

Mysore sentinel-2: deep learning for image classification with optimizer exploration

Natya Sathyanarayana^{1,2}, Seema Singh³

¹Department of Electronics and Telecommunication Engineering, BMS Institute of Technology and Management, Autonomous under Visvesvaraya Technological University, Jnana Sangama, India

²Department of Electronics and Communication Engineering, Presidency University, Bengaluru, India

³Department of Electronics and Telecommunication Engineering, BMS Institute of Technology and Management, Autonomous under Visvesvaraya Technological University, Jnana Sangama, India

Article Info

Article history:

Received Nov 21, 2023

Revised Feb 14, 2024

Accepted Feb 16, 2024

Keywords:

Deep learning

Image classification

Optimization algorithm

Remote sensing

Sentinel-2 satellite image

ABSTRACT

The classification of Sentinel-2 image is presented in this work using a tile-based methodology. The Mysore district of India's Karnataka state serves as the subject region of this research. By tiling Sentinel-2 images, we were able to construct a distinct dataset and get approximately 3,000 training samples for the five classes. These images are manually labelled and geo-referenced. Three different optimizers were employed in a thorough analysis with deep learning models such as ResNet50, MobileNetV2, ShuffleNet, and VGG16 to achieve better performance metrics. With a classification accuracy of 98.1% on RESNet50 using ADAM surpassed the others. This facilitates investigating various geographical data analytics applications of the study region.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Natya Sathyanarayana

Department of Electronics and Communication Engineering, Presidency University

Bengaluru, India

Email: natyas@presidencyuniversity.in

1. INTRODUCTION

The procurement of data (i.e., physical characteristics) over the surface of the Earth by determining the radiation it emits/reflects when examined from the satellites/aircraft (based-sensor-technology) is referred to as remote sensing (RS) [1]. The classification of RS images into multi-spectral, super-spectral, and hyper-spectral categories is based on the number and range of spectral bands captured by the imaging sensors [2]. Massive RS data provides a challenge in terms of successfully extracting information due to vast features present in the images in addition to other complex traits like dimensionality, scalability and non-stationary (i.e., due to different resolution, scale and acquired at different times) [3]. With image classification techniques, it is possible to effectively extract information from RS data. Many sectors employ RS which also includes environmental monitoring (e.g., forest fires, flood zones, erupting volcanoes, dust storms) managing natural resources, urban planning, and disaster response [4], [5]. It is usually more affordable than time-consuming and expensive ground-based surveys. As a result, we are able to recognize and examine various types of land cover (LC), including forests, water bodies, and urban areas. We can track surface changes to the Earth, like deforestation, urbanization and land use (LU) changes, by examining satellite imagery over time. Analyzing satellite data requires a deep understanding of both the geospatial and image vision fields [6]–[10].

The European space agency (ESA) developed the Sentinel-2 (S2) satellite system to observe the earth. It is made up of two identical satellites that orbit the earth and are constantly collecting data about its

surface. High-resolution optical imagery data of land and coastal areas is made available by S2 satellites. S2 images are especially helpful for classifying LU and LC because they capture comprehensive data on the reflectance of various types of LC [11], [12]. In order to distinguish between vegetation, water sources, urban, forest, mountains, desert and bare-land and so on the system's multispectral sensors collect information in thirteen spectral bands. To create more detailed land use and land cover (LULC) maps, S2 images can also be combined with other datasets, such as elevation data or climate data. Numerous applications, including urban planning, resource management, and disaster response, can make use of these maps [13]-[15].

This research is motivated by the critical need to refine the understanding of geographic landscapes in an era where sophisticated RS predominates. Utilizing S2 image, the research focuses on the specific study region of Mysore. The task involves downloading high resolution images (HIS), preprocessing them converting them into smaller patches, and then applying diverse deep learning models and optimizers. By doing so, the research will not only contribute to improving the accuracy of LC classification in the designated region but aims to create a broader impact by laying the groundwork for advanced analyses in diverse areas that can benefit from a more refined understanding of LC.

This research makes a substantial contribution to the field of satellite image (SI) analysis by examining S2 imagery for the targeted region of Mysore in great detail. The work aims to improve kappa and accuracy as well as other performance metrics by systematically dividing large S2 images into patches and applying several deep learning models paired with different optimizers. The results have the potential to offer insightful information into best model-optimizer combinations, enhancing the reliability of RS applications. By bridging the gap between geographical analysis and cutting-edge technology, this research provides a substantial contribution to accuracy and effectiveness of classifying SI, with potential implications across various applications such as environmental monitoring, urban planning, agriculture and more.

The research comprises eight sections. The first section highlights the significance of LULC and introduces S2 imagery, and outlines the research's motivation and contribution. The second section is a literature review, while the third presents the methodology of the research. The fourth section describes the study area and the creation of the image dataset. The fifth details four deep learning model architectures, the sixth covers training parameters and the three optimizers used. The seventh section presents results from the models, including confusion matrices obtained from MATLAB tool for the distinct five classes/category. The final section concludes the research.

