploration through a user-friendly web application interface, which would also raise awareness among people about the global climate crisis.

II. DATA AND STUDY AREA

A majority of the studies attempting to use satellite data for detecting deforestation use the 'Planets: Understanding the Amazon from Space' dataset to train their deep learning models. It has been derived from the Planets 4-band satellites' full-frame analytic scene products, while in a sun-synchronous orbit (SSO) and International Space Station (ISS) orbit. Hence the images have four bands of data each, viz. red, green, blue and near-infrared. GeoTIFF data may also be captured. The dataset reasonably represents the places of interest in the Amazon basin. It has been observed that the label primary is associated with almost all of the 17 classes, implying that most images have some degree of primary forests along with other labels. The label agriculture has been associated with a few labels like road, habitation and cultivation by the Planets Impact team.

The Planets dataset contains over 20,000 images with a ground-sample distance (GSD) of 3.7 m and an orthorectified pixel size of 3 m. The data was collected from satellites between January 1, 2016, and February 1, 2017. It is a multilabel dataset, and an image can belong to up to 9 classes as found from exploratory data analysis. The study aims to compare and analyse the performance of several Convolutional Neural Networks on this dataset from the Amazon basin, and to try and account for their behaviours, ultimately leading to the model that suits the application best and can be worked upon that would enable automation of the entire deforestation detection process in the future.

III. RESEARCH METHODOLOGY

The study is proposed to be conducted on a variety of Deep Neural Networks, starting with a very basic Convolutional Neural Network before heading on to more complex ones, and finally using a pre-trained network to analyse the performance improvement attained by transfer learning[16]. The dataset described previously was first subjected to Exploratory Data Analysis to get a first-hand understanding of the images in it and their classes. The implementation strategies and characteristic models experimented with are described below.

Exploratory Data Analysis

With the objective of analyzing and investigating the data, to summarize its main characteristics, data visualization and image processing techniques were applied on the dataset. It was found that the label 'Primary' is associated with most images, implying that a majority of the images have detectable spreads of forest cover, along with possible matches for features of other labels too. It was also observed that the label 'Agriculture' matched a couple of other labels such as habitation. It was a conscious decision, in accordance with previous studies[17], to not include habitation as a deforestation indication, since these regions could be villages or other domains occupied by people from several years ago.

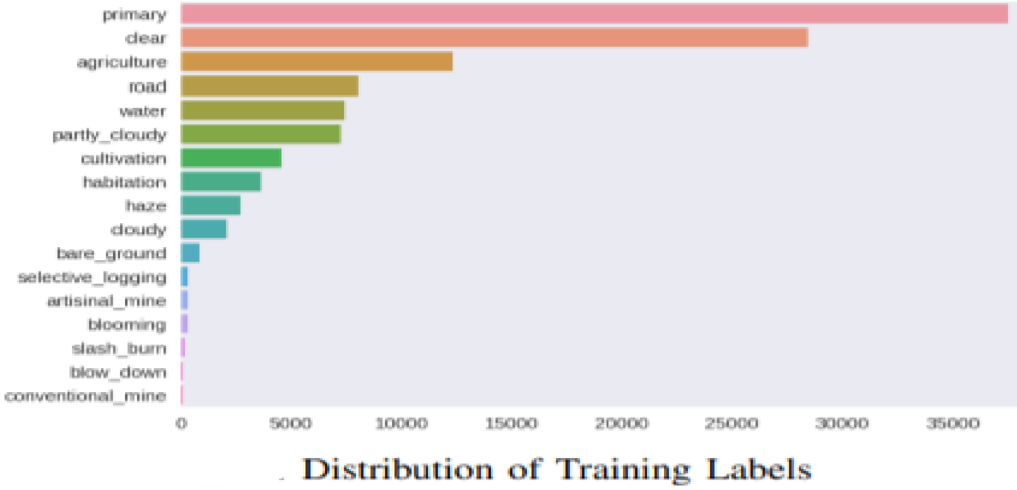

Fig. 2 Number of samples belonging to each class as found by Exploratory Data Analysis

