Fastai and Convolutional Neural Network Based Land Cover Classification

Priya Surana[1]*, Bhagwan Phulpagar [2] and Pramod Patil [3]*
[1] Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune 411044, India. priya.surana0204@gmail.com
[2]Department of Computer Engineering, PES, Modern College of Engineering, Pune 411005, India. phulpagarbd@gmail.com
[3]Department of Computer Engineering, Dr. D.Y.Patil Institute of Technology, Pune 411018
pdpatiljune@gmail.com
* Correspondence: priya.surana0204@gmail.com (P.S) ; pdpatiljune@gmail.com (P.P.)

Abstract: The primary objective of this research is to create a Deep Learning model that can accurately classify satellite images into predefined categories. To accomplish this goal, we developed an effective approach for satellite image classification that utilizes deep learning and the convolutional neural network (CNN) for feature extraction. We trained our model using a labeled dataset of satellite images provided by Planet Labs, which specializes in detecting various types of land covers. By utilizing the CNN algorithm, we were able to automatically extract features from satellite data with relatively minimal processing compared to other image classification algorithms. To develop our model, we employed the Fastai library, which enables us to quickly and effortlessly achieve state-of-the-art results in image classification tasks.

Keywords: Planet, Satellite image classification, Deep learning, Convolutional neural network, Features extraction,Fastai, ResNet50.

1. Introduction

Every minute, an area of forest the size of 48 football fields is lost worldwide, largely due to deforestation. This widespread issue results in reduced biodiversity, habitat loss, climate change, and other devastating effects. The primary cause of deforestation is the lack of

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

available data on dense forested areas. Despite efforts to develop forested areas, various challenges, including geographic and social obstacles, have prevented progress. These challenges include a lack of education, poor connectivity between locations, unreliable government policies, and, most significantly, insufficient or nonexistent records of many forested areas. The primary reason for this lack of information is the inaccessibility of many forested locations due to numerous issues. However, better data on the location of deforestation and human encroachment in forests can help governments and local stakeholders respond more quickly and effectively to this critical issue.

This paper presents a method to classify various geographical and environmental phenomena, such as haze/fog, land, river, forest fires, and tree blooms. We will delve into the data processing techniques utilized and provide a detailed description of the CNN architecture developed for solving this problem using convolutional neural networks (CNN).

This paper presents an approach for solving the classification problem of various natural and geographical elements using a CNN model. The need for this model is vital because classification of geographical phenomena is essential for gaining knowledge about various forests and their surrounding areas, which can aid in the development and protection of these regions by various industries and government sectors. Although a vast amount of data is readily available, it is often raw and requires appropriate rectification to gain meaningful insights. Traditional feature-based methods such as machine learning can be time-consuming when dealing with such large amounts of data. Therefore, deep learning techniques, especially CNNs, will be used in this project to classify images efficiently. CNNs have proven to be highly robust in the classification of natural elements.

Section-wise outline - structure of this paper:

In section 2 , provide an explanation of the fundamentals of Convolutional Neural Networks. Section 3 gives details the architectures used in this study. Section 4 and 5 Dataset and its pre-processing respectively. Section 6 implementation outlines of the model. Section 7 discusses the fine-tuning process and Results are presented in section 8 , and section 9 provides concluding remarks.

2. Convolutional Neural Networks (CNNs) are comprised of two basic elements: convolutional layers and pooling layers. The effectiveness and accuracy of the CNN model depends on how these layers are arranged during the model building process. CNNs are built using kernels, also known as filters, which extract the necessary features from input images. Since images in CNNs can be multi-dimensional, depending on the number of color-bands, CNNs consist of the following layers:

- Input layer
- Convolution layer
- Rectified Linear Unit (ReLU) layer
- Pooling layer
- Dropout layer
- Dense layer or Fully Connected layer

The convolution layer subdivides the input image into smaller regions, and the ReLU layer carries out the activation function. The pooling and dropout layers are used to reduce the links and size of the model. The pooling layer is mainly used to decrease the number of training samples. In the final fully connected layer, probability is used to generate labels for the images. The convolution layer is the heart of the CNN model, distinguishing it from regular neural networks. Its primary purpose is to extract features from input images. Different types of filters are applied to input images for feature extraction.[1]

2.1. Convolutional layer : The Convolutional layer serves as the foundation for deep learning architectures. Its input and output are known as feature maps. Each convolutional layer contains a fixed number of filters with a predetermined size, chosen beforehand before the training process begins. Each filter identifies a specific feature and generates a corresponding feature map in the output. The filter is characterized by a set of weights, and its size is typically dependent on the dimensions of the preceding layers. During training, each filter detects specific features of the images as it moves through them in each step. These features may include shapes, edges, or colors. With an increase in the number of layers, the filters can identify higher-level features from the images.