2. RELATED WORK

The majority of the recent research using deep learning (DL) to classify RS scenes is presented in this section. Prior to training a network to classify Landsat SI, Xu *et al.* [16] used principal-component-analysis (PCA) to reduce data redundancy. This method performed better than the maximum-likelihood (ML) by 18.5%. For the first time a hybrid framework that combines DL, logistic regression, and PCA was introduced by Chen *et al.* [17] on hyperspectral data. Using this DL framework, features were extracted using stacked autoencoders. Scott *et al.* [18] used UC-Merced dataset for LC classification employing convolutional neural network (CNN) models such as CaffeNet, GoogLeNet, and ResNet50. The training samples in the network was increased by the use of data augmentation, and the features of the images were adjusted for better performance (up to 99.3% accuracy). Gardner and Nichols [19] used DL models including VGG16, InceptionV3, and ResNet50 to achieve multi-label classification using 17 class labels. The best of these three was ResNet50. For remote sensing image classification (RSIC), Basu *et al.* [20] and Zou *et al.* [21] used deep belief network (DBN), and they experimentally proved the model's effects. Piramanayagam *et al.* [22] demonstrated CNN's potential for LULC by selecting training samples at every DL iteration for improved performance. Recurrent Neural Network and Random Forest are two improved classification techniques that Xu *et al.* [23] developed for LULC. Deep CNNs (DCNNs) such as VGG16, GoogLeNet, ResNet50, and others are used in most high precision classification, and handcrafted features are added to the features extracted by DCNNs. To reduce calculation time, DCNNs has numerous parameters that call for powerful computers. ShuffleNet was put forth by Ma *et al.* [24]. The complexity of the convolution technique was reduced by using pointwise group convolution rather than 1×1 Convolutional (Conv). It also introduced a channel shuffle operation to enhance the flow of feature information between channels in order to counteract the negative impacts of group Conv. With MobileNetv2, Li *et al.* [25] introduced feature fusion in the bilinear model and created BiMobileNet, an efficient and lightweight CNN for RSIC. The use of Depthwise (Dwise) separable convolution significantly reduces CNN parameters and calculations. Helber *et al.* [26] applied various band combinations with the GoogleNet and ResNet50 architectures. When compared to GoogleNet with RGB bands, they discovered that ResNet50 with RGB bands had the highest accuracy.

3. METHOD

The research approach, which outlines the key steps and processes carried out in the present study, is depicted in Figure 1. We begin by procuring a S2 satellite image of Mysore, followed by a pre-processing aimed at creating false color image (FCI) to enrich the features in the image. We then extract smaller image patches from FCI for classification task. Then manually label these smaller patches with the help of domain experts from Karnataka State Remote Sensing Applications Centre (KSRSAC) to create five classes (i.e., Bare-Land, Forest, Urban, Vegetation and Water) which acts as ground truth for our supervised learning. The next step involves choosing four deep learning models for classification tasks (ResNet50, MobileNetV2, ShuffleNet, and VGG16) and three optimizers (ADAM, SGDM, and RMSProp) for training. The DL models are trained using the optimizer, and assessed using a variety of metrics on the validation dataset. We finally conclude the best model-optimizer combo by assessing how well the models performed with various optimizers for each predefined class labels.

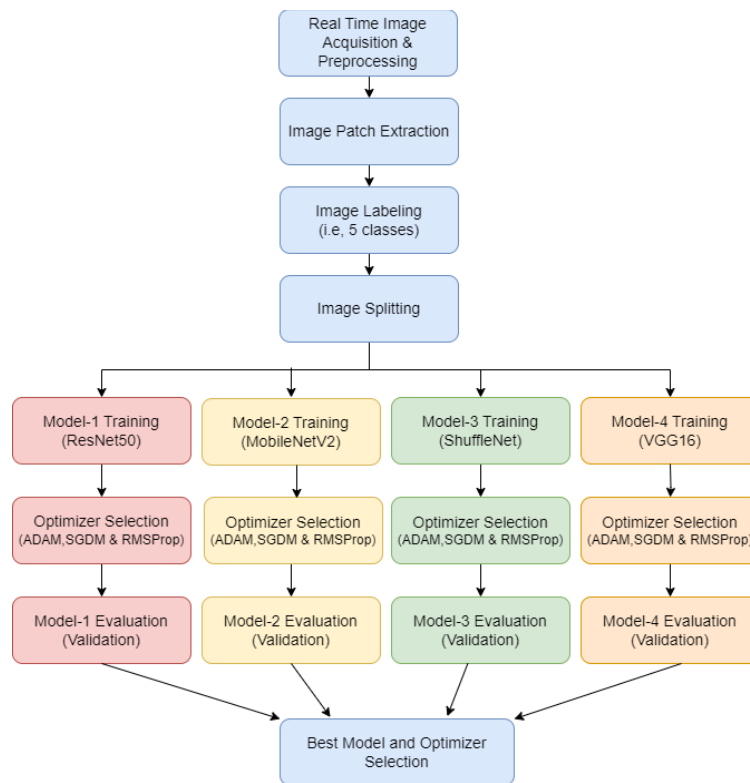


Figure 1. Process Flow of Methodology

4. DESCRIPTIONS OF THE STUDY AREA AND THE IMAGE DATASET CREATION

The Mysore district is located in the southernmost part of the Indian state Karnataka at coordinates 12° N and 76° E. Its elevation is between 660 and 788 meters. Mysore LU classification is crucial due to its cultural significance (palaces, festivals), historical sites (Wodeyar dynasty relics), and economic aspects (silk industry, tourism, forest-based industries, information technology). Efficient land management sustains heritage, supports economic sectors, and harmonizes development, ensuring the city's cultural, historical, and economic resilience. The S2 image of Mysore is obtained by Earth Explorer the acquired S2 image information is given in the Table 1. Using sentinel application platform (SNAP) this S2 image is converted to false color processing where Band8 (NRI), Band4 (RED) and Band3 (GREEN) are combined together as illustrated in Figure 2. Each pixel in the S2 image represents a 10×10 m square area of land. Image patches were extracted from a larger sentinel image (of the dimension 10,980×10,980 pixel), and each patch size is of 224×224 pixel. In Figure 3, visual representations of the five class labels are depicted: Figure 3(a) corresponds to Bare-Land, Figure 3(b) represents forest, Figure 3(c) illustrates urban, Figure 3(d) showcases Vegetation, and Figure 3(e) displays Water each class having around 600 images. A total of 3000 image patches were created with 70% of that for training (i.e., 2100 images) and 30% for testing (i.e., 900 images).

