International Journal for Modern Trends in Science and Technology
Volume 11, Issue 05, pages 105-114
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol11issue05.html
DOI: https://doi.org/10.5281/zenodo.15266770

Satellite Image Categorization Using Scalable Deep Learning

Dr.T.Lakshmi Narayana, Bhavitha Lakkoju, Venkata Kavya Angirekula, Hepsibha Ponnuru

Department of Electronics and Communication Engineering, Andhra Loyola Institute of Engineering and Technology, Andhra Pradesh, India.

To Cite this Article
Dr.T.Lakshmi Narayana, Bhavitha Lakkoju, Venkata Kavya Angirekula & Hepsibha Ponnuru (2025). Satellite Image Categorization Using Scalable Deep Learning. International Journal for Modern Trends in Science and Technology, 11(05), 105-114. https://doi.org/10.5281/zenodo.15266770

Article Info
Received: 22 March 2025; Accepted: 18 April 2025.; Published: 20 April 2025.

KEYWORDS

maritime monitoring; satellite imagery; deep learning; remote sensing; image classification

ABSTRACT

Detecting and classifying objects from satellite images are crucial for many applications, ranging from marine monitoring to land planning, ecology to warfare, etc. Spatial and temporal information-rich satellite images are exploited in a variety of manners to solve many real-world remote sensing problems. Satellite image classification has many associated challenges. These challenges include data availability, the quality of data, the quantity of data, and data distribution. These challenges make the analysis of satellite images more challenging. A convolutional neural network architecture with a scaling method is proposed for the classification of satellite images. The scaling method can evenly scale all dimensions of depth, width, and resolution using a compound coefficient. It can be used as a preliminary task in urban planning, satellite surveillance, monitoring, etc. It can also be helpful in geo-information and maritime monitoring systems. The proposed methodology is based on an end-to-end, scalable satellite image interpretation. It uses spatial information from satellite images to categorize these into four categories. The proposed method gives encouraging and promising results on a challenging dataset with a high inter-class similarity and intra-class variation. The proposed method shows 99.64% accuracy on the RSI-CB256 dataset.

1. INTRODUCTION

Earth images collected by satellite are called satellite image sources. These images are also referred to as spaceborne photographs. Satellite companies provide these images for utilization in a variety of application domains.

Satellite images can be helpful in a diverse variety of application areas. Some of the key application domains are meteorology, land planning , academia ,

surveil- lance , monitoring, agriculture , marine studies , conservation, geology, and warfare . The time-series data of satellite images can be helpful in analyses and long-term planning in various domains. They can be utilized for anomaly detection by observing unusual or unexpected patterns. Satellite images can be presented in a variety of spectra, including visible colors.

Domain experts from different walks of life are interested in analyzing satellite images for various applications. The computer vision community provides services to solve the application-specific analysis of satellite images. However, certain tasks in satellite image analyses are very frequently performed to satisfy application-specific services. Image classification, image interpretation, and object recognition in satellite images are the essen- tial tasks that computer vision experts often employ when dealing with satellite images. Satellite image classification is primarily used for land cover classification, which entails classifying various land cover types, such as forests, croplands, urban areas, and water bodies, from satellite images. Precise land cover classification is crucial for a range of applications, including natural resource management, land use planning, agricultural management, urban planning, environmental impact assessment, and disaster management. It offers valuable insights into land cover changes, ecosystem health monitoring, and sustainable land management practices.

This paper presents a satellite image interpretation method to categorize satellite images into four different areas, i.e., water bodies, green areas, cloudy areas, and deserts. This interpretation can then be utilized for a number of potential applications. The proposed model provides a better accuracy, as well as empirical results verify it. To show the supremacy of the proposed model, it is also compared with some deep learning models. This work introduces an Efficient Net architecture for image classification, which has been demonstrated to be more effective than other commonly used models. This innovative approach not only increases the accuracy of classification but also reduces the computational cost of training the model. By incorporating Efficient Net, the research demonstrates a major advancement in the field of satellite image classification, opening up possibilities for a more precise and efficient analysis of remote sensing data.

The main contributions of the study can be summarized as follows:

It gives an in-depth analysis of existing techniques for satellite image classification.

It proposes a scalable deep learning model for the interpretation of satellite images.

It compares the proposed methodology with state-of-the-art techniques. Visualizations of the feature embeddings of different techniques are also presented.

The remainder of this paper is arranged as follows: Section 2 briefly discusses the work in the literature, Section 3 introduces the proposed methodology, and Section 4 discusses the achieved results and the analysis of the proposed methodology.