Deep Neural Networks

The study aims to analyse the performance of different kinds of Convolutional Neural Networks (CNNs)[18] on the dataset, with the same parameters, and attempts to best decipher the cause for the observed behaviour. CNNs work on the principle of applying 2D convolution kernels or filters of different sizes and values (or weights) in parallel on an image matrix to detect the specific features that the filter is designed for. The outcome of applying these filters is a set of matrices – one each for a filter. This outcome is fed to an activation layer, the output of which is input to a pooling layer, for reducing the size of the matrices, thereby also achieving the objective of making the obtained features deeper. These layers are repeated for a different number of times depending on the number of features that the model must learn for the application in question. It is commonly observed that a pooling layer may follow a set of two or more convolutional layers. The aforementioned layers are usually partially connected, and a fully connected artificial neural network is mounted on top of this setup for the purpose of classification. The model is trained in a similar fashion as a generic neural network, by the strategy of back propagation.

The choice of CNN for the application is justified since each standard image typically has hundreds of pixels and features, and a fully connected neural network would imply huge expenditure of resources on computations and weight adjustment. CNN overcomes this issue by using partially connected layers, drawing inspiration from the working of the visual cortex in the brain.

The convolutional layer in combination with the activation function, usually the Rectified Linear Unit (ReLU)[18], is typically responsible for extracting the most important feature or the feature looked for by the filter in a small region of the image matrix. The activation function is responsible for bringing non-linearity to the model. The ReLU function promotes all negative input values to 0 and retains all other values as such.

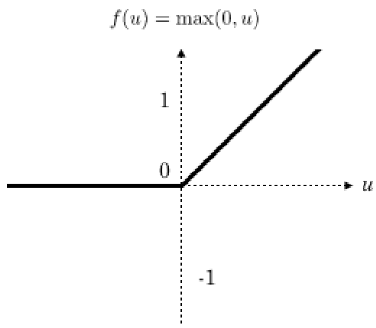

Fig. 3 The graph for the ReLU activation function

The filter then moves horizontally by a fixed number of steps called stride. Since the convolution operation produces one single value for each time the filter aligns with the image matrix after the jump specified, a bigger stride length reduces the dimension of the image by a greater proportion than a smaller stride length does. To avoid or bring down such a reduction, as well as to ensure that the information in the edges is not lost due to being involved in very few convolutions, zero padding is used around the image matrix. This is achieved by inserting additional rows and columns of 0’s to the left and right of, above and below the matrix being processed. The number of rows and columns required depends upon the filter size and stride length.

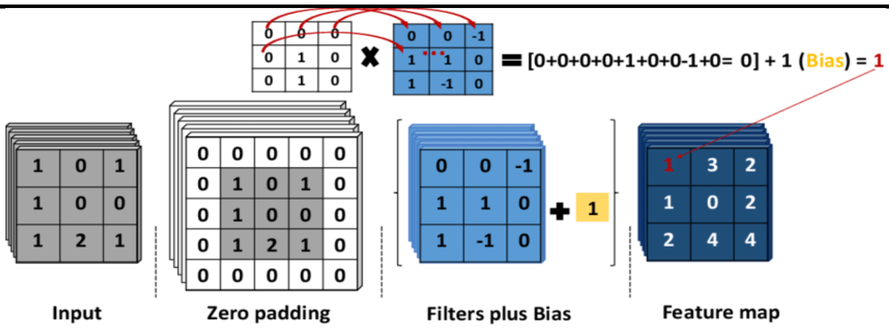

Fig. 4 Zero padding to retain the size of the image matrix after convolution[19]

The outcome of the convolutional-pooling sets of layers in the network is a set of matrices, each obtained by applying different filters and then reducing the dimension, then repeating this process again. The fully connected neural network at the end, however, expects a single dimensional vector as its input. To make the matrices obtained from the previous layers compatible with this, the process of flattening[20] is applied, which converts all the matrices into a single, long line of values that will finally be used for classification.