2.2. ReLu Layer: The ReLu layer is a type of nonlinear layer commonly used after convolution layers in deep learning. Its purpose is to improve the network's ability to approximate complex functions by performing a nonlinear transformation on the inputs. The most c ommonly used nonlinear function is the rectified linear unit (ReLU), which is expressed as $f(x) = \max(0, x)$. Compared to other activation functions such as sigmoid and hyperbolic tangent, ReLu is faster and less likely to saturate or suffer from the problem of vanishing gradient, as it has a constant gradient.

2.3. Pooling layer: The purpose of the pooling layer is to reduce the size of the feature maps. This layer includes three main types: Max Pooling, Average Pooling, and Global Pooling.Max Pooling divides images into non-overlapping rectangles of 2x2 and considers only the maximum value in each set, resulting in a four-fold decrease in size.

- Average Pooling divides images into non-overlapping 2x2 rectangles and considers the average of the four values.
- Global Pooling is used to replace fully connected layers in the network.
- Max Pooling is the most well-known type among these.

2.4. Dropout Layer : The dropout layer is a technique used during training that randomly drops out neurons. The dropout rate determines the probability of a neuron being dropped out, which helps to make the network less dense and prevents overfitting. This technique is only implemented during training.

2.5. Fully Connected Layer: The fully connected layer in a neural network receives input from the preceding layer and calculates class scores. It then produces a 1-dimensional array with a size equal to the number of classes.

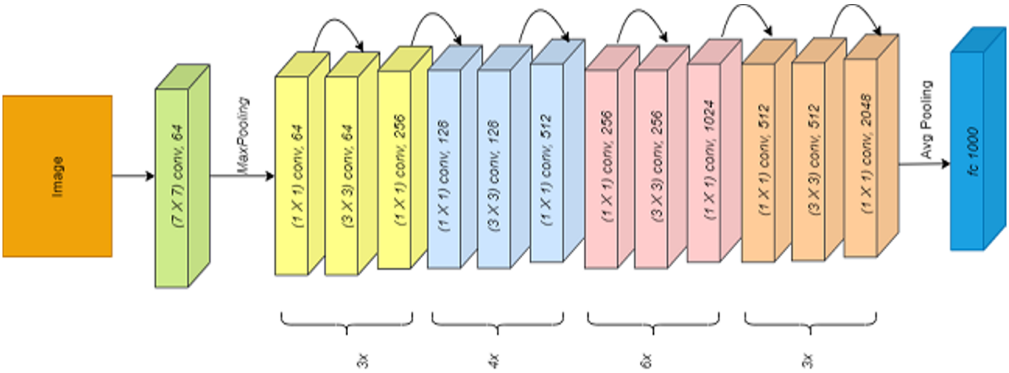


Fig. 1: ResNet50 Architecture

3. Neural Networks Architecture Used

- **ResNet50 : ResNet50** is a type of neural network model that was developed by Microsoft, and is characterized by its 48 convolution layers, 1 average pooling layer, and 1 max pooling layer. What sets ResNet50 apart from other models is its use of skip-layers, which are shortcut connections that help to overcome the vanishing gradient problem. The ResNet50 model consists of convolution and pooling layers, as well as a series of residual blocks that are stacked in a repetitive sequence. Batch normalization layers are added after each convolutional layer. The network ends with a fully connected layer, which is preceded by a pooling average layer. ResNet50 has about 24 million trainable parameters.

4. Data set : Available and Used

In the following section, the data set used in this study is described. The pre-processing of this data set is also summarized below. For our research implementation, we selected labels that are most viable and commonly occurring in the Amazon rainforest. These labels were divided into three major categories:

1) Cloud cover labels: encompassing various types of clouds and haze.

2) Common labels: including water, habitation, agriculture, road, cultivation, etc.

3) Rare labels: comprising mining, blooming, blowdown, and other naturally occurring phenomena in the Amazon Rainforests.