Table 1. Acquired S2 image information

Attribute	Value
ID	L1C_T43PFP_A035094_20220312T052910
Acquisition Date	2022-03-12 T05:29:10.446Z
Agency	ESA
Platform	SENTINEL-2A
Center Latitude	12°09'51.89"N
Center Longitude	76°25'25.23"E

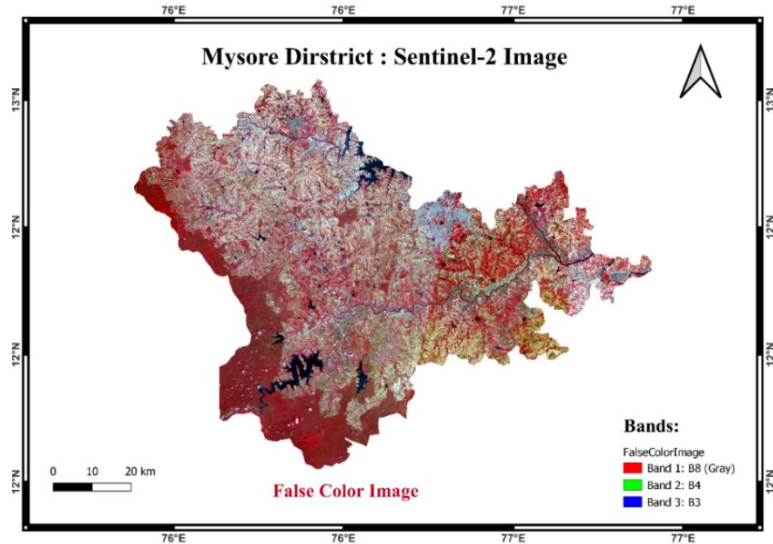


Figure 2. Mysore province's administrative boundaries in false color image (map developed using QGIS)

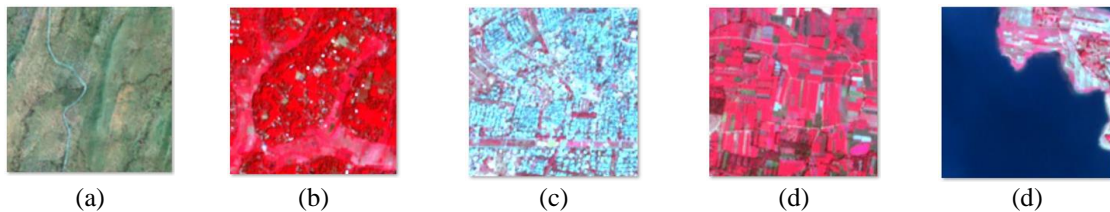


Figure 3. S2 dataset created for Mysore province and labeled accordingly as: (a) bare-land, (b) forest, (c) urban, (d) vegetation, and (e) water

5. DL MODELS

The selection of DL models such as ResNet50, MobileNetV2, ShuffleNet, and VGG16 for RSIC is their balanced trade-off between complexity and performance, suitability for diverse dataset characteristics, efficiency in computational resources, and availability of pre-trained weights for transfer learning. Consideration of training time and task-specific requirements, such as fine-grained classification, further guided the choice. When selecting a DL model for image classification, it's essential to evaluate these factors, ensuring the model aligns with the research specific needs and constraints in terms of computational resources and training efficiency.

5.1. VGG16

The visual geometry group (VGG) at the University of Oxford initially developed the VGG16 in 2014 [27] as shown in Figure 4. VGG16 is a 16-layer algorithm with 3-fully connected layers, 13 Conv layers, and 16 total layers. A string of Conv layers, followed by max-pooling (MaxPool) layers, define the VGG16 architecture. Filters with a 3x3 size are used in the Conv layers, which enables the network to learn more intricate aspects from the input image. The output of the Conv layers can be down-sampled using the MaxPool layer, which also helps to minimize the dimensionality of the feature maps. The fully connected (FC) layers then carry out the final categorization of the image using the output from the last Conv layer.

5.2. ResNet50

Microsoft Research unveiled residual Network-50 (ResNet50) in 2015 [28] as shown in Figure 5. The name ResNet50 refers to a network that uses residual connections to enhance network training and comprises 50 levels total, including both Conv and FC layers. Residual blocks in the ResNet50 design enable the network to learn residual functions rather than the underlying mapping directly. The residual block is made up of two Conv layers with 64 filters (3x3 size), along with a shortcut connection that adds the input through the second Conv layer's output. This aids in resolving the issue of vanishing gradients (VG) that arises during network training.

5.3. ShuffleNet

Zhang *et al.* [29] as shown in Figure 6. Using a channel shuffle operation to enable feature maps from several channels to communicate with one another is the main concept behind ShuffleNet. This increases accuracy while simultaneously reducing the number of parameters needed in the network. Group convolutions, which group the channels of the feature maps and conduct convolutions inside each group, are used to achieve the channel shuffle process. ShuffleNet also features a number of optimizations, such as group Conv, Dwise separable Conv, and Bottleneck design, to lower the computational overhead of the network.

5.4. MobileNetV2

Sandler *et al.* [30] as shown in Figure 7. The "inverted residual block" Conv layer and the "linear bottleneck" method, which minimizes the number of parameters in the network, are the two main advancements of MobileNetV2. The layers of an inverted residual block are as follows:

- Pointwise Conv layer: The input is subjected to a 1x1 pointwise Conv in the pointwise Conv layer.
- Dwise Conv layer: This layer transforms the pointwise Conv layer's output into a Dwise Conv.
- Linear bottleneck layer: Using a 1x1 pointwise Conv, this layer lowers the number of channels in the output of the Dwise Conv layer.