2.Related Work

Satellite remote sensing technology has enabled the acquisition of high-resolution imagery that can provide a wealth of information about the Earth's surface. One of the most common applications of satellite imagery is land cover classification, which involves the identification and mapping of different types of land cover, such as forests, urban areas, water bodies, and agricultural fields. Accurate land cover classification is critical for a range of applications, including natural resource management, urban planning, and disaster management. In this literature review, we aim to explore the latest advances in satellite image land cover classification, including the methods and algorithms used, the challenges involved, and the potential for further improvement. Satellite images were once considered the property of research and development giants. However, nowadays, they are becoming increasingly more accessible to research communities working in a variety of domains. Experts in these domains require an automatic interpretation of these satellite images to perform high-level analyses and to support the decision-making process. This is because the manual interpretation of such massive data is not feasible in many scenarios and can lead to inefficient solutions based on these interpretations. So, these domain experts seek support from the computer vision community to provide some solutions for the automatic interpretation and classification of satellite images.

Land cover classification using satellite images can be significant for monitoring, analyses, and planning for the improved utilization of resources. The satellite image inter- pretation process is dependent upon two basic steps: feature extraction and classification. Traditionally, handcrafted features have been utilized, and then these features are fed to conventional machine learning classifiers . Feature extraction in traditional setups has many associated and inherited challenges, affecting the quality of classification . Conventional machine learning methods, such as decision trees, support vector machines (SVMs), and random forests, have been widely used for land cover classification . These methods require manual feature engineering, where domain experts must carefully choose a set of relevant features to represent the data. The accomplishment of these techniques greatly relies on the quality of the selected features. Conventional machine learning meth- ods can achieve a good accuracy for simple classification tasks, but they may struggle when dealing with complex and high-dimensional challenging data, such as satellite images.

In the recent past, these challenges have been well dealt with due to the dawn of deep learning approaches. Deep learning has provided solutions to many challenging problems . Handcrafted feature extraction has been replaced with automatic feature extraction. This approach has demonstrated its robustness in a variety of computer vision problems. Satellite images are no exception, and researchers have employed deep learning approaches to provide satellite image classification, object detection, change detection, shoreline and landslide detection, etc.. These approaches greatly depend on the amount of data and the distribution of data.

Mehran et al. employed CNNs for ship detection in satellite images, with state-of- the-art models, such as Inception-Resnet, pre-trained on the Image-Net dataset, showing a high accuracy of over 99%. Li et al. used Dataset-RSI-CB256 to analyze the interpretation of weak labeling on satellite image classification and interpretation . They employed deep learning and adapted the dataset for this analysis. Gargees et al. worked on change detection using Satellite Image Classification Dataset-RSI-CB256 . They utilized the deep visual features of these satellite images to analyze changes in the landscape. Kwak et al. presented a semi-supervised land cover classification of remote sensing imagery using CycleGAN and EfficientNet. The related work is summarized in Table 1. Satellite image classification for land cover using traditional methods involves manual feature engineering and conventional machine learning classifiers. However, these methods may struggle with complex and high-dimensional data. To address this, deep learning approaches, such as CNNs, have been utilized and have shown robustness in various computer vision problems, including satellite image classification. Inception-Resnet, a state-of-the-art model, has demonstrated a high accuracy of over 99% for ship detection in satellite images. Other studies have also employed deep learning techniques for analyzing weak labeling, change detection, and land cover classification using satellite images. The use of EfficientNet further enhances the potential of the proposed methodology for accurate and efficient satellite image classification. The presented methodology achieves encouraging results on a challenging dataset with a high inter-class similarity and intra-class variation.

Table 1. Summary of Existing Methodologies

Reference	Problem	Feature Extraction Approach	Machine Learning Model
Abedi et al. [28]	Forest attribute categorization	Handcrafted	KNN
Sayali et al. [29]	Land region classification	Handcrafted	SVM
Kadir et al. [30]	Forest type classification	Handcrafted	KNN, MLP
Harish Kundra et al. [32]	Terrain feature extraction	Swarm optimizatio n	-
Pan et al. [34]	ver classification and segmentation	Deep learning based	Unet
Kalinicheva et al. [36]	Change detection	Deep learning based	GRU auto-encode rs
Aung et al. [37]	Building footprint extraction		CGAN
Cheng et al. [38]	Scene classification	Deep learning based	CNN
Hamida et al. [39]	Land area classification	Deep learning based	CNN
Mehran et al. [44]	Ship detection	Deep learning	Inception-Re snet

		based	

Table 1. *Cont.*

Reference	Problem	Feature Extraction Approach	Machine Learning Model
Li et al. [45]	Scene classification	Deep learning based	CNN
Gargees et al. [46]	Change detection	Deep learning based	Clustering
Kwak et al. [47]	Land cover classification	Deep learning	CycleGAN and EfficientNet