After evaluating several datasets, we found that the one provided by Planet Labs, Geo-Airbus Defense, and the National Institute for Space Research in Brazil were suitable for our study. The USGS Earth Explorer and Digital Global platforms had larger and sharper datasets, but they were unlabeled, requiring manual labeling.

Satellite Land Cover was not suitable as it did not cover all the parameters we needed. However, we had to discard the datasets from Geo-Airbus and the National Institute for Space Research in Brazil due to proprietary limitations and limited coverage, respectively. We ultimately selected the dataset from Planet Labs, which included 40,479 labeled training images in GeoTiff format with red, blue, green, and near-infrared images. The images were sampled from a larger 6600x2200 pixel "Planetscope Scene" and were provided as 256x256 pixel "chips"

- ✓ The Planet Data set: The data for this competition was obtained from Planet's website from their 4-band satellites. Big sized images were divided into sets of image chips which were in the GeoTiff format and Each image contained four bands of data: red, green, blue, and near infrared.
- ✓ For the purpose of the paper the data set was stripped out of all of the geotiff information regarding the chip footprint and ground control points. The data comes from Planet's Flock 2 satellites and the data was collected between January 1, 2016 and February 1, 2017. The main area under consideration is along the Amazon basin which includes Peru, Uruguay, Brazil, Ecuador, Colombia and Venezuela. A set of JPG images are also included for reference and practice.

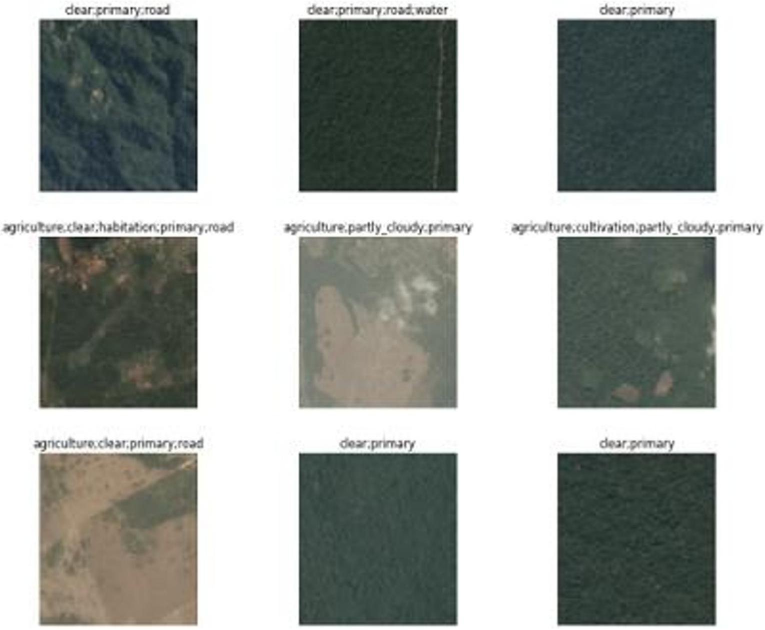

clear:primary:road
clear:primary:road:water
clear:primary
agriculture:clear:habitation:primary:road
agriculture:partly_cloudy:primary
agriculture:cultivation:partly_cloudy:primary
agriculture:clear:primary:road
clear:primary
clear:primary

Figure 2: Data-Set Sample Images

Figure 2 shows few samples of the data set original images that are labeled with one or more land cover labels, and these images have been classified into seventeen different classes based on their visual characteristics. Figure 3 provides a count of the number of images in each class.