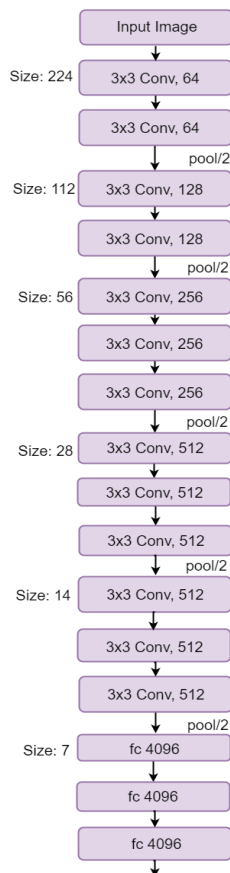


Figure 4. The VGG16 architecture

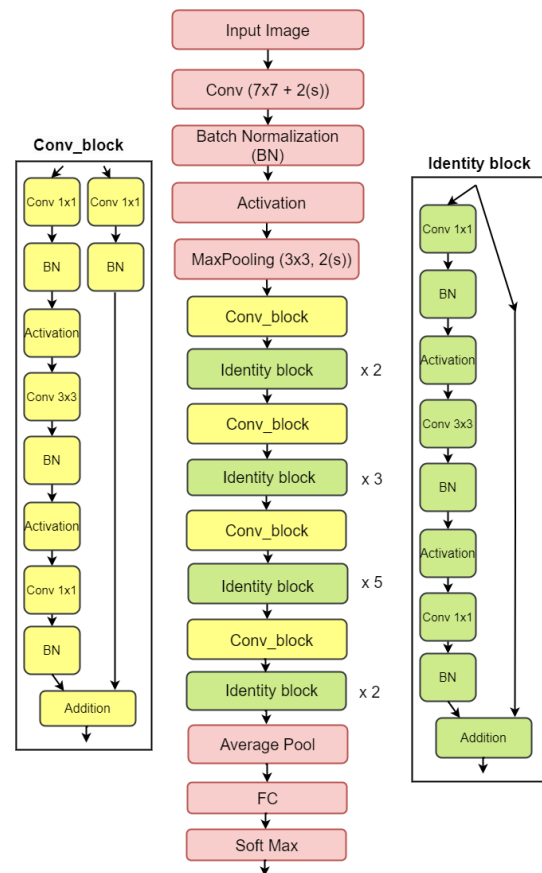


Figure 5. The ResNet50 architecture

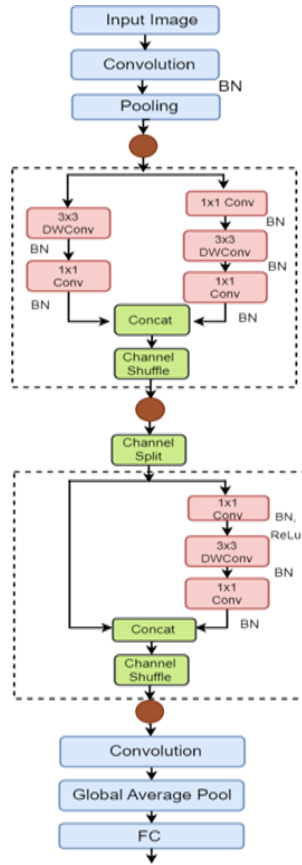


Figure 6. The ShuffleNet architecture

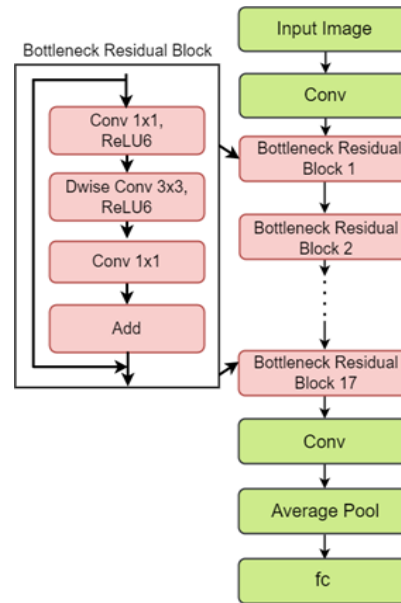


Figure 7. The MobileNetV2 architecture

6. TRAINING ALGORITHMS

During the training process of the DL model, it is imperative to adapt the weights of each epoch and minimize the loss function [31]. To achieve this, adjustments can be made to crucial neural network attributes, including learning rates (LR) and weights, utilizing a specialized function called an optimizer [32]. The optimizer plays a pivotal role in the modification of these properties, leading to an overall reduction in the total loss and an improvement in accuracy. The training of DL models involves the application of specific TA, as detailed in Table 2, with the utilization of three distinct optimizers: ADAM, SGDM, and RMSprop.

Table 2. Options for TA

Architecture	ResNet50, VGG16, ShuffleNet, MobileNetV2
Optimizer	SGDM, RMSprop, ADAM
Momentum	0.9
Initial Learn Rate	0.001
Learn Rate Drop Factor	0.1
Learn Rate Drop Period	10
Maximum Epochs	25
Mini Batch Size	128
Validation Frequency	50

6.1. Stochastic gradient descent with momentum (SGDM)

The conventional stochastic gradient descent (SGD) is improved with SGDM to enhance stability and convergence speed. This is achieved by introducing a momentum factor in the parameter update equation, smoothing the optimization process and preventing parameter space oscillations. SGDM's modification of the original SGD technique results in an optimized approach that strikes a balance between stability and convergence speed. The SGDM is stated in (1) and (2). Here, β is momentum coefficient, this regulates the preceding velocity's contribution to recent update. $u(t)$ is the momentum vector at time t , which is initialized as zero. $\nabla Q(\theta(t), x_i, y_i)$ is Gradient of the Loss Function (GLF) w.r.t the model parameters θ for

a single training example (x_i, y_i) . α is the LR, which controls the step size of the parameter update. $\theta(t)$ is the current model parameters (CMP) at time “t”.

$$u(t) = \beta * u(t - 1) + (1 - \beta) * \nabla Q (\theta(t), x_i, y_i) \tag{1}$$

$$\theta(t + 1) = \theta(t) - \alpha * u(t) \tag{2}$$

6.2. Root mean square propagation (RMSProp)

The primary objective of RMSProp is to adjust the scale of the gradient by considering the historical average of squared gradients. This approach proves beneficial in reducing the step size for parameters characterized by substantial gradients, while simultaneously increasing the step size for parameters with more modest gradients. In essence, RMSProp tailors the gradient rescaling to the historical behavior of squared gradients, resulting in an adaptive optimization strategy that can effectively navigate diverse gradient magnitudes. The RMSprop is stated in (3). Here, $\theta(t)$ is the CMP. α is the LR. $Q(t)$ is the GLF w.r.t the parameters at time t. $E[Q^2(t)]$ is the exponentially weighted moving average of the squared gradient. ϵ is epsilon a small constant to avoid division by zero.