3.Materials and Methods

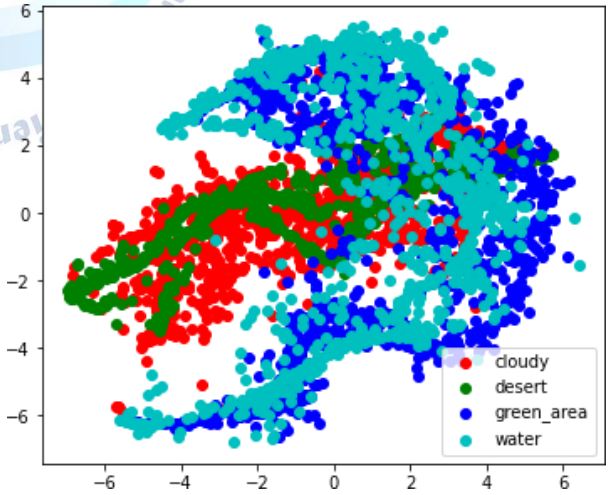
3.1 Data Description

Satellite Image Classification Dataset-RSI-CB256 is used for this research. This is a publicly available dataset and can be found in . Figure 1 shows sample images of this dataset. This dataset comprises four distinct classes acquired from sensors and Google map snapshots. The size of each image is 256 × 256. Figure 2 shows the category spread of the data. It also shows the dataset's inter-class similarities and intra-class variational challenge through t-distributed stochastic neighbor embedding (T-SNE) visualization. T- SNE visualization allows for the mapping of higher-dimensional data points to two- or three-dimensional spaces. T-SNE preserves the pairwise similarities between the high- dimensional data points as much as possible in the lower-dimensional space. In other words, the data points that are related in the high-dimensional space will be near each other in the low-dimensional space, while the data points that are disparate in the high-dimensional space will be apportioned in the low-dimensional space. Visualization shows the challenging gravity of a dataset.

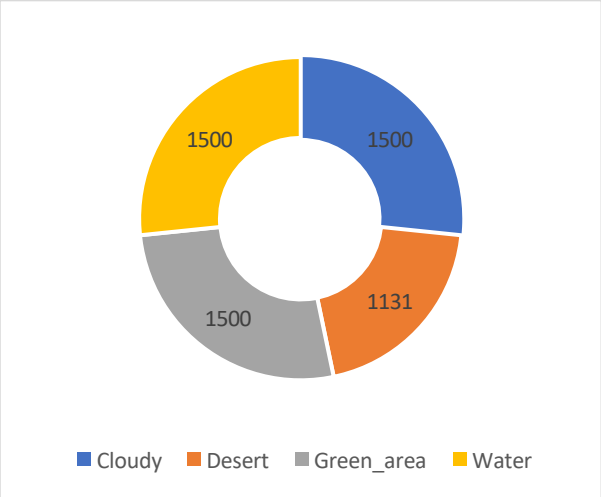
Figure 1. Samples of satellite images from Dataset-RSI-CB256.

3.2. Proposed Methodology

This research aims to classify satellite images into different categories. The proposed methodology takes the satellite image as input and categorizes it into four different classes of terrains. Pre-processing is first applied to the input images, then a feature map is extracted, and, finally, the images are classified. A complete architectural diagram is presented in Figure 3.



(**a**) T-SNE visualization



(b) doughnut chart for category distribution.

Figure 3. Architectural diagram of the proposed methodology.

3.2.1.Data Augmentation

Data scarcity is a major concern in deep learning solutions. Data augmentation is a powerful technique that can help to improve the accuracy and generalization of ma- chine learning models, especially when the size of the training dataset is limited. Data augmentation can improve the performance of a machine learning model by providing it with a more diverse range of training data. It also helps to reduce overfitting by in- troducing variations in the training data, which makes the model more robust and gen- eralizable. Data augmentation E is used to minimize the aforementioned issue, where E ∈ R_θ, HS_s, VS_t, Sh_u, Z_v, HF_w, VF_x, B_y and

$$\aleph(i,j)^c = \frac{\aleph(i,j)}{max_{s,t}\aleph(s,t)}$$

$$R_\theta(\aleph(i,j)) = \aleph(i*\cos(\theta) - j*\sin(\theta),\ i*\sin(\theta) + j*\cos(\theta))$$