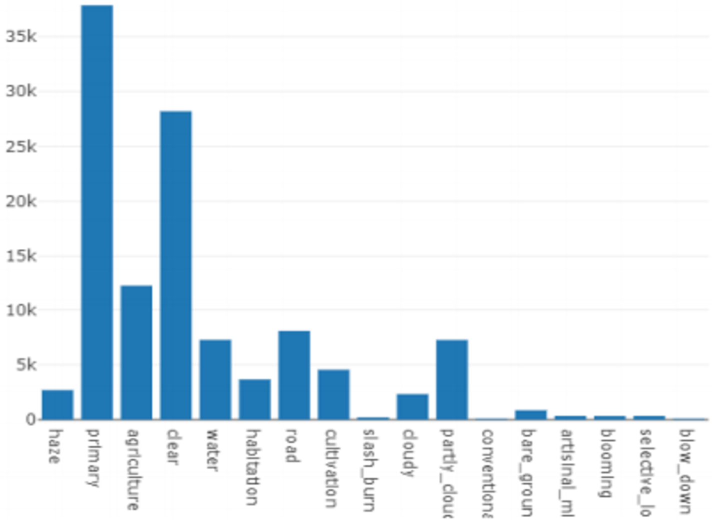


Fig.3. Different Classes in the data-set and their count

5. Data Pre-processing: This research utilized a pre-processed data-set that had already been labeled with accurate classifications. To improve the data-set, some pre-processing techniques were employed. One of these techniques involved resizing images to 128x128 for one model and 256x256 for another, and then augmenting the data-set with additional images. Image augmentation is a method that increases the number of images in the data-set, which improves the model's classification accuracy. The augmentation process utilized brightness augmentation rather than rotation, as the latter would have increased the number of images excessively. The image below illustrates the augmentation process.

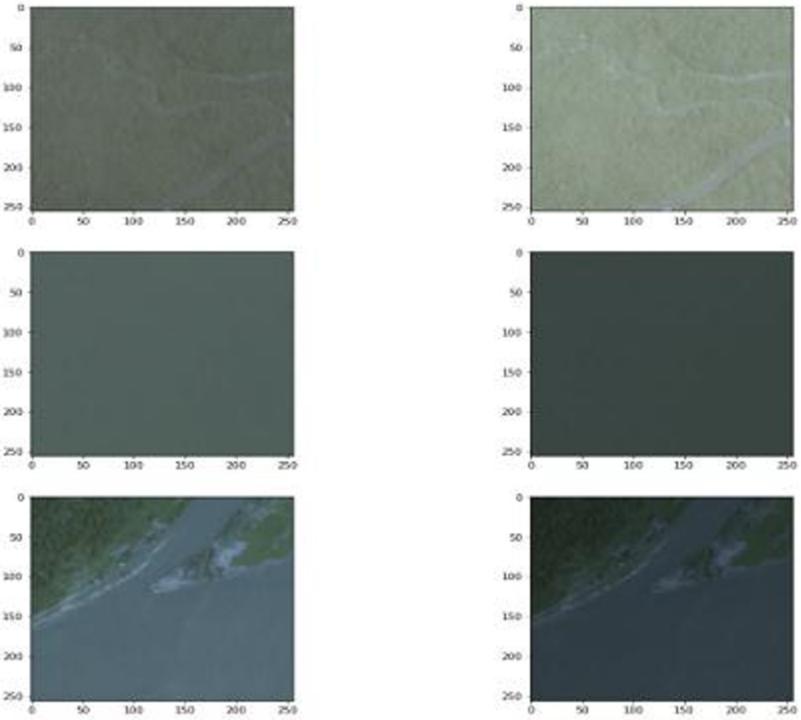


Figure 4: Augmentation by increasing the brightness Implementation Details

6. Implementation Details

Learning Algorithm:

The fastai library was used to train both networks, with an initial learning rate of 0.01. Unlike traditional methods that require trial-and-error in an iterative process to select the optimal learning rate, fastai simplified this process. It selected a suitable learning rate from a graph displaying the correlation between loss and learning rate. Therefore, a proper learning rate was determined.

Hardware and Software details : System Requirements

The Fastai library was utilized to train the networks, providing a straightforward and rapid way to prototype neural networks. The NVidia K80 GPU was used to train the network, while the data augmentation operation was conducted by the CPU. On average, the training process lasted for 4 hours and 30 minutes. Utilizing GPUs for training resulted in faster training times compared to using CPUs. All tests were conducted on a computing unit that had an Intel Core i5-5200U processor and 4 GB of RAM memory. The model was saved locally in a .pkl file.

7. Fine-Tuning

Fine-tuning involves making minor adjustments to the hyper-parameters of the neural network model to improve its accuracy. Rather than employing arbitrary values for the learning rates, the fastai library offered a more effective approach for setting the parameters. The graph below illustrates the correlation between loss and learning rate.