The fully-connected Artificial Neural Network following the feature extraction section[21] of the CNN described above is responsible for classifying the image sample into one class – in case of single class classification problem, or all those classes that the image fits into – in case of a multiclass classification[22] problem such as this one. The activation function in the uppermost layer of this network is a sigmoid function for binary classification, and a softmax function[23] if there are more than two classes of data samples. Models that were designed specifically for the ImageNet dataset such as the VGG16, VGG19, AlexNet, Inception, MobileNet, ResNet – some of which are explored in this study, have a softmax activation function with 1000 neurons in the last layer, one each for the 1000 classes of images in the ImageNet dataset. In this problem, the dataset being dealt with has 17 classes, and hence the ultimate layer of the deep neural network has 17 neurons, and is followed by a softmax activation function. The softmax activation function assigns labels to a data sample on the basis of probability values for it belonging to each class. The class for which the probability value is the highest, or ones for which it is above a specific threshold value will be assigned to the data sample. This probability is computed as follows:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (1)$$

Here, σ represents the sigmoid function, and the probability that a data sample belongs to a class i where $1 \leq i \leq K$ for a dataset with samples belonging to K different classes is given by the ratio of the standard exponential function of the input vector z_i to the summation of the standard exponential functions of output vectors z_j.

The models are trained using mini-batch gradient descent[24] strategy since it would be infeasible to pass the entire dataset through the network for every epoch – the limited resources available to this research team would not support it. A batch size of 100 was settled upon after multiple trials. Accordingly, a set of 100 satellite images would be passed through the network, predictions would be made based on arbitrarily assigned weights, or those obtained from the pre-trained models, and the errors computed separately for each would be stored. At the end of these 100 loss computations, the weights would be modified by deducting or incrementing the existing weights by a partial derivative of the loss with respect to the weight being recomputed. This weight modification is done from the upper layers down to the lower layers. In case of models where transfer learning is applied, the weights in the lower layers would not be changed. Only those in the upper layers would be modified by this logic. This process completes one iteration. Once all the batches have been passed through the network and weights modified accordingly, one epoch is said to be completed. Most popular models are trained for a huge number of epochs, but since this study was limited by the availability of resources, the maximum number of epochs attained for any model was 10.

The dataset was split as 60 – 20 – 20 for training, testing and validation. Training is the process wherein the deep learning model 'learns'. The data is passed through the network along with the labels. The model predicts, computes the error and adjusts the weights. After every epoch, the model is subjected to a process called validation, during which it is exposed to a section of the dataset that it has never seen before. This gives the developers an idea of how well the model will perform on new images. Validation may also be used as an indication to stop training when the model has achieved scores high enough. Testing is conducted once the model is completely trained and there will no longer be any kind of modification to its weights. This is an indication of the quality of the training process on the fully trained model.

The experimentation in the scope of this study began with a simple CNN of just 10 layers, putting together four sets of convolutional-pooling layer sets piece by piece using the Sequential() API. The number of layers were then gradually increased, making observations on the achieved results along the way. Models from the AlexNet contest – VGG16 and VGG19 – were given special attention in an attempt to understand what actually made them shine in the contest, and if at all they would display the same kind of efficiency on the dataset being considered.

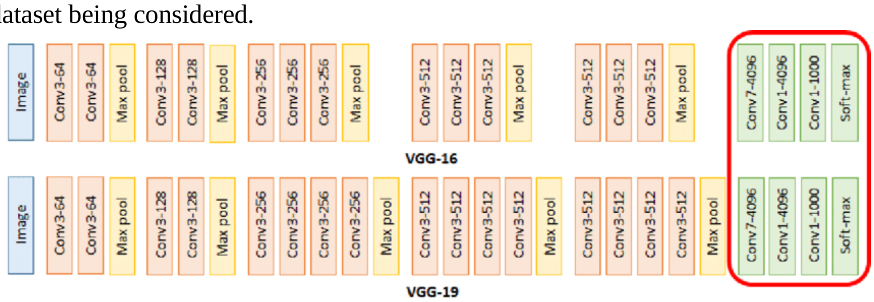

Fig. 5 Structure of VGG16 and VGG19 models [25]

Once the performance of these generic CNNs did not show much improvement on increasing the number of layers, the focus was shifted to the pre-trained models. Transfer learning has emerged as a reliable technique to train deep neural networks for high performances with considerably less requirement of resources, time and training data. In this strategy, well performing model that has been trained on a huge generic dataset is saved along with all its weights, so that it can later be imported by a developer working on a similar application, and trained only for the upper layers by freezing the lower layers. This allows the model to learn the specific features of the dataset in question and to mold it as per the output requirements of the new application. Given that several researchers have found that the ResNet model is faster and better than most other ImageNet models, the pre-trained ResNet50 model was emphasized upon to explore how the slightly different identity mapping strategy used in it would benefit the model's performance.