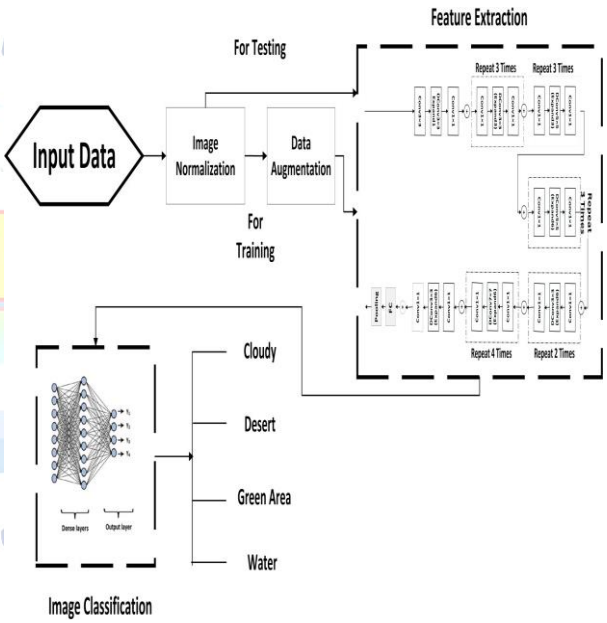
$$HS_s(\aleph(i,j)) = \aleph(i+s,j)$$
$$VS_t(\aleph(i,j)) = \aleph(i,j+t)$$
$$Sh_u(\aleph(i,j)) = \aleph(i+u*j,j)$$
$$Z_v(\aleph(i,j)) = \aleph(i*v,j*v)$$
$$HF_w(\aleph(i,j)) = \aleph(M-i-1,j)$$
$$VF_x(\aleph(i,j)) = \aleph(i,N-j-1)$$

$$B_y(\aleph(i,j)) = y*\aleph(i,j)$$

R_θ rotates an image by an angle θ, HS_s shifts an image horizontally by s pixels, VS_t shifts an image vertically by t pixels, Sh_u shears an image horizontally by a factor of u, Z_v zooms an image by a factor of v, HF_w flips an image horizontally, VF_x flips an image vertically, and B_y changes the brightness of an image by multiplying each pixel by a factor of y. By using these data augmentation techniques, the machine learning model is exposed to more variations in the data, which can help it to generalize better to new and unseen data.

3.2.2 Feature Extraction

EfficientNets are a kind of convolutional neural network that works on the principle of scaling .



EfficientNets are used as feature extractors in the presented research. The EfficientNet architecture consists of three components: the stem, the blocks, and the head. The stem is the initial part of the network that processes the input image, while the blocks are repeated units of convolutional layers that learn increasingly complex features. The head is the final part of the network that maps the learned features to the output classes. They are designed to provide state-of-the-art performance in image classification tasks while being computationally efficient. These adjust the architecture's width, resolution, and depth with compound coefficients φ. This scaling is very effective, as larger input images require more complicated architectures to compute results with more channels

and layers to analyze fine-grained patterns. Compound coefficients φ evenly adjust the width (ω), resolution (□), and depth (H) of the underlying architecture, where