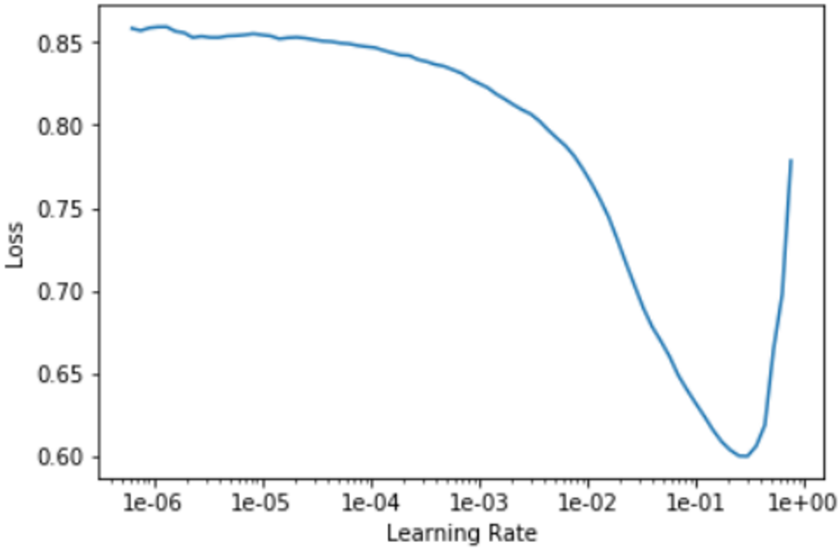


Fig. 5. Graph of Loss against Learning Rate for model with image size(128X128)

The graph demonstrates that the steepest decline in the loss-line occurs around 0.01, indicating that this value was utilized as the learning rate. The results for images sized at

128x128 are shown in Figure 8. After achieving favorable outcomes on smaller images, the model was retrained utilizing images of their original size (256x256). The following graph exhibits the correlation between loss and learning rate for the updated model.

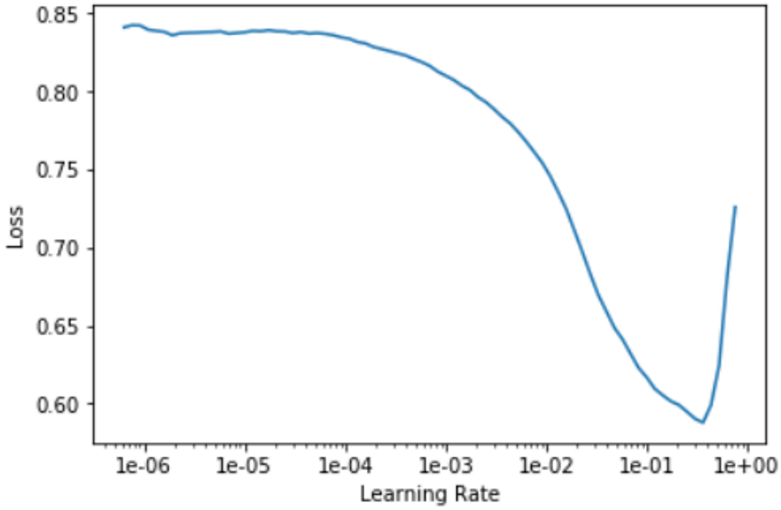


Fig. 6: Graph of Loss against Learning Rate for model with image size(256X256) without fine-tuning.