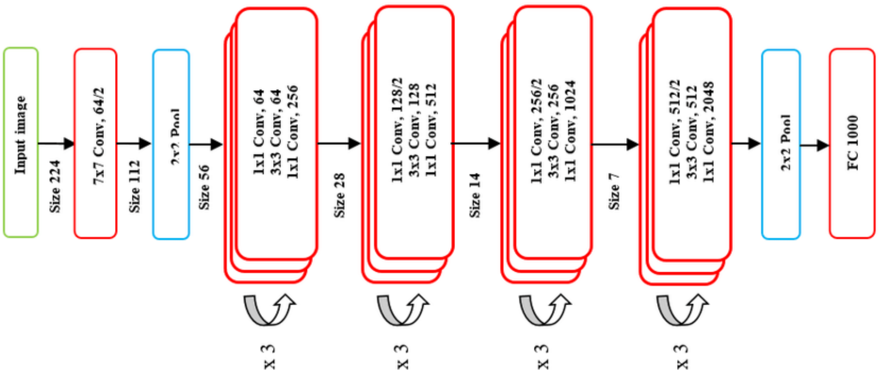

Fig. 6 Structure of the original ResNet50 model [26]

These models were initially trained using the TensorFlow library for its extremely convenient to use and also immensely popularly classes and APIs including Keras, which is the conventional method followed by most deep learning applications. Eventually, we also attempted the model building and training using FastAI. The difference that was obvious right off the bat was that FastAI did not provide the free hand to customize the models as TensorFlow did. However, the trials were carried out since it is known to make the process much simpler and faster.

Performance Metrics

The performance of the deep learning models is evaluated in terms of the F-β score[27] where β=2, giving the F2 score, a metric that combines precision and recall, giving the latter a greater weightage. This implies a data sample is more likely to be categorized as one in which deforestation is detected although that isn't the case, rather than it being written off as a piece of land where no deforestation is detected while in reality it is subjected to deforestation. This kind of a metric is highly favourable to the application, and for a model to score well in these circumstances, it should have fewer false negatives. The F2 score is also the preferred metric to evaluate models trained on this dataset by most researchers experimenting on the same, hence giving us benchmarks to compare our models against. Besides, as found during the course of this study, the other most popular metric – accuracy – is not as effective on the dataset and does not always increase when the overall loss for the network decreases.

The F-β score is computed as follows:

$$F - \beta\ score = (1 + \beta^2)\frac{pr}{\beta^2 p + r} \qquad (2)$$

Here, p stands for precision, r stands for recall, and are computed as:

$$p = \frac{TP}{TP+FP} \qquad (3)$$

$$r = \frac{TP}{TP+FN} \qquad (4)$$

The notation TP is used to indicate True Positives – the number of True Positives predicted by the model, FP indicates False Positives – the number of samples labelled 'Positive' by the model although in reality they belong to the 'Negative' class, and FN indicates False Negatives – the number of samples that have been labelled 'Negative' by the model although they belong to the 'Positive' class. Precision can thereby be viewed as a ratio of the correct number of 'Positive' predictions to the total number of 'Positive' predictions, while Recall may be defined as the ratio of the correct number of 'Positive' predictions to the total number of 'Positive' labelled samples present in the dataset. Hence, the equation for F2 score becomes:

$$F2\ score = 5\frac{pr}{4p+r} \qquad (5)$$

Since it is always better to label an image with no deforestation as a deforested region, rather than predicting for a land that has been wiped clean of its green cover as a region unaffected by deforestation, the preference given to the F2 score is justified.