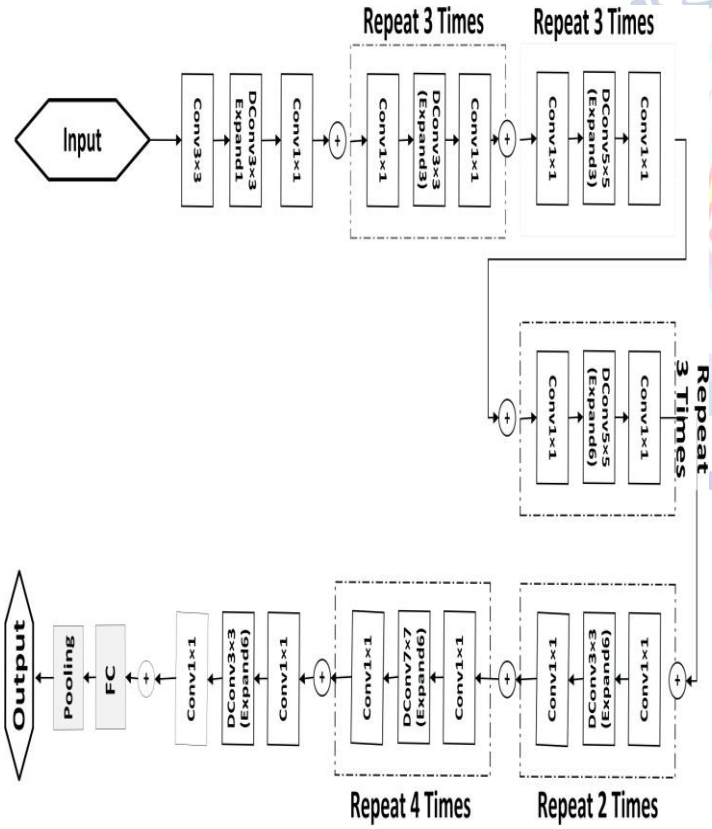
$H = X^{\varphi}$

$\omega = y^{\varphi}$

$□ = z^{\varphi}$

$\therefore X.y^2.z^2 \approx 2$

φ is the input taken from the user, and X, y, z specifies the correlation with the network width, resolution, and depth. The value of φ affects the width, depth, and resolution of the network. When φ is greater than 1, the network size increases, leading to a wider, deeper, and higher resolution. Conversely, when φ is less than 1, the network size decreases, resulting in a narrower, shallower, and lower resolution. The values of φ used in this research for EfficientNet B4–B7 are 1.8, 2.0, 2.2, and 2.4, respectively. X, y, and z can be computed through a



minor grid search. A minor grid search is a process of finding the optimal values for the coefficients y, X, and z so that the constraint $X.y^2.z^2 \approx 2$ is satisfied. This constraint ensures that the computational cost of the model remains roughly constant as these coefficients are adjusted. The equations involving the coefficients φ adjust the architecture's width, resolution, and depth. Specifically, H is the depth, ω is the width, and □ is the resolution. The equations show that φ is raised to the power of these values to adjust them.

EfficientNet proposes better and more accurate results via the compound scaling of these three network parameters. However, this will slightly increase the floating-point op- erations per second (FLOPS) by $X.y^2.z^2{}^{\varphi}$. However, $X.y^2.z^2 \approx 2$, so FLOPS are increased approximately by 2^{φ} only . X, y, and z are fixed, and the compound coefficient φ value is adjusted for B0 to obtain architectures B1–B7.

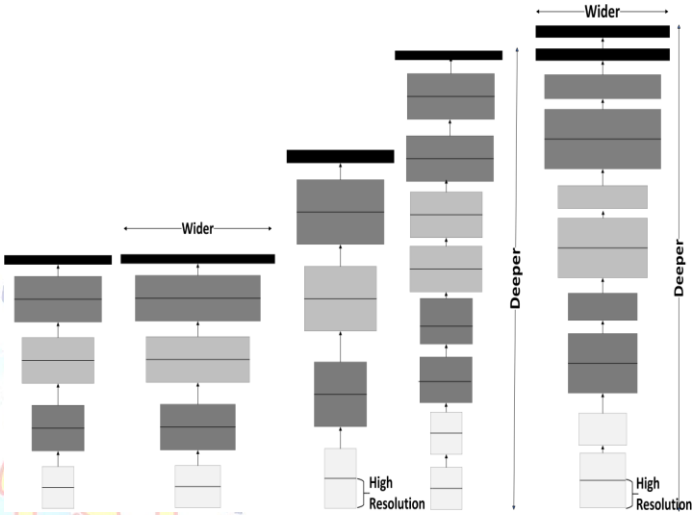


Figure 4. Scalable deep learning model: (**a**) base architecture; (**b**) width scaling; (**c**) resolution scaling; (**d**) depth scaling; (**e**) compound scaling

Figure 5. Feature extraction using EfficientNet.

MnasNet is an efficient neural network architecture designed to be versatile and optimized for efficiency. However, there are opportunities for further improvement in accuracy through modifications to the architecture. MnasNet employs SGD as the optimizer. Utilizing more sophisticated optimization techniques, such as Adam, could potentially accelerate convergence and boost the model's precision. MnasNet employs a mixture of 3 × 3 and 5 × 5 convolutional filters to process input data. Nevertheless, employing larger filters, such as 7 × 7, could capture more spatial features from the input images and result in improved accuracy. EfficientNets (B4–B7) are developed using the defined architectures and compound scaling technique outlined in . This approach allowed us to develop highly efficient models with superior performance.