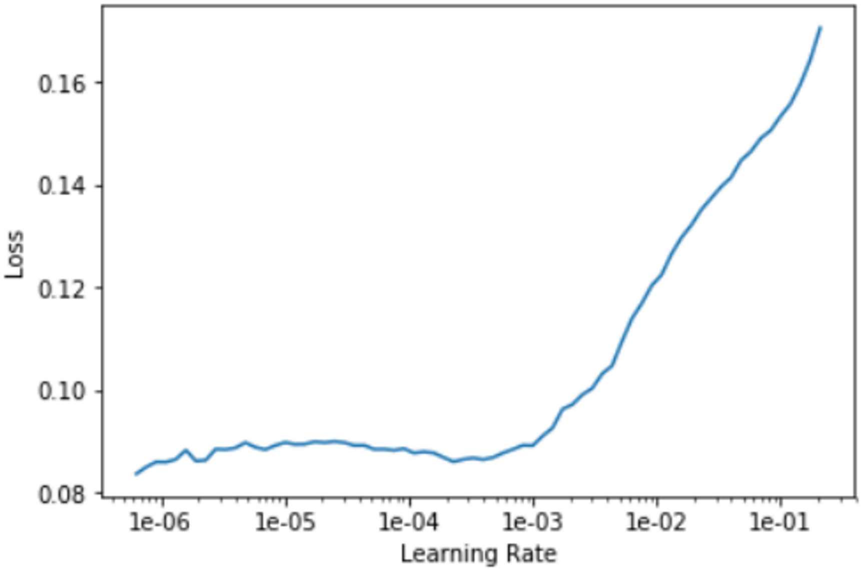
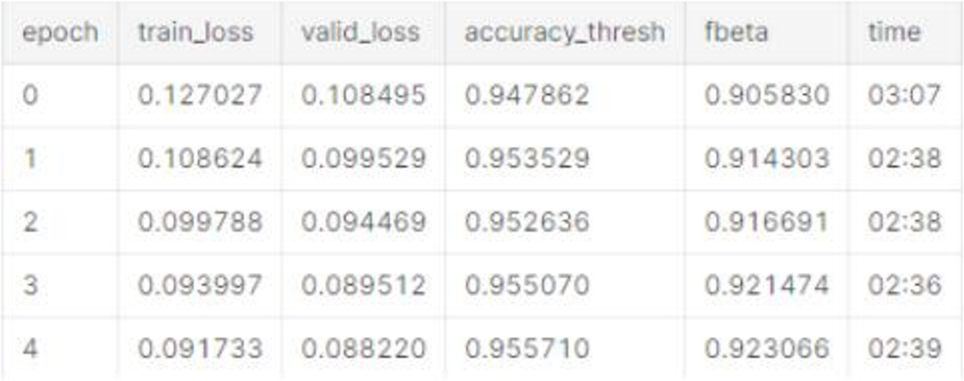


Fig. 7. Graph of Loss against Learning Rate for model with image size(256X256) with fine-tuning. (Graph Upon setting the new parameter value)

Once more, the graph indicated that 0.0001 was an optimal turning point before the line rapidly increased. Consequently, a learning rate lower than 0.0001 was selected, and ultimately 0.00001 was determined to be the best option. The network was then finalized, and the outcomes are illustrated in Figure 10

8. Results and Discussion

During the experiments, we evaluated the proposed model on a dataset consisting of over 60,000 images with 17 different labels. The test set, which was entirely separate from the main dataset, was used to assess the accuracy of the model. The classification accuracy of the input images into their respective land cover categories is shown in the table below, varying based on the input image size and different parameters with respect to the number of epochs. Since this was a multi-classification problem, the accuracy thresh was used as the primary performance metric, with the model returning multiple labels for each image as long as the probabilities of those labels were above a certain threshold. Additionally, the F2-score was also employed as an additional performance metric, generated using the fbeta function from the Fastai library.

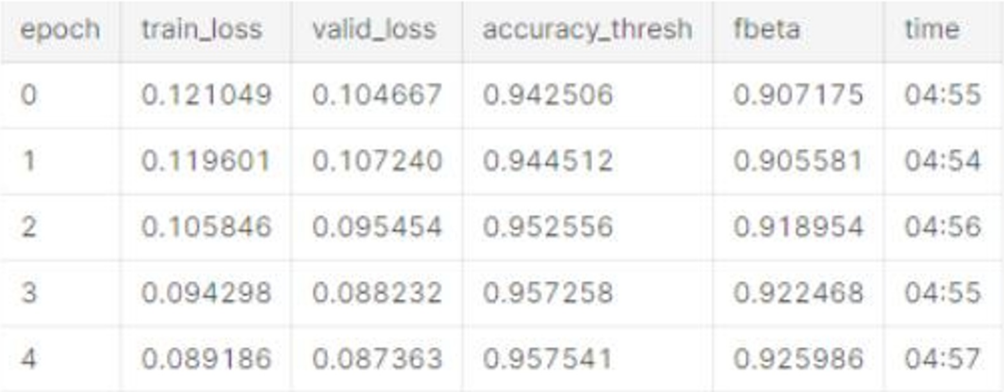


epoch	train_loss	valid_loss	accuracy_thresh	fbeta	time
0	0.127027	0.108495	0.947862	0.905830	03:07
1	0.108624	0.099529	0.953529	0.914303	02:38
2	0.099788	0.094469	0.952636	0.916691	02:38
3	0.093997	0.089512	0.955070	0.921474	02:36
4	0.091733	0.088220	0.955710	0.923066	02:39

Fig. 8. Result for Model with image size(128X128).