$$\theta(t + 1) = \theta(t) - \frac{\alpha * Q(t)}{\sqrt{E[Q^2(t)] + \epsilon}} \tag{3}$$

6.3. Adaptive moment estimation (ADAM)

By applying estimations of the 1st and 2nd moments of the gradients, ADAM calculates adaptive LR for each parameter. The average gradient is the 1st moment, while the average squared gradient is the 2nd moment. Hyperparameters are used to modify the decay rates of exponentially decaying moving averages, which are used to calculate these estimations. For the purpose of correcting the initial bias in the moving averages, the algorithm also employs bias reduction. The ADAM is stated in equation (4). Here, $\theta(t)$ is the CMP. α is the LR, $m^{\wedge}(t)$ and $n^{\wedge}(t)$ are bias-corrected estimates of the 1st and 2nd moment vectors of the gradients. ϵ is epsilon a small constant to avoid division by zero.

$$\theta(t + 1) = \theta(t) - \frac{\alpha * m^{\wedge}(t)}{\sqrt{n^{\wedge}(t) + \epsilon}} \tag{4}$$

7. RESULTS AND DISCUSSION

In previous research, the impacts of various spectral bands and image processing techniques on LC classification have been explored, yet there's a notable gap in understanding the effectiveness of combining false-colour processing with advanced DL models like ResNet50, InceptionNet, MobileNet, and ShuffleNet, particularly in the unique context of the Mysore district. Furthermore, while DL optimizers such as Adam, RMSprop, and SGDM are commonly employed, their performance comparison in LC classification within this region remains relatively unexplored. Therefore, our study aims to bridge this gap by investigating the integration of false-colour processing with diverse DL models and optimizers for precise LC classification in the Mysore district. Through meticulous analysis of various model-optimizer combinations, we seek to uncover the most effective approach for accurately mapping LC in this distinct geographical area. Our investigation revealed significant variations in accuracy among different DL models and optimizer combinations for LC classification in the Mysore district.

The details of the research setup, including system specifications, are documented in Table 3. The generation of results is carried out using MATLAB as the primary tool. This section offers a thorough elucidation of how the accuracy evaluation of 4 distinct DL models, spanning 5 class/categories, is conducted through the utilization of 3 diverse optimizers. The confusion matrix (CM) obtained from MATLAB serves as a pivotal metric in this assessment, providing insights into the performance of the models under different optimization strategies.

Table 3. System specification

System name	Dell Inspiron 7501
Processor	10 th Generation Intel Core i7-10750H
Graphics	NVIDIA GeForce GTX 1650 Ti
Wireless card	Intel® Wireless-AC 9560 160 MHz
RAM	16 GB
Tool	MATLAB-R2022b
OS	Windows10

The Figure 8 presents the CM for ResNet50, the accuracy of the ADAM optimizer is shown in Figure 8(a) which is 98.1% and that it performed better across all 5 class/categories. The ResNet50-SGDM scored 96.3% as seen in Figure 8(b) and RMSprop 97.2% as seen in Figure 8(c). For ResNet50, ADAM scored best for the forest databases but poorly for both water and vegetation databases. The CM for VGG16 is provided in Figure 9, the SGDM optimizer achieved better across 5 given class/categories and raised accuracy by about 92.2% as shown in Figure 9(b). Whereas ADAM scored second best with 91.4% as shown Figure 9(a) and the least is RMSprop with 89.9% as shown in Figure 9(c). For VGG16, SGDM worked best for the databases on forest while it fared horribly for the databases on water. ShuffleNet's CM is presented in Figure 10, the ADAM optimizer as seen in Figure 10(a) boosted accuracy by about 95.3% and performed better across all 5 class/categories. ShuffleNet's-SGDM has accuracy of 93.7% as seen in Figure 10(b) and RMSprop has second highest accuracy of 94.1% as seen in Figure 10(c). ShuffleNet's forest and urban databases fared well, but the water database had poor ADAM performance. The Figure 11 presents the CM for MobileNetV2, the RMSprop optimizer as seen in Figure 11(c) improved accuracy by approximately 93.4% while performing better across all 5 categories. MobileNetV2-ADAM has 93% accuracy seen in Figure 11(a) and SGDM with 92.8% accuracy as seen in Figure 11(b). When it came to MobileNetV2's bare land and forest database, RMSprop performed well, but the water database had low performance.

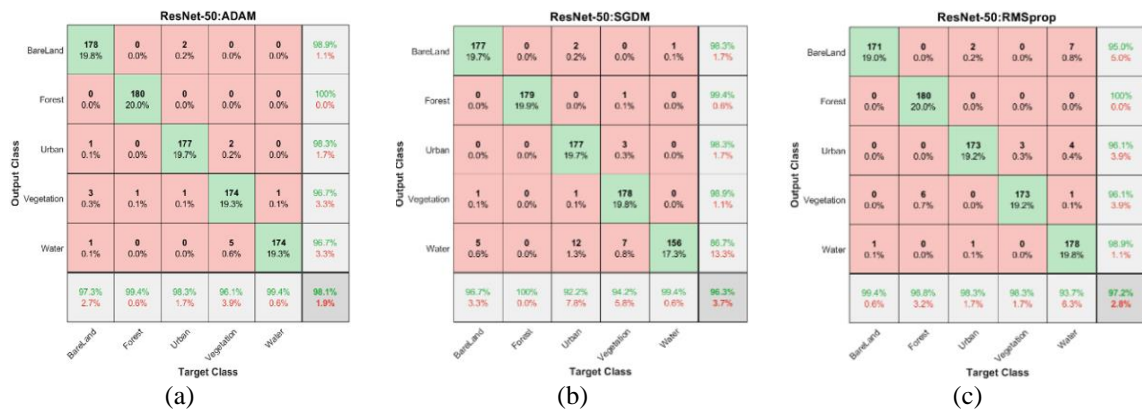


Figure 8. ResNet50 - MATLAB output: (a) CM of ADAM, (b) CM of SGDM, and (c) CM of RMSprop