Optimization Techniques

The images in the dataset and the huge number of neurons in the models lead to the extraction of an overabundant number of features, which may lead to the problem of overfitting. An overfitted model has learnt the training dataset too well, because of which, although it may perform exceedingly well during training, it fails during testing. To avoid this, the technique of regularization is widely used. This study applies the regularization technique called dropout[28]. Here, during training, some neurons are randomly turned inactive for a few epochs. This process is controlled by a hyperparameter termed dropout probability. This value is commonly set to 0.5. Applying the dropout strategy creates an illusion of a single layer appearing like multiple different ones each time a neuron in it is made inactive. Thus, the model is less likely to get overfitted.

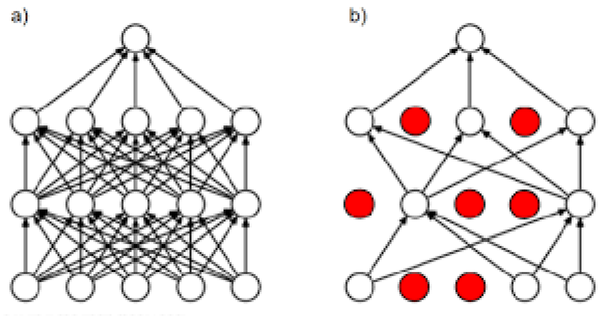

Fig. 7 Dropping some neurons to avoid overfitting[29]

IV. Results and Discussion

The general observation is that the pre-trained models perform much better than the traditional CNNs that were built from the scratch, just as expected. This behaviour, as is made pretty obvious by the fact that deep neural networks thrive on training data, can be attributed to the huge bulk of the training samples from the ImageNet dataset that the original models were initially trained on, and the certainty that heavy computational resources would have been used to train those models for multiple epochs already. Since the lower layers are known to learn only the very basic features of an image – borders, lines at angles, colours, contours, etc., the pre-trained models provide the obvious upper hand in learning the satellite imagery where the possibility of deforestation and the purpose for which any deforested land is being used is majorly dependent on these primary features of the image. The performances of the two kinds of models were sufficiently far apart.

Deep Learning Libraries and Classes

The findings with respect to the library used to develop and train the model aligned along the presumptions – the FastAI library proves itself to be very useful, especially in transfer learning cases, due to the high speed at which the models completed training and the substantial reduction in the size of the code to be written. Each epoch only took up to four and a half minutes to be completed, which is an incredible improvement from the TensorFlow variations, which with the exact same structures and parameters took around three and a half hours to complete a single epoch. The TensorFlow library, on the other hand, provides classes and APIs which have high degrees of customization. Right from the size of the convolution kernel, the stride length and the activation function, every single parameter can be adjusted as per the requirement of the application, making way for a much greater horizon of research. Its Sequential() API gives the freedom to also include dropout regularization wherever necessary, thereby providing better control over the model.

The choice of the library hence is a delicate tradeoff between the speed at which the model needs to be trained, the quantity of resources available for the model, ease of use in terms of lesser code on one hand, and the benefit of multiple resources already existing for any setback, the degree of adaptability the model must have and the self-explanatory nature of the functions on the other hand. This varies from one application to another, and since in this case, the onus was on studying the behaviour of different models on a single dataset rather than a single model on a single dataset but with varying parameters, the FastAI library emerged as the better choice.

Traditional CNN Models

The simple Convolutional Neural Networks that this study stresses upon – the CNN with 10 layers, the CNN resembling VGG16 and the one resembling VGG19 had several disadvantages. These had to be trained from the scratch, with absolutely no prior knowledge of the weights to begin with, which simply implied that the training process was extremely tedious, both in terms of the resources consumed and the time taken for training. This was more problematic than initially anticipated, since carrying out such intensive processes on PaaS resources put a limit on the maximum duration for which the temporarily allocated resources could be used in one stretch, and any disruption meant having to start over from the very beginning.

Performance-wise, it was found that the scores improved steadily as the number of layers in the model increased. This is because a small model – say the one with 10 convolutional layers – would not have the capacity to process the huge number of features that would be extracted from a dataset of so many images. The ImageNet contest winning VGG16 model's replica obviously performed better, since it has 6 more convolutional layers, but the results were nowhere close to the leading F2 scores on the Kaggle page for the dataset. Another consistent increase was observed with the introduction of the VGG19-like model. However, it was still much lesser than what can be accepted as a decent outcome from a deep learning model.