3.2.3. Sample code

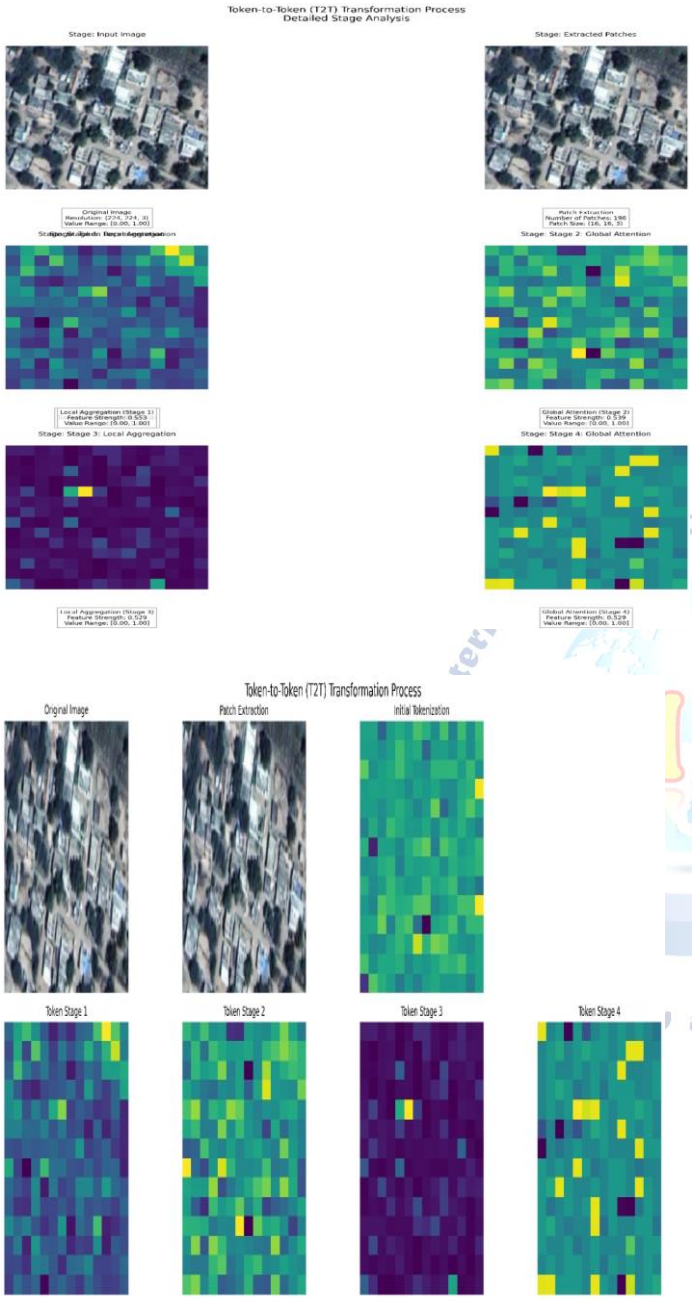
```
import cv2
import numpy as np
import os
import matplotlib.pyplot as plt

# Define preprocessing functions
def resize_image(image, target_size):
"""Resize image to the target size."""
return cv2.resize(image, target_size)

def histogram_equalization(image):
"""Apply histogram equalization."""
yuv = cv2.cvtColor(image, cv2.COLOR_BGR2YUV)
yuv[:, :, 0] = cv2.equalizeHist(yuv[:, :, 0])
return cv2.cvtColor(yuv, cv2.COLOR_YUV2BGR)

def noise_reduction(image):
    """Apply Gaussian blur for noise reduction."""
    return cv2.GaussianBlur(image, (5, 5), 0)

def augment_image(image):
    """Apply random rotation and cropping."""
    # Random rotation
    angle = np.random.randint(-30, 30)
    rows, cols, _ = image.shape
    M = cv2.getRotationMatrix2D((cols // 2, rows // 2), angle, 1)
    rotated_image = cv2.warpAffine(image, M, (cols, rows))

    # Random cropping and resizing
    crop_size = np.random.randint(int(min(rows, cols) * 0.5), min(rows, cols))
    x = np.random.randint(0, cols - crop_size)
    y = np.random.randint(0, rows - crop_size)
    cropped_image = rotated_image[y:y+crop_size, x:x+crop_size]
    cropped_resized_image = cv2.resize(cropped_image, (cols, rows))

    return rotated_image, cropped_resized_image

def random_erasing(image):
    """Apply random erasing (random black box)."""
    h, w, _ = image.shape
    eraser_area = np.random.randint(int(h * 0.1), int(h * 0.2))
    x1 = np.random.randint(0, w - eraser_area)
    y1 = np.random.randint(0, h - eraser_area)
    image[y1:y1+eraser_area, x1:x1+eraser_area] = 0  # Erase with black
    return image

def color_jittering(image):
    """Apply color jittering."""
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    hue_shift = np.random.uniform(-10, 10)
    saturation_scale = np.random.uniform(0.8, 1.2)
    value_scale = np.random.uniform(0.8, 1.2)

    hsv[:, :, 0] = np.clip(hsv[:, :, 0] + hue_shift, 0, 179)
    hsv[:, :, 1] = np.clip(hsv[:, :, 1] * saturation_scale, 0, 255)
    hsv[:, :, 2] = np.clip(hsv[:, :, 2] * value_scale, 0, 255)

    return cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

def preprocess_and_display(input_folder, output_folder, target_size=(244, 244)):
    """Preprocess the dataset and display processed images."""
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    classes = ['desert', 'forest', 'green fields', 'oceans', 'urban areas']

    # Prepare the figure for plotting (7 preprocessing steps, 5 classes)
    fig, axes = plt.subplots(len(classes), 7, figsize=(20, 5 * len(classes)))

    for i, class_name in enumerate(classes):
        class_folder = os.path.join(input_folder, class_name)