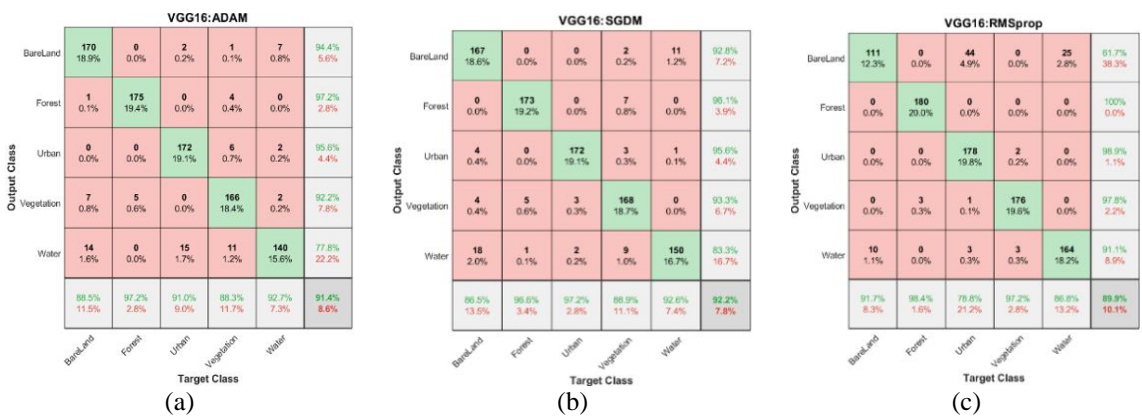


Figure 9. VGG16 - MATLAB output: (a) CM of ADAM, (b) CM of SGDM, and (c) CM of RMSprop

ResNet50 with “ADAM” as the optimizer, as shown in Table 4, achieves 98.1% accuracy with a kappa of 0.9764 and 98 minutes and 11 seconds of processing time. The accuracy for the same dataset with VGG16 using the “SGDM” optimizer is 92.2%, with a kappa of 0.9028, and it takes 110 minutes and 58 seconds. For the exact same dataset, ShuffleNet’s “ADAM” optimizer exhibits accuracy of 95.3% with a kappa of 0.9417 and processing time of 63 minutes and 21 seconds. The accuracy for the same dataset with MobileNet-V2 with “RMSprop” optimizer is 93.4%, with a kappa of 0.9181, and it takes 68 minutes and 55 seconds. It is significant to consider that network depth affects computing time during both training and

inference. The DL model employed in this study has the following network depth; ResNet50 has 50 layers, VGG16 has 16 layers, ShuffleNet has 50 layers, and MobileNet-V2 has 53 layers. ShuffleNet's with ADAM required less computation time but had the second-highest accuracy out of the four DL models, while VGG16 with SGDM consumed the most computation time with the least accuracy. If accuracy alone is taken into account, ResNet50 outperformed all three optimizers and has the highest accuracy in comparison to the others. Therefore, based on the results of this study, we can say that the simulation's precision and execution time are influenced by the database's size and the degree of complexity of the DL model as well as the optimizer's chosen.

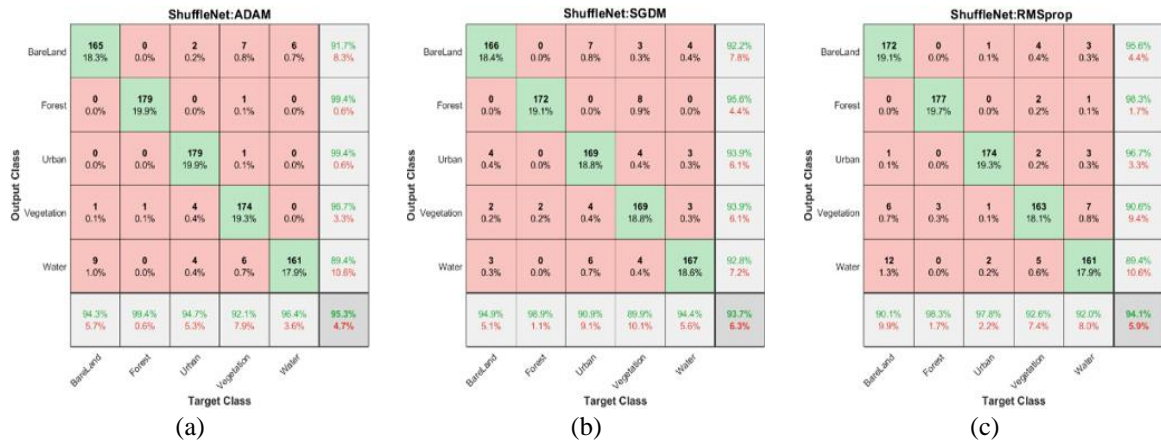


Figure 10. ShuffleNet-MATLAB output: (a) CM of ADAM, (b) CM of SGDM, and (c) CM of RMSprop