Another very interesting – although peculiar – observation made while training these models was that the accuracy in some cases remained very low even after a couple of epochs had been trained for, and it did not show any kind of relationship with the loss computed, must against the expectation that a decrease in loss should have brought about an increase in both accuracy and F2 score. Although the F2 score did increase, the accuracy did not.

Table 1 Performance of traditional CNN models

Models and their scores	**CNN Model**		
	CNN with 10 layers	**VGG16-like model**	**VGG19-like model**
Loss	0.2536	0.2312	0.2267
Accuracy	0.4280	0.9262	0.1447
F2 score	0.5671	0.5996	0.6188

Model with Transfer Learning

The ResNet50 model used as a pre-trained model which was later retrained on the satellite imagery dataset from the Amazon held a great amount of interest in this study. The ResNet50 model, when imported, came with weights that had achieved an accuracy of 81% with 30 epochs on the ImageNet dataset. This model was freezed as is and trained for another four epochs, at the end of which the accuracy and F2 score both had crossed the value of 0.9. Later the upper layers of the model were trained and the weights were modified to suit this application.

epoch	train_loss	valid_loss	accuracy_multi	fbeta_score
0	0.299750	0.132463	0.936083	0.889056
1	0.179756	0.118149	0.942063	0.896491
2	0.159194	0.104181	0.943436	0.911254
3	0.152496	0.101443	0.948719	0.913187

epoch	train_loss	valid_loss	accuracy_multi	fbeta_score
0	0.146750	0.105480	0.951015	0.913723
1	0.148924	0.107337	0.934731	0.911124
2	0.142897	0.096772	0.952004	0.913915
3	0.139369	0.090828	0.953610	0.921393
4	0.133889	0.086599	0.953959	0.924952
5	0.129884	0.085913	0.955913	0.925960

Fig. 8 Performance of model developed from the pre-trained ResNet50 model

From figure 8, it is evident that when the model started retraining the modifiable weights on the dataset, the accuracy dropped for a bit before it picked up. This could be attributed to the weight adjustment having gone a little off the track at that point, before the values increase again. The loss decreases consistently during both training and validation, while both accuracy and F2 scores increase. The time consumed for each epoch too was continuously under five minutes. These results were obtained on training with the FastAI library, thereby reinstating with certainty that pre-trained models perform better than traditional CNNs, and that the FastAI functions proved very convenient during the course of this study.

V. ACKNOWLEDGMENT

The authors would like to thank the immense contributions of the faculty of the Department of Computer Science and Engineering, JSS Science and Technology University, Mysore, India for furnishing several necessary inputs and helping steer the course of the study throughout.

REFERENCES

[1] C A Nobre, 2016. Land-use and climate change risks in the Amazon and the need of a novel sustainable development paradigm from Proceedings of the National Academy of Sciences**,** 113(39): 10759-10768

[2] Rhyma Purnamasayangsukasih P, Norizah K, Ismail Adnan A M, and Shamsudin I, 2016. A review of uses of satellite imagery in monitoring mangrove forests from 8th IGRSM International Conference and Exhibition on Remote Sensing & GIS**,** IOP Conf. Series: Earth and Environmental Science, 37 (2016)**:** 012034

[3] Caleb Robinson, Anthony Ortiz, Juan Miguel Lavista Ferres and Daniel E. Ho, 2021. Deep learning with satellite imagery to enhance environmental enforcement from Data Science Applied to Sustainability Analysis (pp.205-228)

[4] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do and Kaori Togashi, 2018. Insights into Imaging volume 9, pages 611–629 (2018)

[5] Srikar Appalaraju and Vineet Chaoji, 2017. Image similarity using Deep CNN and Curriculum Learning from Oral talk in Artificial Intelligence, Data mining and Machine learning track: GHCI 17

[6] T. Kehl, V. Todt, M. Veronez, S. Cazella, 2015. Real time deforestation detection using ANN and Satellite images from SpringerBriefs in Computer Science, 2015, IV, 58 p.