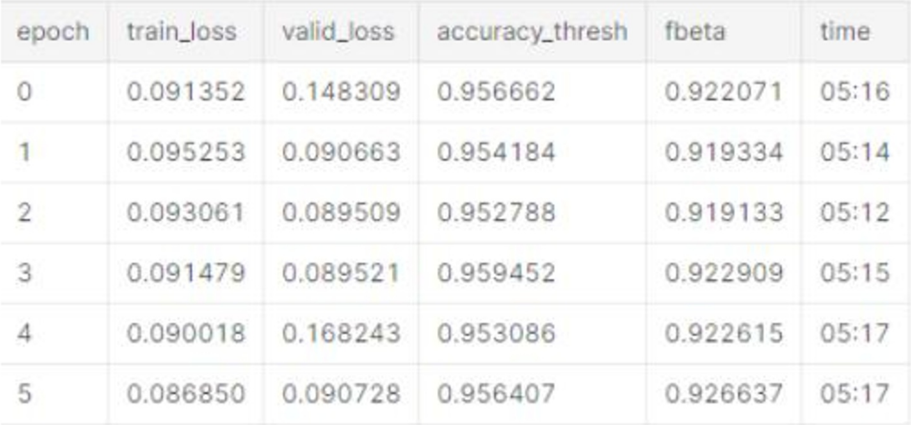
size (128 X 128) and getting good results, the model was trained on images of original size (256 X 256) to obtain an improved performance and better classification.

The results are:- • Base model:- Learning rate of – 0.01



epoch	train_loss	valid_loss	accuracy_thresh	fbeta	time
0	0.121049	0.104667	0.942506	0.907175	04:55
1	0.119601	0.107240	0.944512	0.905581	04:54
2	0.105846	0.095454	0.952556	0.918954	04:56
3	0.094298	0.088232	0.957258	0.922468	04:55
4	0.089186	0.087363	0.957541	0.925986	04:57

Fig. 9. Result for Model with image size(256X256) without fine-tuning.



epoch	train_loss	valid_loss	accuracy_thresh	fbeta	time
0	0.091352	0.148309	0.956662	0.922071	05:16
1	0.095253	0.090663	0.954184	0.919334	05:14
2	0.093061	0.089509	0.952788	0.919133	05:12
3	0.091479	0.089521	0.959452	0.922909	05:15
4	0.090018	0.168243	0.953086	0.922615	05:17
5	0.086850	0.090728	0.956407	0.926637	05:17

Fig.10: Results for Model with image size(256X256) with fine-tuning

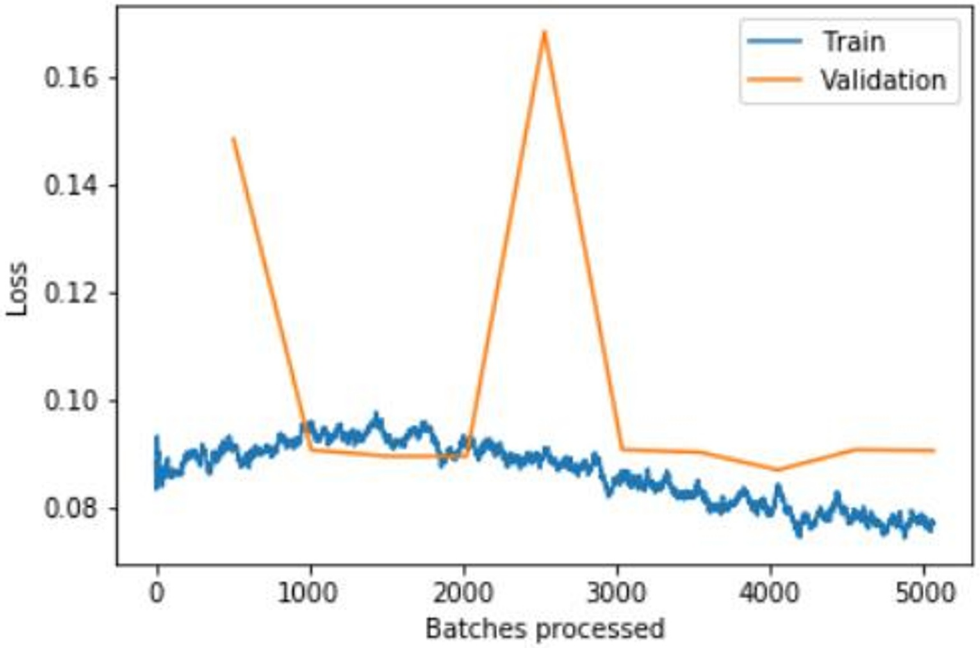


Figure 11: Train and Validation Loss graph.