        output_class_folder = os.path.join(output_folder, class_name)
        os.makedirs(output_class_folder, exist_ok=True)

        # Get all image files in the class folder
        image_filenames = [f for f in os.listdir(class_folder) if f.lower().endswith(('.jpg', '.jpeg', '.png', '.bmp'))]

        if not image_filenames:
```

```
        print(f"No images found in {class_name}. Skipping class.")
        continue

    # Process only one sample image per class for simplicity
    sample_image = cv2.imread(os.path.join(class_folder, image_filenames[0]))

    # Preprocessing steps
    resized_image = resize_image(sample_image, target_size)
    hist_eq_image = histogram_equalization(resized_image)
    noise_reduced_image = noise_reduction(hist_eq_image)
    rotated_image, cropped_resized_image = augment_image(noise_reduced_image)
    erased_image = random_erasing(cropped_resized_image)
    color_jittered_image = color_jittering(resized_image)

    # Save processed images
    cv2.imwrite(os.path.join(output_class_folder, f'resized_{image_filenames[0]}'), resized_image)
    cv2.imwrite(os.path.join(output_class_folder, f'hist_eq_{image_filenames[0]}'), hist_eq_image)
    cv2.imwrite(os.path.join(output_class_folder, f'noise_reduced_{image_filenames[0]}'), noise_reduced_image)
    cv2.imwrite(os.path.join(output_class_folder, f'rotated_{image_filenames[0]}'), rotated_image)
    cv2.imwrite(os.path.join(output_class_folder, f'cropped_resized_{image_filenames[0]}'), cropped_resized_image)
    cv2.imwrite(os.path.join(output_class_folder, f'erased_{image_filenames[0]}'), erased_image)
    cv2.imwrite(os.path.join(output_class_folder, f'color_jittered_{image_filenames[0]}'), color_jittered_image)

    # Plotting the processed images
    images_to_plot = [
      resized_image, hist_eq_image, noise_reduced_image, rotated_image, cropped_resized_image, erased_image, color_jittered_image
    ]
    titles = [
      'Resized', 'Histogram Equalization', 'Noise Reduced', 'Rotated', 'Cropped Resized', 'Random Erased', 'Color Jittered'
    ]

    for j, (img, title) in enumerate(zip(images_to_plot, titles)):
        img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        axes[i, j].imshow(img_rgb)
        axes[i, j].set_title(f"{class_name} - {title}")
        axes[i, j].axis('off')
  plt.tight_layout()
  plt.show()

# Example usage:
input_folder = r"C:\Users\HP\OneDrive\Desktop\SIC\1 SIC preprocessing\data1"  # Replace with your folder path
output_folder = r"C:\Users\HP\OneDrive\Desktop\SIC\1 SIC preprocessing\data"  # Replace with your folder path
preprocess_and_display(input_folder, output_folder, target_size=(244, 244))
```