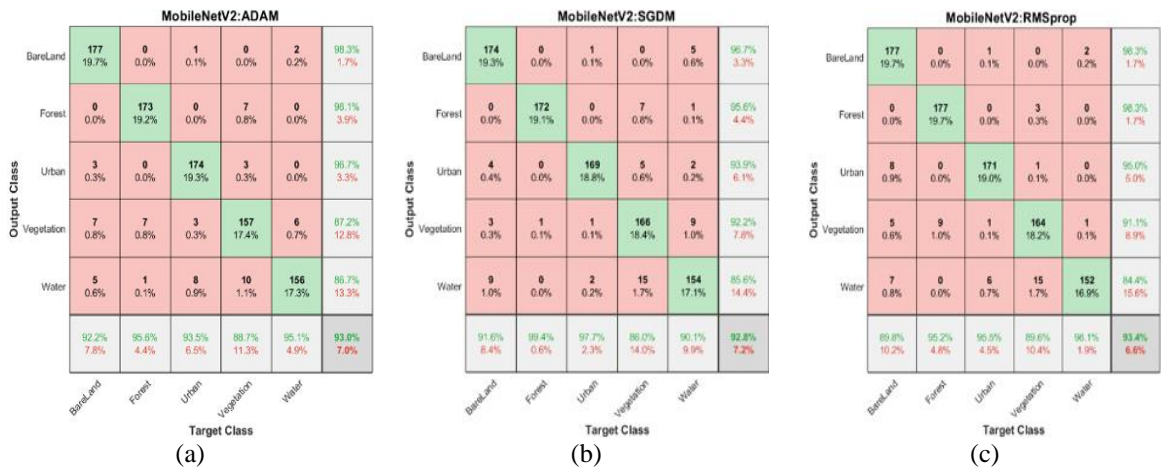


Figure 11. MobileNetV2-MATLAB output: (a) CM of ADAM, (b) CM of SGDM, and (c) CM of RMSprop

Table 4. Performance metrics of the DL Models with the optimizers

DL models	Optimizer	Kappa	Overall precision	Overall recall	F1 score	Training time
ResNet50	ADAM	0.9764	0.9811	0.9812	0.9812	98 min & 11 sec
	SGDM	0.9542	0.9633	0.9649	0.9641	106 min & 24 sec
	RMSprop	0.9653	0.9722	0.9729	0.9726	83 min & 19 sec
VGG16	ADAM	0.8931	0.9144	0.9156	0.9150	87 min & 07 sec
	SGDM	0.9028	0.9222	0.9237	0.9229	110 min & 58 sec
	RMSprop	0.8736	0.8989	0.9057	0.9023	47 min & 15 sec
ShuffleNet	ADAM	0.9417	0.9533	0.9538	0.9536	63 min & 21sec
	SGDM	0.9208	0.9367	0.9376	0.9371	76 min & 03 sec
	RMSprop	0.9264	0.9411	0.9415	0.9413	62 min & 06 sec
MobileNet-V2	ADAM	0.9125	0.9300	0.9303	0.9301	75 min & 15 sec
	SGDM	0.9097	0.9278	0.9295	0.9286	54 min & 31 sec
	RMSprop	0.9181	0.9344	0.9364	0.9354	68 min & 55 sec

8. CONCLUSION

We demonstrate a tile-based solution for classifying S2 images for Mysore. An effort is made to implement and assess various DL approaches along with three different optimizers which have produced promising outcomes for LULC classification. High accuracy of 98.10% has been achieved in classification task while using ResNet50 and the ADAM optimizer. Also, the comparison of three distinct optimizers as discussed in result section it reveals considerable gain in accuracy and convergence rate, especially for the given S2 datasets. However, the choice of optimizer depends on the model architecture employing RS images (dataset), which are employed by decision-makers in a wide range of domains of application. Classification accuracy may be improved further by fine-tuning model parameters, such as architecture and hyper-parameter optimization. However, there are some constraints that must be addressed. Robust validation techniques must be implemented to ensure the reliability and generalizability of the models. Additionally, accepting potential challenges in extending discoveries to diverse geographical areas highlights the obligation for flexibility and cautious consideration of regional variations in LU characteristics.