[7] Carlos Eduardo PachecoMaría Inmaculada AguadoDanilo Mollicone, 2014. Identification and characterization of deforestation hot spots in Venezuela using MODIS satellite images from Acta Amazonica, 44(2):185-196

[8] Danang Surya Candra, 2020. Deforestation detection using multitemporal satellite images from IOP Conference Series Earth and Environmental Science 500(1):012037

[9] Lee, Seong-Hyeok, Kuk-Jin Han, Kwon Lee, Kwang-Jae Lee, Kwan-Young Oh, and Moung-Jin Lee. 2020. "Classification of Landscape Affected by Deforestation Using High-Resolution Remote Sensing Data and Deep-Learning Techniques" Remote Sensing 12, no. 20: 3372

[10] Indersheel Kaur H. S. Nanare, Dinesh B. Bhoyar, Sanjay V. Balamwar, 2021. Remote Sensing Satellite Image Analysis for Deforestation in Yavatmal District, Maharashtra, India from 3rd International Conference on Signal Processing and Communication (ICPSC), 2021, pp. 684-688

[11] Julius Akinyemi, Josiane Mothe, Nathalie Neptune, 2018. Deforestation detection on satellite imagery with deep learning from HAL archives, version 1 ⟨hal-03044231⟩

[12] Karen Simonyan, Andrew Zisserman, 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition from ICLR 2015, arxiv-1409.1556v6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015. Deep Residual Learning for Image Recognition from ICLR 2015, arXiv:1512.03385v1

[14] Jeremy Howard, Sylvain Gugger, 2020. fastai: A Layered API for Deep Learning from Information 2020, 11(2), 108

[15] Tiago Carneiro Pessoa, Raul Medeiros, Thiago Neopmucino and Gui-Bin Bian, 2018. Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications, IEEE Access. PP(99): 1-1

[16] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, Qing He, 2019. A Comprehensive Survey on Transfer Learning, arXiv:1911.02685

[17] Giri Tejaswi, 2007. Manual on Deforestation, Degradation, and Fragmentation using Remote Sensing and GIS, Strengthening Monitoring, Assessment And Reporting On Sustainable Forest Management In Asia (Gcp/Int/988/Jpn)

[18] Abien Fred M. Agarap, 2019. Deep Learning using Rectified Linear Units (ReLU), arXiv-1803.08375

[19] Serveh Kamrava, Pejman Tahmasebi, Muhammad Sahimi, 2020. Linking Morphology of Porous Media To Their Macroscopic Permeability by Deep Learning from Transport in Porous Media, 10.1007/s11242-019-01352-5

[20] Jonghoon Jin, Aysegul Dundar & Eugenio Culurciello, 2015. Flattened Convolutional Neural Networks for Feedforward Acceleration from ICLR 2015, 10.1007/s11242-019-01352-5

[21] S. Albawi, T. A. Mohammed and S. Al-Zawi, 2017. Understanding of a convolutional neural network from International Conference on Engineering and Technology (ICET), pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186

[22] Baidaa M Alsafy, Zahoor M. Aydam, Wamidh K. Mutlag, 2019. Multiclass Classification Methods: A Review from International Journal of Advanced Engineering Technology and Innovative Science, Volume 5, Issue 3, Page No: 01-10

[23] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall, 2018. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning, arXiv-1811.03378

[24] Mu Li, Tong Zhang, Yuqiang Chen, Alexander J. Smola, 2014. Efficient mini-batch training for stochastic optimization, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 10.1145/2623330.2623612

[25] Ghada A. Shaheed, Mohammed Ali Tawfeeq, Sawsan M. Mahmoud, 2020. Automatic Medical Images Segmentation Based on Deep Learning Networks from IOP Conference Series Materials Science and Engineering. 870(1): 012117

[26] Bendjillali Ridha Ilyas, Moh13 Beladgham, Khaled Merit, Abdelmalik taleb-ahmed, 2019. Illumination – robust face recognition based on deep convolutional neural networks architectures from Indonesian Journal of Electrical Engineering and Computer Science. Vol 18(No 2): 1015~1027

[27] Cyril Goutte and Eric Gaussier, 2005. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation from Proceedings of the 27th European conference on Advances in Information Retrieval Research 3408:345-359

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. JMLR 15(56):1929−1958, 2014

[29] Mareck Florkowski, 2020. Classification of Partial Discharge Images Using Deep Convolutional Neural Networks. Energies 13(5496):5496