4.RESULTS

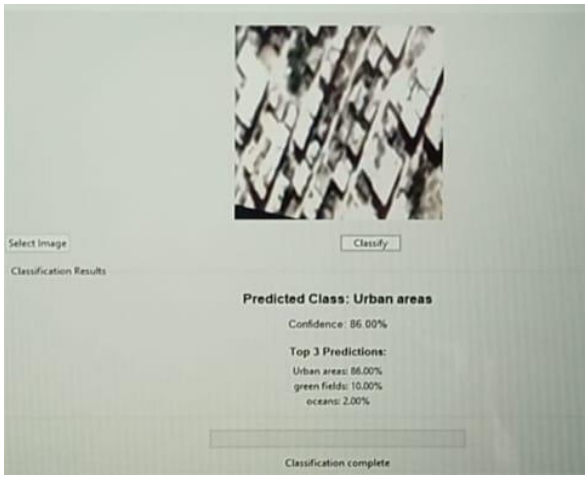
Satellite Image Classification Dataset-RSI-CB256 is used for this research. The batch size for experiments is 32. The input size of the proposed network is 600 × 600 px. The image samples are resized to fit in the model. Our approach involves utilizing the RMSProp optimizer, which has a decay rate of 0.9 and a momentum of 0.9. Additionally, we incorporate batch normalization after each convolution layer with a momentum of 0.99, and we apply a weight decay of 1×10^{-5}. To prevent overfitting, we implement a dropout rate of 0.2 on the final layer. As suggested by , we gradually increase the learning rate from 0 to 0.256 during the first 5 epochs, and we subsequently decrease it by a factor of 0.97 every 2.4 epochs. Splits of 80%, 10%,

and 10% are used for training, validation, and testing. The sample selection for training and validation remains the same for all experiments. A validation set is utilized for an early stopping process.



4.1. Experimental Results

The proposed methodology is tested on the RSI-CB256 dataset. The results of other popular architectures are also reported using the same dataset. Table 2 shows the detailed results of different architectures on the aforementioned dataset. EfficientNets in general, and EfficientNet-B7 in particular, has the ability to classify the satellite image data. Efficient- Net B7 reports the highest accuracy amongst all architectures. However, it is important to note that accuracy alone may not be the best metric for evaluating the performance of a machine learning model, especially in cases where the classes are imbalanced or there are multiple classes.



5.CONCLUSIONS

Satellite image analyses have a variety of potential applications. Satellite images present many additional challenges as compared with ordinary digital images. This paper presents an end-to-end deep-learning-based satellite image categorization method. The proposed method relies on visual clues from the spatial information of satellite images. Sat+

ellite Image Classification Dataset-RSI-CB256 is employed for experiments. The dataset has high intra-class and low inter-class variations. This makes the analysis of satellite images even more challenging. However, the proposed method shows encouraging results.

The proposed model applies scalable networks for feature extraction. EfficientNets are a type of convolutional neural network that specializes in image classification tasks while being designed to be computationally efficient. The network is composed of three main parts: the stem, blocks, and head. The network's width, resolution, and depth can be adjusted using compound coefficients. The input parameter φ determines the size of the network, with a larger φ resulting in wider, deeper, and higher-resolution networks.

The proposed model is compared with many popular deep learning models, such as Inception V3, ResNet-50, and MobileNet. The experimental results testify to the proposed methodology's dominance. The

proposed methodology gives a 0.997 F1 score, the highest among all the models studied.

The presented study can be extended by applying it to different datasets. It can also be tested for more than four classes of satellite images. Moreover, other deep learning models can also be analyzed. It can also be a good research direction to see the effect of different backbone models in Efficient Nets

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

1. Singh, K.K.; Singh, D.K.; Thakur, N.K.;Dewali, S.K.; Negi, H.S.; Snehmani ; Mishra, V.D. Detection and mapping of snow avalanche debris from western Himalaya, India using remote sensing satellite images. *Geocarto Int.* 2022, *37*, 2561–2579.
2. Tripodi, S.; Girard, N.; Fonteix, G.; Duan, L.; Mapurisa, W.; Leras, M.; Trastour, F.; Tarabalka, Y.; Laurore, L. Brightearth: Pipeline for on-the-fly 3D reconstruction of urban and rural scenes from one satellite image, ISPRS Annals of the Photogrammetry. Remote Sens. Spat. Inf. Sci. 2022, 3, 263–270.
3. Soldi, G.; Gaglione, D.; Forti, N.; Simone, A.D.; Daffinà, F.C.; Bottini, G.; Quattrociocchi, D.; Millefiori, L.M.; Braca, P.; Carniel, S.Space-based global maritime surveillance. IEEE Aerosp. Electron. Syst. Mag. 2021, 36, 8–28.
4. Dehkordi, R.H.; Pelgrum, H.; Meersmans, J. High spatio-temporal monitoring of century-old biochar effects on evapotranspiration through the etlook model: A case study with uav and satellite image fusion based on additive wavelet transform. GIScience Remote Sens. 2022, 59, 111–141
5. Sasaki, K.; Sekine, T.; Burtz, L.J.; Emery, W.J. Coastal marine debris detection and density mapping with very high resolution satellite imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2022, 15, 6391–6401.
6. Bulushi, S.A. From the sky to the streets, and back: Geographies of imperial warfare in east Africa. Soc. Text 2022, 40, 37–59.