Upon analysis of the gathered results, it is evident that the proposed network has achieved excellent classification results. The decision system that utilized the modified ResNet50 CNN with an image shape of (256X256)px and parameter tuning achieved the best performance. However, slightly poorer results were obtained by the ResNet50 classifier for the reduced size dataset of (128X128)px. It is noteworthy that all the networks demonstrated good performance with an accuracy threshold of over 94 percent.

9. Conclusions

This paper presents a CNN model that can classify satellite images to determine various types of land cover such as rivers, agriculture, forests, roads, and more. The model uses a CNN architecture that is trained on pre-processed data and is further fine-tuned to improve accuracy. The study uses Python programming language and fastai library for classification. Compared to traditional systems that use hand-extracted features, this model uses trainable convolutional layers as feature extractors. The results show that the proposed deep learning approach outperforms classical neural networks or fuzzy reasoning-based systems. The ResNet50 networks were used as decision-making systems and were validated on a dataset containing over 60,000 images. The best performance was achieved by ResNet50 with tuned hyper-parameters and original image size. However, the model with reduced image size performed slightly poorly. The paper proposes using fastai instead of traditional approaches to solve CNN problems, which reduces the time and resource requirements by providing easy-to-use methods for the classification process.

References

1. S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep Learning Convolutional Neural Networks for Radio Identification," IEEE Commun. Mag., vol. 56, no. 9, pp. 146– 152, 2018, doi: 10.1109/MCOM.2018.1800153.
2. Jeremy Howard and Sylvain Gugger, "fastai: A layered API for Deep Learning", CoRR, 2020.
3. Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, "Deep Residual Learning for Image Recognition" in arXiv 1512.03385, 2015
4. Kadhim, Mohammed, Abed, Mohammed, "Convolutional Neural Network for Satellite Image Classifica-tion", Studies in Computational Intelligence 10.1007/978-3-030-14132-5- 13., 2020.
5. Zhong, Yanfei Fei, Feng Liu, Yanfei Zhao, Bei Hongzan, Jiao Zhang, Liangpei, "SatCNN: satellite image dataset classification using agile convolutional neural networks.", Remote Sensing Letters, 8. 136-145. 10.1080/2150704X.2016.1235299, 2017
6. Yulang Chen, Jingmin Gao, Kebei Zhang, "R-CNN-Based Satellite Components Detection in Optical Im-ages", International Journal of Aerospace Engineering, vol.2020, Article-ID- 8816187, 10 pages, 2020.https://doi.org/10.1155/2020/8816187
7. Karen Simonyan Andrew Zisserman, " Very Deep Convolutional Networks For Large-scale image recognition", ArXiv 1409.1556., 2014.
8. Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna R. Nemani. 2015. DeepSat - A Learning framework for Satellite Imagery. CoRR abs/1509.03602 (2015).
9. Marco Castelluccio, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. 2015. Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. CoRR abs/ 1508.00092 (2015). http://arxiv.org/abs/1508.00092
10. Chen, C., Zhang, B., Su, H., Li, W., Wang, L.: Land-use scene classification using multi- scale completed local binary patterns. Signal Image Video Process. 10(4), 745–752 (2016)
11. Albert, A., Kaur, J., Gonzalez, M.: Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale. In: Proceeding of the

23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp. 1357– 1366 (2017)

12. D. C. Cires ̦an, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “Flexible, high performance convolutional neural networks for image classification,” in Proceedings of the 22nd International Joint Conference on Artificial Intelligence, vol. 2, pp. 1237–1242, 2011.
13. Zeiler, Matthew Fergus, Rob, “Stochastic Pooling for Regularization of Deep Convolutional Neural Net-works”, ICLR, 2013.
14. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural net-works by preventing co-adaptation of feature detectors”, CoRR, abs/1207.0580, 2012.
15. C. Szegedy et al., “Going deeper with convolutions,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9
16. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton,”ImageNet classification with deep convolutional neural networks”, Advances in Neural Information Processing Systems, 2012.