REFERENCES

- [1] G. Cheng, X. Xie, J. Han, L. Guo, and G. S. Xia, "Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3735–3756, 2020, doi: 10.1109/JSTARS.2020.3005403.
- [2] S. Dhingra and D. Kumar, "A review of remotely sensed satellite image classification," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, pp. 1720–1731, 2019, doi: 10.11591/ijece.v9i3.pp1720-1731.
- [3] J. Song, S. Gao, Y. Zhu, and C. Ma, "A survey of remote sensing image classification based on CNNs," *Big Earth Data*, vol. 3, no. 3, pp. 232–254, 2019, doi: 10.1080/20964471.2019.1657720.
- [4] C. Zhang, G. Li, and S. Du, "Multi-scale dense networks for hyperspectral remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9201–9222, 2019, doi: 10.1109/TGRS.2019.2925615.
- [5] M. Kanthi, T. H. Sarma, and C. S. Bindu, "Multi-scale 3D-convolutional neural network for hyperspectral image classification," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 25, no. 1, pp. 307–316, 2022, doi: 10.11591/ijeecs.v25.i1.pp307-316.
- [6] R. Qin and T. Liu, "A review of landcover classification with very-high resolution remotely sensed optical images—analysis unit, model scalability and transferability," *Remote Sensing*, vol. 14, no. 3, p. 646, 2022, doi: 10.3390/rs14030646.
- [7] S. Natya and V. Rehna, "Land cover classification schemes using remote sensing images: a recent survey," *British Journal of Applied Science & Technology*, vol. 13, no. 4, pp. 1–11, 2016, doi: 10.9734/bjast/2016/22037.
- [8] A. Asokan, J. Anitha, M. Ciobanu, A. Gabor, A. Naaji, and D. J. Hemanth, "Image processing techniques for analysis of satellite images for historical maps classification-An overview," *Applied Sciences (Switzerland)*, vol. 10, no. 12, p. 4207, 2020, doi: 10.3390/app10124207.
- [9] A. A. Adegun, S. Viriri, and J. R. Tapamo, "Review of deep learning methods for remote sensing satellite images classification: experimental survey and comparative analysis," *Journal of Big Data*, vol. 10, no. 1, 2023, doi: 10.1186/s40537-023-00772-x.
- [10] S. Y. J. Prasetyo, W. Sulisty, P. N. Basuki, K. D. Hartomo, and B. Hasiholan, "Computer model of Tsunami vulnerability using machine learning and multispectral satellite imagery," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 11, no. 2, pp. 986–997, 2022, doi: 10.11591/eei.v11i2.3372.
- [11] M. Campos-Taberner *et al.*, "Understanding deep learning in land use classification based on Sentinel-2 time series," *Scientific Reports*, vol. 10, no. 1, p. 17188, 2020, doi: 10.1038/s41598-020-74215-5.
- [12] M. A. Zaytar and C. El Amrani, "Satellite image inpainting with deep generative adversarial neural networks," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, p. 121, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp121-130.
- [13] P. Fazzini *et al.*, "Sentinel-2 Remote Sensed Image Classification with Patchwise Trained ConvNets for Grassland Habitat Discrimination," *Remote Sensing*, vol. 13, no. 12, p. 2276, Jun. 2021, doi: 10.3390/rs13122276.
- [14] D. Scharvogel, M. Brandmeier, and M. Weis, "A deep learning approach for calamity assessment using sentinel-2 data," *Forests*, vol. 11, no. 12, pp. 1–21, 2020, doi: 10.3390/f11121239.
- [15] S. Natya, R. K. K. and S. Singh, "Insights on Deep Learning based Segmentation Schemes Towards Analyzing Satellite Imageries," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 119–129, 2021, doi: 10.14569/IJACSA.2021.0121114.
- [16] J. B. Xu, L. S. Song, D. F. Zhong, Z. Z. Zhao, and K. Zhao, "Remote sensing image classification based on a modified self-organizing neural network with a priori knowledge," *Sensors and Transducers*, vol. 153, no. 6, pp. 29–36, 2013, [Online]. Available: https://www.sensorsportal.com/HTML/DIGEST/june_2013/P_1216.pdf
- [17] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014, doi: 10.1109/JSTARS.2014.2329330.
- [18] G. J. Scott, R. A. Marcum, C. H. Davis, and T. W. Nivin, "Fusion of deep convolutional neural networks for land cover classification of high-resolution imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 9, pp. 1638–1642, Sep. 2017, doi: 10.1109/LGRS.2017.2722988.
- [19] D. Gardner and D. Nichols, "Multi-label Classification of Satellite Images with Deep Learning," 2017. Accessed: Nov. 18, 2023. [Online]. Available: <http://cs231n.stanford.edu/reports/2017/pdfs/908.pdf>
- [20] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "DeepSAT: a learning framework for satellite imagery," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, New York, NY, USA: ACM, Nov. 2015, pp. 1–10. doi: 10.1145/2820783.2820816.
- [21] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015, doi: 10.1109/LGRS.2015.2475299.
- [22] S. Piramanayagam, W. Schwartzkopf, F. W. Koehler, and E. Saber, "Classification of remote sensed images using random forests and deep learning framework," in *Proceedings of SPIE*, L. Bruzzone and F. Bovolo, Eds., SPIE, Oct. 2016, p. 100040L. doi: 10.1117/12.2243169.





- [23] X. Xu, Y. Chen, J. Zhang, Y. Chen, P. Anandhan, and A. Manickam, "A novel approach for scene classification from remote sensing images using deep learning methods," *European Journal of Remote Sensing*, vol. 54, no. sup2, pp. 383–395, Mar. 2021, doi: 10.1080/22797254.2020.1790995.
- [24] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: practical guidelines for efficient CNN architecture design," in *Computer Vision – ECCV 2018*, 2018, pp. 122–138. doi: 10.1007/978-3-030-01264-9_8.
- [25] E. Li, A. Samat, P. Du, W. Liu, and J. Hu, "Improved bilinear CNN model for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2020.3040153.
- [26] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: a novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, Jul. 2019, doi: 10.1109/JSTARS.2019.2918242.
- [27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision – ECCV 2016*, 2016, pp. 630–645. doi: 10.1007/978-3-319-46493-0_38.
- [29] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: an extremely efficient convolutional neural network for mobile devices," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018, pp. 6848–6856. doi: 10.1109/CVPR.2018.00716.
- [30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
- [31] R. Sun, "Optimization for deep learning: theory and algorithms," *arXiv:1912.08957 [cs, math, stat]*. pp. 1–60, 2019. doi: 10.48550/arXiv.1912.08957.
- [32] S. Bera and V. K. Shrivastava, "Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification," *International Journal of Remote Sensing*, vol. 41, no. 7, pp. 2664–2683, Apr. 2020, doi: 10.1080/01431161.2019.1694725.

BIOGRAPHIES OF AUTHORS



Ms. Natya Sathyanarayana     is currently working as an Assistant Professor in the Department of Electronics and Communication Engineering at Presidency University, Bengaluru, Karnataka, India. She has over 13+ years of experience in Academia. She has completed her B.E degree from AIT, Chikkamagaluru and M.E degree from U.V.C.E, Bengaluru. She's pursuing a Ph.D. from BMSIT&M, Bengaluru affiliated to Visvesvaraya Technological University, Jnana Sangama, Belagavi. She has published around 10 research papers in peer reviewed journals and conferences of national and international repute. Her research and teaching interests include image processing, machine learning, deep learning, remote sensing, and embedded systems. She can be contacted at email: natyas@presidencyuniversity.in.



Dr. Seema Singh     received B.Tech degree in Electronics and Communication Engineering from Jamia Millia Islamia, Delhi in 2002. Her M.Tech degree is in Electronics branch from Sir MVIT, Bangalore in 2007. She is Gold medalist and VTU rank holder in M.Tech degree. She is awarded the PhD degree from JNTU, Hyderabad in the year 2015 in the area of Neural Networks and Flight control systems. She has 20+ years of academic, research and administrative experience and currently working as Professor and Dean (External Relations), Electronics and Telecommunication Engineering Department at BMS Institute of Technology and Management, Bengaluru, Karnataka. Her research work is mainly dedicated to application of neural networks in varied fields with undergraduate, postgraduate and Ph.D. scholars. She can be contacted at email: seemasingh@bmsit.in.